This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Copyright © 2012, 2013, Oracle and/or its affiliates. All rights reserved.
Contents

Merging User and System Menus ................................................................. 40

4 Installing Themes ........................................................................................................ 43
Themes Overview ...................................................................................................... 43
Theme Index File ...................................................................................................... 44
Installing a Controls Option .................................................................................. 45
Installing a Window Border Option ....................................................................... 45
Installing an Icons Option ...................................................................................... 46
Installing Icons for Themes .................................................................................... 46
▼ How to Install an Icon for a Theme ................................................................. 47
Creating a Custom Controls Option ...................................................................... 47
▼ How to Create a Custom Controls Option ......................................................... 47

5 Customizing Fonts .................................................................................................. 49
Fonts Overview ....................................................................................................... 49
fontconfig Library ................................................................................................. 49
Legacy X11 Font System ......................................................................................... 50
Substituting Fonts ................................................................................................. 52

6 Working With MIME Types ...................................................................................... 55
MIME Types Overview ............................................................................................ 55
About the MIME Database ...................................................................................... 56
MIME Database Contents ....................................................................................... 57
Refreshing the MIME Database .............................................................................. 59
Understanding MIME Type XML Files .................................................................. 59
Modifying MIME Types .......................................................................................... 61
▼ How to Add or Modify MIME Types ............................................................... 61
Registering Applications for MIME Types ............................................................ 63
▼ How to Register Applications for MIME Types .............................................. 64
▼ How to Associate a MIME Type With an Application ..................................... 64

7 Managing Screensavers .......................................................................................... 67
Setting Screensaver Preferences .......................................................................... 67
Setting Default Screensaver Preferences for All Users ....................................... 67
Modifying Screensaver Preferences ................................................................. 68
Modifying Look and Feel of Your Screensaver .............................................. 68
▼ How to Add a Screensaver Display ................................................................. 68
Disabling a Screensaver Display ..................................................................... 68

8 Managing Sessions .......................................................................................... 71
Session Manager Overview ............................................................................... 71
Setting Session Defaults .................................................................................. 71
Restoring the Default Session Settings ............................................................ 73
Saving the Current Session as the Default Session .......................................... 73

9 Overview of the Yelp Help Browser ................................................................. 75
Online Help Source Documents .......................................................................... 75
Open Source Metadata Framework Files .......................................................... 75
Rarian Cataloging System .................................................................................. 76

10 Improving the Performance of the Oracle Solaris Desktop System .............. 77
Reducing CPU Usage ....................................................................................... 77
  Using Theme Options .................................................................................. 78
  Turning Off the Display of Icons in Menus .................................................. 78
  Turning Off Panel Animation ..................................................................... 79
  Using a Solid Color for the Desktop Background ........................................ 79
Improving File Manager Performance ............................................................ 79
  Using the Appropriate Window Manager .................................................... 81
Reducing X Window System Network Traffic .................................................. 82
Reducing Color Usage and Improving Display Quality .................................... 82

11 Disabling Features in the Oracle Solaris Desktop System ............................. 85
Setting Lockdown Preferences ........................................................................ 85
  Disabling Lock Screen ................................................................................ 85
  Disabling Log Out ...................................................................................... 86
Disabling Command-Line Operations ............................................................. 86
Disabling Panel Configuration ......................................................................... 87
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working With the X Window System</td>
<td>89</td>
</tr>
<tr>
<td>X Window System Overview</td>
<td>89</td>
</tr>
<tr>
<td>Understanding the X Server Process</td>
<td>90</td>
</tr>
<tr>
<td>Working With X Server Process</td>
<td>90</td>
</tr>
<tr>
<td>Working With X Clients</td>
<td>91</td>
</tr>
<tr>
<td>Configuring an X Server in Oracle Solaris</td>
<td>92</td>
</tr>
<tr>
<td>Working With Xorg Configuration Files</td>
<td>92</td>
</tr>
<tr>
<td>Working With SMF Configuration Properties</td>
<td>93</td>
</tr>
<tr>
<td>Working With HAL fdi Files</td>
<td>94</td>
</tr>
<tr>
<td>Accessing X11 Display</td>
<td>94</td>
</tr>
<tr>
<td>Providing Other User Accounts With Access to Your Display</td>
<td>95</td>
</tr>
<tr>
<td>Displaying a Client From a Remote Machine</td>
<td>95</td>
</tr>
<tr>
<td>Setting Up Remote Desktop Access Using VNC</td>
<td>96</td>
</tr>
<tr>
<td>▼ How to Set Up VNC to Provide a Guest Graphical Login</td>
<td>96</td>
</tr>
<tr>
<td>▼ How to Start VNC Manually</td>
<td>97</td>
</tr>
<tr>
<td>Hidden Directories</td>
<td>99</td>
</tr>
<tr>
<td>Glossary</td>
<td>101</td>
</tr>
<tr>
<td>Index</td>
<td>103</td>
</tr>
</tbody>
</table>
Preface

Oracle Solaris 11.1 Desktop Administrator’s Guide describes how to administer systems running the Oracle Solaris desktop. Most of the information in this guide is generic to all releases of the desktop. Where the information is not generic, the platform is indicated.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the Oracle Solaris OS: Hardware Compatibility Lists.

This document cites any implementation differences between the platform types.

Supported Systems

This release of the desktop supports systems running the Oracle Solaris 11 operating system on SPARC and x86 platforms.

Who Should Read This Guide

This guide is for system administrators and anyone who is interested in using the Oracle Solaris Desktop.

How This Book Is Organized

This guide is structured in this manner:

- Chapter 1, “Administering the Oracle Solaris Desktop,” introduces you to some of the basic components used for administering the Oracle Solaris Desktop, which is based on the GNOME Desktop version 2.30.2.
- Chapter 2, “Managing User Preferences With GConf,” describes how to use the GNOME configuration framework (GConf) to set mandatory and default desktop preferences for all users.
- Chapter 3, “Customizing Menus,” describes how the Oracle Solaris Desktop implements menus and how you can customize menus.
Chapter 4, “Installing Themes,” describes the settings that control the visual appearance of the desktop, how they are organized into themes, and how to create and install new themes.

Chapter 5, “Customizing Fonts,” describes how to install new fonts, and how to create font substitutions.

Chapter 6, “Working With MIME Types,” describes how applications detect MIME types, how to register MIME types, and how to add applications to the desktop.

Chapter 7, “Managing Screensavers,” describes how to set preferences for the screensaver. This chapter also provides information about how to modify the displays that are available for the screensaver.

Chapter 8, “Managing Sessions,” introduces session management, and describes how to set session defaults.

Chapter 9, “Overview of the Yelp Help Browser,” describes the help system in the Oracle Solaris Desktop system.

Chapter 10, “Improving the Performance of the Oracle Solaris Desktop System,” describes how to reduce CPU usage and X Window System network traffic, if necessary, and how to optimize display quality on 8-bit displays.

Chapter 11, “Disabling Features in the Oracle Solaris Desktop System,” describes how to prevent users from accessing or customizing certain features of the desktop.

Chapter 12, “Working With the X Window System,” provides an overview of the X Window System that is available in the Oracle Solaris OS.

Appendix A, “Hidden Directories,” describes the hidden configuration directories that the Oracle Solaris Desktop system adds to the home directories of users.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

<table>
<thead>
<tr>
<th>Typeface</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AaBbCc123</td>
<td>The names of commands, files, and directories, and onscreen computer output</td>
<td>Edit your .login file. Use ls -a to list all files. machine_name% you have mail.</td>
</tr>
</tbody>
</table>
The command to remove a file is `rm filename`.

A cache is a copy that is stored locally.

Do not save the file.

Note: Some emphasized items appear bold online.

---

**Shell Prompts in Command Examples**

The following table shows UNIX system prompts and superuser prompts for shells that are included in the Oracle Solaris OS. In command examples, the shell prompt indicates whether the command should be executed by a regular user or a user with privileges.

**TABLE P-2  Shell Prompts**

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bash shell, Korn shell, and Bourne shell</td>
<td>$</td>
</tr>
<tr>
<td>Bash shell, Korn shell, and Bourne shell for superuser</td>
<td>#</td>
</tr>
<tr>
<td>C shell for superuser</td>
<td>machine_name#</td>
</tr>
<tr>
<td>C shell</td>
<td>machine_name%</td>
</tr>
</tbody>
</table>
This chapter introduces you to some of the basic components used to administer the Oracle Solaris Desktop, which is based on the GNOME Desktop version 2.30.2.

The common administrative tasks include setting user preferences, customizing menus and fonts, installing themes, and managing screensavers. The chapters in this guide describe how to perform these administrative tasks in detail.

**Desktop Overview**

The default desktop environment in Oracle Solaris 11 is the Oracle Solaris Desktop, which includes GNOME 2.30.2 from the GNOME Foundation. The desktop lies behind all other components on the screen. When no windows are visible, the desktop is that part of the screen between the top and bottom panels. You can place files and folders that you want to be able to access easily on the desktop.

By default, the following icons are visible on the desktop:

- **Computer icon** – Provides access to CDs, removable media such as floppy disks, and the entire file system (also known as the root file system). By default, you do not have security permissions to read other users' files or edit system files.
- **Home folder** – Stores personal files of the user. You can also open this folder from the Places menu.
- **Trash** – Contains files and folders you have deleted and no longer need.

For more information about the Oracle Solaris Desktop, see *Oracle Solaris 11.1 Desktop User’s Guide*.

For an overview of the X Window System that is available in the Oracle Solaris OS, see Chapter 12, “Working With the X Window System.”
Managing User Preferences With GConf

This chapter describes how to use the GNOME configuration framework (GConf) and manage user preferences with GConf.

This chapter includes the following information:
- “GConf Overview” on page 13
- “GConf Repository Components” on page 14
- “GConf Daemon” on page 18
- “Working With the GConf Command-Line Tool” on page 18
- “Setting Preference Values” on page 21
- “Restoring Default Preference Values” on page 29

GConf Overview

GConf is a framework used by the Oracle Solaris Desktop environment to store configuration settings of the desktop and applications. GConf simplifies the administration of preferences for Oracle Solaris Desktop users.

GConf enables system administrators to do the following tasks:
- 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.

GConf also notifies applications when a preference value changes across a network. When you change a preference, all applications that use the preference are immediately updated.
GConf has the following components:

- Repository of user preferences
- Daemon, gconfd-2
- Command-line tool, gconf tool-2

GConf Repository Components

The GConf repository is structured like a simple file system, which contains keys organized into a hierarchy. Each key is either a directory containing more keys or has a value.

The repository is organized into logical groups based on the application type and contains the following directories:

- Directories that correspond to applications that use the GConf repository. You can see the following directories:
  - system
  - desktop
  - apps

  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.

- A directory called schemas that contains schema files. The schema files describe all preference keys.

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.

Preference keys typically have simple data types, such as the following:

- Strings
- Integers
- Lists of strings
- Lists of integers

The format of the preference key in the repository depends on the Extensible Markup Language (XML) 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 XML backend module is used to read the repository:

```xml
<entry name="font_name" mtime="1038323555" muser="user123" type="string">
  <stringvalue>Sans 10</stringvalue>
</entry>
```
**Note** – When this guide refers to a preference key, the path to the key is added to the name of the key. For example, the `font_name` preference key in the `/desktop/gnome/interface` subdirectory is referred to as `/desktop/gnome/interface/font_name`.

### 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, which is located in the `/etc/gconf/2` directory.

The path file specifies the following information for each configuration source:

- Backend module to use to read the repository
- Permissions on the repository
- Location of the repository

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

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

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

<table>
<thead>
<tr>
<th>Configuration Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandatory</td>
<td>The permissions to the configuration source are set to read only. Users cannot overwrite the values in this source. So, the preferences in the source are mandatory.</td>
</tr>
<tr>
<td>User</td>
<td>The configuration source is stored in the <code>.gconf</code> directory in the home directory of the user. When the user sets a preference, the new preference information is added to this location. You can use the Configuration Editor application to modify the user configuration source.</td>
</tr>
<tr>
<td>Default</td>
<td>The configuration source contains the default preference settings.</td>
</tr>
</tbody>
</table>
GConf applies preferences in the following order of priority:

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

The include instructions in the GConf path file enable you to specify other configuration sources.

<table>
<thead>
<tr>
<th>Included Configuration Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/gconf/2/local-mandatory.path</td>
<td>Use this configuration source to store mandatory preference values for a particular system.</td>
</tr>
<tr>
<td>$(HOME)/.gconf.path</td>
<td>Specify the location of the configuration source in the home directory in the .gconf.path file.</td>
</tr>
<tr>
<td>/etc/gconf/2/local-defaults.path</td>
<td>Use this configuration source to store default preference values for a particular system.</td>
</tr>
</tbody>
</table>

**Schema**

A GConf schema is a collective term for a GConf schema key and a GConf schema object. Schema keys, schema objects, and their relationship to preference keys are defined as follows:

- **Preference key** – An element in the GConf repository that corresponds to an application preference, for example, /desktop/gnome/interface/font_name.
- **Schema key** – A key that stores a schema object for a preference key, for example, /schemas/desktop/gnome/interface/font_name.
- **Schema object** – An element in a configuration source that contains the following information for a preference key:
  - Name of the application that uses the preference key
  - Type of value required for the preference key, for example, integer, boolean, and so on.
  - Default value for the preference key
  - Brief documentation about the preference key

For example, the schema object for the /schemas/desktop/gnome/interface/font_name key is as follows:

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

```xml
<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.

**Schema Definition Files**

Schemas are generated from schema definition files. A schema definition file defines the
characteristics of all keys for a particular application and is used to create a new configuration
source. The schema definition files have a .schemas extension and are included in the
/etc/gconf/schemas directory.

Some schema definition files correspond closely to a part of the Oracle Solaris Desktop user
interface. For example, the system_http_proxy.schemas file describes keys that correspond to
the preferences in the Internet preference tool.

Other schema definition files describe keys that are not exposed by the Oracle Solaris Desktop
user interface. For example, the panel-global.schemas file describes the
/apps/panel/global/tooltips_enabled key. This key, which controls whether tooltips are
shown on desktop panels, is not exposed as a preference in any desktop preference tool. Such
keys might be modified using the gconf tool-2 command. For more information, see "Working
With the GConf Command-Line Tool" on page 18.

Some parts of the Oracle Solaris Desktop user interface contain preferences that represent
GConf 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.
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 and 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.

By default, an instance of the GConf daemon is started for each user. The GConf daemon does not handle complex problems such as authentication and data security. When the GConf daemon starts, the daemon loads the GConf path file from the `/etc/gconf/version/path` directory. The GConf daemon manages all access between applications and the configuration sources.

When an application requests the value of a preference key, the GConf daemon uses the following process to search the configuration sources in a particular order:

1. Searches for the value of the preference key in each configuration source in the order specified in the path file. If the value is found, returns the value.
2. If a value is not found, searches 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, checks the value of the schema key.
4. If the value of the schema key is a schema object, returns the suggested value in the `<default>` element of the schema object.

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

To stop the GConf daemon, type the following command:

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

### Working With the 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, you would use the following command to display the values of all keys in the `/desktop/gnome` directory and subdirectories.

```
# gconftool-2 --recursive-list /desktop/gnome
```
The following table describes some of the options that you can use with the `gconftool-2` command. For detailed command-line options, see the `gconftool-2(1)` man page.

**TABLE 2–1**  
**gconftool-2 Command-Line Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>--all-dirs</td>
<td>Lists all subdirectories in the specified directory.</td>
</tr>
<tr>
<td>--all-entries</td>
<td>Displays the values of all keys in the specified directory.</td>
</tr>
<tr>
<td>--config-source=configuration-source</td>
<td>Use this option with the --direct option to specify a configuration source to use.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> – If you do not specify a configuration source with this option, the command runs on all configuration sources in the path file.</td>
</tr>
<tr>
<td>--direct</td>
<td>Use this option with the --config-source option to access a configuration source directly. When you use this option, GConf bypasses the server.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> – Ensure that the GConf daemon, <code>gconfd-2</code>, is not running before you use this option.</td>
</tr>
<tr>
<td>--dump</td>
<td>Generates a list that contains all preference keys in the specified GConf repository directory. The list contains XML descriptions of all the keys. The list is contained in a <code>&lt;gconfentryfile&gt;</code> element.</td>
</tr>
<tr>
<td></td>
<td>For example, you can redirect the output from this option to generate a file that lists all keys that are related to your panel configuration. You can use the --load option with this file.</td>
</tr>
<tr>
<td>--get</td>
<td>Displays the value of the specified preference key. Also displays the values of the elements in the schema object for the specified schema key.</td>
</tr>
<tr>
<td>--help</td>
<td>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.</td>
</tr>
<tr>
<td>--load=filename</td>
<td>Sets the values of preference keys in the current directory in a configuration source to the values in a specified file. The file that you specify must contain XML descriptions of the keys in a <code>&lt;gconfentryfile&gt;</code> element.</td>
</tr>
<tr>
<td>--long-desc=description</td>
<td>Use this option with the --set-schema option to specify a long description for a schema key.</td>
</tr>
<tr>
<td>--makefile-install-rule</td>
<td>Installs schema definition files to applications.</td>
</tr>
<tr>
<td>--owner=owner</td>
<td>Use this option with the --set-schema option to specify an owner for a schema key.</td>
</tr>
<tr>
<td>--recursive-list</td>
<td>Displays the values of all preference keys in all subdirectories in the specified directory.</td>
</tr>
</tbody>
</table>
## gconftool-2 Command-Line Options (Continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>--recursive-unset</td>
<td>Resets the values of all preference keys in all subdirectories in a directory to the settings in the default configuration source.</td>
</tr>
</tbody>
</table>
| --set          | Sets the value of a preference key and writes the value to the user configuration source. Use the --type option with the --set option to specify the data type of the value that you want to set. For example, the following command sets the value of the /apps/gnome-terminal/profiles/Default/background_color key in the user configuration source:  
```
# gconftool-2 --set 
"/apps/gnome-terminal/profiles/Default/background_color" \ 
--type string "#000000"
```
You can also use the --direct option and the --config-source option with the --set option to write a value to another configuration source. |
| --set-schema    | Sets the value of an attribute in a schema key and writes the value to the default configuration source. Use the following options with the --set-schema option to specify the attribute that you want to update:  
- --type  
- --short-desc  
- --long-desc  
- --owner  
For example, the following command sets the short description in the schema key for the /apps/gnome-terminal/profiles/Default/background_color key:  
```
# gconftool-2 --set-schema \ 
"/schemas/apps/gnome-terminal/profiles/Default/background_color" \ 
--short-desc "Default background color of terminal"
```
| --short-desc=description | Use this option with the --set-schema option to specify a short description for a schema key. |
| --shutdown       | Terminates the GConf daemon.                                               |
**TABLE 2-1  gconftool-2 Command-Line Options (Continued)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
</table>
| $\cdot$-type=data-type | Use this option to specify the data type when you set a value of a preference key. You can also use this option when you set the value of an attribute in a schema key. The valid data types are:  
  - bool  
  - float  
  - int  
  - list  
  - pair  
  - string |
| $\cdot$-unset | Resets the value of a preference key from the user setting to the setting in the default configuration source. |

**Note** – In a multiuser environment, you can set up a server such that the server provides desktop environments to multiple clients, such as Sun Ray clients or VNC viewers. The desktop environment provided to clients can be optimized for better performance and usability.

The optimizations for the desktop environment are available in the group/feature/multi-user-desktop image packaging system (IPS) package. You must install this IPS package on the server to enable optimizations for the desktop. For detailed information, see Optimizing the Oracle Solaris Desktop for a Multi-User Environment.

### Setting Preference Values

You can set a mandatory value or a default value for a preference key. Mandatory values can be set only by administrators and users cannot override these values. However, users can change the default values.

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 -d  
--config-source configuration-source  
--type data-type  
--set preference-key value
```

For example, to set **www.proxy.xyz.com** as the mandatory HTTP proxy host, type the following command:
Setting Preference Values

```bash
# gconf-tool-2 --direct \
  --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory \
  --type string --set /system/http_proxy/host www.proxy.xyz.com
```

**Note** – The user cannot override this mandatory preference value.

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

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

**Note** – The user can override this default preference value.

## General Preferences

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

### HTTP Proxy Preferences

To set HTTP proxy preferences, 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, type the following command:

```bash
# gconf-tool-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, type the following command:

```bash
# gconf-tool-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 about the other HTTP proxy preferences, see the `system_http_proxy.schemas` schema definition file.

### Number of Workspaces

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

```bash
# gconf-tool-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, type 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 about the other window manager preferences, see the `metacity.schemas` schema definition file.

**Keyboard Accessibility Preferences**

To set keyboard accessibility preferences, 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, type 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, type 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 about the other keyboard accessibility preferences, see the `desktop_gnome_accessibility_keyboard.schemas` schema definition file.

**Keyboard Shortcut Preferences**

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

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

You can also set other keyboard shortcut preferences. For information about the other keyboard shortcut preferences, see the `metacity.schemas` schema definition file.
Specifying Panel and Panel Object Preferences

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

- Number of panels
- Types of panels
- Properties of panels
- Contents of panels

To configure individual panels and panel objects, you must first understand the structure of the `panel-default-setup.entries` file.

To set preferences for individual panels and panel objects, you must set the values of multiple preferences in a configuration source. The easiest way to set the values of panel preferences is to use the `gconftool-2` command with the `--dump` and `--load` options.

Individual Panels and Panel Objects Structure

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

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

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

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

```xml
<entry>
  <key>toplevel_id_listkey>toplevel_id_list>
  <schema_key>/schemas/apps/panel/general/toplevel_id_listschema_key>/schemas/apps/panel/general/toplevel_id_list>
  <value>
    <list type="string">
      <value>
        <string>bottom_panelstring>bottom_panel>
      </value>
    </list>
  </value>
</entry>
```

In the `panel-default-setup.entries` file, the identifier `bottom_panel` identifies the bottom edge panel.
Keys that specify the properties of the panels. The panel property keys are structured as follows:

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

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

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

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

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

```xml
<entrylist base="/apps/panel/default_setup/applets/window_list">
  <entry>
    <key>object_type</key>
    <schema_key>/schemas/apps/panel/objects/object_type</schema_key>
    <value>
      <string>bonobo-applet</string>
    </value>
  </entry>
  <entry>
    <key>toplevel_id</key>
    <schema_key>/schemas/apps/panel/objects/toplevel_id</schema_key>
    <value>
      <string>bottom_panel</string>
    </value>
  </entry>
  <entry>
    <key>position</key>
    <schema_key>/schemas/apps/panel/objects/position</schema_key>
    <value>
      <int>0</int>
    </value>
  </entry>
</entrylist>
```
The OAFIID is a unique identifier for an applet. To find the OAFIID for a particular applet, see the .server file for the applet in the /usr/lib/bonobo/servers directory. For example, the following excerpt from GNOME_Wncklet_Factory.server shows the OAFIID for the Window List applet:

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

### How to Set Preferences for Individual Panels and Panel Objects

1. **Log in with a user account.**

2. **Use the `--dump` option with the gconftool-2 command to generate a file that contains an XML description of your panel configuration.**

   The `--dump` option generates a list that contains all preference keys in the specified GConf repository directory. For example, the following command creates an XML description of the default panel configuration in the /my-panel-setup.entries file:

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

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

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

   ```xml
   <entry>
     <key>objects/object_16/launcher_location</key>
     <value>
       <string>hadjaha-00adce02f7.desktop</string>
     </value>
   </entry>
   ```

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

4. **Change the positions of panel objects from absolute positions to relative positions.**

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

If you want object positions to be relative to the right side of the panel, set the value of the panel_right_stick key to true. For example, the following excerpt places the Show Desktop button in the second available location from the extreme right of the panel.

```
<entry>
  <key>applets/show_desktop_button/panel_right_stick</key>
  <schema_key>/schemas/apps/panel/objects/panel_right_stick</schema_key>
  <value>
    <bool>true</bool>
  </value>
</entry>
<entry>
  <key>applets/show_desktop_button/position</key>
  <schema_key>/schemas/apps/panel/objects/position</schema_key>
  <value>
    <int>1</int>
  </value>
</entry>
```

Use the --load option with the gconftool-2 command to set the values of the default configuration source to the values in the my-panel-setup.entries file.

For example, the following command sets the values of the keys in the default configuration source to the values of the corresponding keys in my-panel-setup.entries:

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

**Look-and-Feel Preferences**

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

**Font Preferences**

To set font preferences, 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.

<table>
<thead>
<tr>
<th>GConf Location</th>
<th>User Interface Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>/desktop/gnome/interface/font_name</td>
<td>Application font option in the Font preference tool</td>
</tr>
<tr>
<td>/desktop/gnome/interface/document_font_name</td>
<td>Document font option in the Font preference tool</td>
</tr>
<tr>
<td>/desktop/gnome/interface/monospace_font_name</td>
<td>Fixed Width font option in the Font preference tool</td>
</tr>
<tr>
<td>/apps/metacity/general/titlebar_font</td>
<td>Window Title font option in the Font preference tool</td>
</tr>
</tbody>
</table>
GConfig Location | User Interface Component
--- | ---
/apps/nautilus/preferences/desktop_font | Desktop font option in the Font preference tool

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

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

To set Palatino 12 as the default desktop object font, type 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"
```

### Background Preferences

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

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

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

### Splash Image Preferences

To set splash image preferences, modify the value of the preference keys in the /apps/gnome-session/options/ location. For example, if you do not want users 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, type the following command:

```
# gconftool-2 --direct --config-source \
xml:readwrite:/etc/gconf/gconf.xml.defaults --type bool \n--set /apps/gnome-session/options/show_splash_screen false
```
You can also set other splash image preferences. For information about the other splash image preferences, see the `gnome-session.schemas` schema definition file.

**Restoring Default Preference Values**

To restore the default preference values for a user, type 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 user’s home directory. For example:

```
# gconftool-2 --direct 
--config-source xml:readwrite:$(HOME)/.gconf --recursive-unset
```

This command resets the values of all the preference keys, in all the subdirectories, from the user setting to the setting in the default configuration source file.
This chapter describes how to customize and edit menus in the Oracle Solaris Desktop. The chapter also describes how to work with the menu definition files, directory entry files, and desktop entry files.

This chapter includes the following information:

- "Menus Overview" on page 31
- "Working With Menu Definition Files" on page 32
- "Working With Directory Entry Files" on page 34
- "Working With Desktop Entry Files" on page 35
- "Editing Menus" on page 37
- "Merging Menus" on page 40

**Menus Overview**

The Oracle Solaris Desktop implements menus according to the **XDG menu specification**. By supporting this specification, the Oracle Solaris Desktop enables you to:

- Customize the menu hierarchy easily.
  - You need to edit only 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 applications.
- Configure menus so that users cannot modify the menus.

The menus in the Oracle Solaris Desktop use the following file components:

- Menu definition files
- Desktop entry files
Working With Menu Definition Files

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

Menu files must reside in the $XDG_CONFIG_DIRS/menus/applications.menu directory. To work with menu files, you must set the $XDG_CONFIG_DIRS environment variable, defined in the XDG base directory specification.

To resolve the location of the applications.menu file, the system searches the default path in the following order:

1. Searches each directory in the $XDG_CONFIG_HOME path to find the menus/applications.menu file. If the $XDG_CONFIG_HOME environment variable is not set, the search defaults to the ~/.config/ directory.
2. Searches each directory in the $XDG_CONFIG_DIRS path to find the menus/applications.menu file. If the $XDG_CONFIG_DIRS environment variable is not set, the search defaults to the /etc/xdg/ directory.
3. Uses the first applications.menu file found.

The following example shows a sample .menu file:

EXAMPLE 3-1 Example of a .menu File

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

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

  <!-- Accessories submenu -->
  <Menu>
    <Name>Accessories</Name>
    <Directory>Accessories.directory</Directory>
    <Include>
      <And>
        <Category>Utility</Category>
        <Not>
          <Category>System</Category>
        </Not>
      </And>
    </Include>
  </Menu>
</Menu> <!-- End Accessories -->
```
In this example, the top level menu is named `Applications`, which is specified using the `<Name>` element. The `Applications` menu contains a single submenu, but several submenus are allowed. Each submenu can also have an `<Include>` element to perform a filter on the set of available desktop entries using matching rules.

For example, the `<Category>` element is a basic matching rule that selects a desktop entry file only if the `Categories` key contains the content of the `<Category>` element. In the example, the `Accessories` menu will include a desktop entry only if the menu contains the word “Utility” but not “System” in the `Categories` key. For more information about the `Categories` key, see “Working With Desktop Entry Files” on page 35.

The following table describes some of the elements in `.menu` files.

<table>
<thead>
<tr>
<th>Table 3-1</th>
<th>Menu Definition File Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Menu</td>
<td>Root element that may contain nested <code>&lt;Menu&gt;</code> elements. The nested elements define submenus. The way these elements are nested determines the menu structure.</td>
</tr>
<tr>
<td>Name</td>
<td>Specifies the name of the menu. Every <code>&lt;Menu&gt;</code> element must contain a <code>&lt;Name&gt;</code> element.</td>
</tr>
<tr>
<td>Directory</td>
<td>Specifies the name of the directory entry file that specifies the name, comment, and icon for the menu. If this element is not specified, then the <code>&lt;Name&gt;</code> element has to be used to display the menu name. By default, <code>.directory</code> files are searched in the <code>~XDG_DATA_DIRS/desktop-directories/</code> directory as defined in the XDG menu specification.</td>
</tr>
<tr>
<td>DefaultAppDirs</td>
<td>An instruction that indicates that all the available desktop entries from the <code>~XDG_DATA_DIRS/applications/</code> directory should be scanned. If this instruction is not included, then these locations are not scanned for desktop entries.</td>
</tr>
<tr>
<td>DefaultDirectoryDirs</td>
<td>An instruction that indicates that all the available directory entries from the <code>~XDG_DATA_DIRS/desktop-directories/</code> directory should be scanned. If the instruction is not included, then these locations are not scanned for directory entries.</td>
</tr>
</tbody>
</table>
### TABLE 3–1 Menu Definition File Elements (Continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include</td>
<td>Contains a list of matching rules by which the contents of a menu are generated. Might include the <code>&lt;Filename&gt;</code>, <code>&lt;Category&gt;</code>, <code>&lt;And&gt;</code>, <code>&lt;Or&gt;</code>, <code>&lt;Not&gt;</code>, or <code>&lt;All&gt;</code> matching rules.</td>
</tr>
<tr>
<td>Exclude</td>
<td>The opposite of the <code>&lt;Include&gt;</code> element. If any desktop entries are matched with this element, the entries are excluded from the previous set of included elements. For this reason, this element must appear after the <code>&lt;Include&gt;</code> element.</td>
</tr>
<tr>
<td>Filename</td>
<td>A matching rule that selects a desktop entry when the Desktop File-ID matches the contents of the <code>&lt;Filename&gt;</code> element.</td>
</tr>
<tr>
<td>Category</td>
<td>A matching rule that selects a desktop entry when the Categories key matches the contents of the <code>&lt;Category&gt;</code> element.</td>
</tr>
<tr>
<td>And</td>
<td>A matching rule that selects a desktop entry when the entry is selected by all nested matching rules in the <code>&lt;And&gt;</code> element.</td>
</tr>
<tr>
<td>Or</td>
<td>A matching rule that selects a desktop entry when the entry is selected by any nested matching rules in the <code>&lt;Or&gt;</code> element.</td>
</tr>
<tr>
<td>Not</td>
<td>A matching rule that does not select a desktop entry when the entry is selected by any nested matching rules in the <code>&lt;Not&gt;</code> element.</td>
</tr>
<tr>
<td>All</td>
<td>A matching rule that selects all desktop entries.</td>
</tr>
</tbody>
</table>

For a detailed description of the elements in the `.menu` files, see the XDG menu specification.

## Working With Directory Entry Files

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

To resolve the location of the `.directory` files, the system performs a search for default path in the following order:

1. Searches in the `$XDG_DATA_HOME/desktop-directories`. If the `$XDG_DATA_HOME` environment variable is not set, search in the `~/.local/share/desktop-directories` default directory.
2. Searches the subdirectory of each of the `desktop-directories` in the `$XDG_DATA_DIRS`. If the `$XDG_DATA_DIRS` environment variable is not set, search in the `/usr/share/desktop-directories` default directory.
   
   If several `.directory` files with the same name are found, the file that is found first takes precedence.

The following example shows a sample directory entry file.
The following table describes some of the important keys in directory entry files.

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

**Working With 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 execute, 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 must reside in the `$XDG_DATA_DIRS/applications` directory and must have a `.desktop` file extension.

The order in which the default paths are searched to resolve the location of `.desktop` files is as follows:

1. Search in the `$XDG_DATA_HOME/applications` directory. If the `$XDG_DATA_HOME` environment variable is not set, search in the `~/.local/share/applications` default directory.
2. Search the applications subdirectory of each directory in `$XDG_DATA_DIRS`. If the `$XDG_DATA_DIRS` environment variable is not set, search in the `/usr/share/applications` default directory.

If several `.desktop` files with the same name are found, the file that is found first takes precedence.

The following example shows a sample desktop entry file.
The following table describes some of the important keys in desktop entry files. For more information about desktop entry files, see the XDG Desktop Entry Specification.

**TABLE 3-3  Desktop Entry Keys**

<table>
<thead>
<tr>
<th>Desktop Entry Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categories</td>
<td>Specifies the keywords that describe the item. The keywords are separated with semicolons (;). For a list of the standard category keywords, see the desktop menu specification at <a href="http://www.freedesktop.org/Standards/menu-spec">http://www.freedesktop.org/Standards/menu-spec</a>. To find out how to map desktop entries to menus by using matching rules with the Categories key, see “Working With Menu Definition Files” on page 32.</td>
</tr>
<tr>
<td>Comment</td>
<td>Specifies a short description of the item. The comment is displayed as a tooltip when you point to the item in the menu.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Specifies the encoding of the desktop entry file.</td>
</tr>
<tr>
<td>Exec</td>
<td>Specifies a command to execute when you choose the item from the menu.</td>
</tr>
<tr>
<td>Icon</td>
<td>Specifies the filename of an icon that represents the item. Does not specify the path to the filename or the file extension.</td>
</tr>
<tr>
<td>MimeType</td>
<td>Specifies the MIME types that the application can handle.</td>
</tr>
<tr>
<td>Name</td>
<td>Specifies the name of the item. This name is displayed on the item in the menu.</td>
</tr>
<tr>
<td>NoDisplay</td>
<td>Specifies that an existing application should not be displayed in the menu.</td>
</tr>
<tr>
<td>Terminal</td>
<td>Specifies whether the command in the Exec key runs in a terminal window. If the value is true, the command runs in a terminal window. If the command does not open a window to run, the value of this key must be set to true.</td>
</tr>
<tr>
<td>Desktop Entry Key</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Type</td>
<td>Specifies the type of item. The value for the Type key is one of the following:</td>
</tr>
<tr>
<td></td>
<td>■ Application: An item that starts an application.</td>
</tr>
<tr>
<td></td>
<td>■ Link: An item that links to a file or folder, or a remote resource (such as an FTP site, or a web page).</td>
</tr>
<tr>
<td></td>
<td>■ FSDevice: An item that is a file system device.</td>
</tr>
<tr>
<td></td>
<td>■ Directory: An item that is a directory.</td>
</tr>
</tbody>
</table>

For more information about the keys in desktop entry files, see the [desktop entry specification](#).

**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.

## Editing Menus

The two types of menus are system menus and user menus.

### Adding and Modifying System Menus

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

**How to Add a Menu**

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

- Using the file manager
- Modifying the menu configuration files and menu data files

This task describes the file modification method.

1. **Create a directory entry file for the item that you want to add.**

2. **Place the directory entry file in the $XDG_DATA_DIRS/desktop-directories directory.**

For more information about directory entry files, see "Working With Directory Entry Files" on page 34.
3 In the $XDG_CONFIG_DIRS/menus/applications.menu file, add a <Menu> element for the new menu.
   For more information about .menu files, see “Working With Menu Definition Files” on page 32.

4 Create a <Name> element below the <Menu> element.
   The content of the element should contain the name for the menu.

5 Create a <Directory> element below the <Menu> element.
   The content of the element should contain the name of the directory entry file.
   The next time users log in, the menu should appear in the menu bar.

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

Next Steps
   To add an item to a menu, see “How to Add an Item to a Menu” on page 38.

▼ How to Add an Item to a Menu

1 Create a desktop entry file for the item that you want to add.
   For more information about desktop entry files, see “Working With Desktop Entry Files” on page 35.

2 Place the desktop entry file in the $XDG_DATA_DIRS/applications folder.

3 In the $XDG_CONFIG_DIRS/menus/applications.menu file, verify that the <Menu> element contains an <Include> element with a matching rule that selects the desktop entry file created in step 1.
   The next time users log in, the menu item is displayed in the assigned location.

▼ How to Edit the Properties of a Menu

1 In the $XDG_CONFIG_DIRS/menus/applications.menu file, find the <Menu> element that corresponds to the menu you want to modify.

2 Note the filename of the directory entry in the <Directory> element.

3 In the directory entry for this menu, modify the contents to change the properties of the menu.
   For more information about .directory files, see “Working With Directory Entry Files” on page 34.
How to Edit a Menu Item

1. Locate the desktop entry in the $XDG_DATA_DIRS/applications directory that corresponds to the menu item.

2. Edit the desktop entry to change the properties of the menu item.
   For more information on desktop entry files, see “Working With Desktop Entry Files” on page 35.

How to Delete an Item From a Menu

1. In the $XDG_CONFIG_DIRS/menus/applications.menu file, find the <Menu> element that contains the desktop entry you want to delete.

2. In the <Menu> element, insert an <Exclude> element after the closing tag for the <Include> element.

3. Insert the <Filename> matching rule as a subelement of the <Exclude> element to delete a desktop entry.
   The next time you log in, the menu item is not displayed in the menu.

Example 3–2 Deleting an Item From a Menu

The following example shows how to explicitly exclude the desktop entry for the dasher.desktop menu item from the applications.menu file.

```xml
<!-- ... -->
<Menu>
  <Name>Accessibility</Name>
  <Directory>Accessibility.directory</Directory>
  <Include>
    <And>
      <Category>Accessibility</Category>
      <Not><Category>Settings</Category></Not>
    </And>
  </Include>
  <Exclude>
    <Filename>dasher.desktop</Filename>
  </Exclude>
</Menu>
<!-- ... -->
```
Creating and Modifying User Menus

You can use the Main Menu preference tool to edit menus. Alternatively, you can manually create and edit a user menu file. For more information, see Oracle Solaris 11.1 Desktop User's Guide.

To manually create a custom menu for a user, the $XDG_CONFIG_HOME/menus/applications.menu file must exist. If the $XDG_CONFIG_HOME environment variable is not set, the default ~/.config directory is used. Because $XDG_CONFIG_HOME is the first location that is searched for the applications.menu file, the $XDG_CONFIG_HOME environment variable takes precedence over all other menu files.

User menus can contain all elements described in the “Working With Menu Definition Files” on page 32. For a complete list of the allowed elements, see the XDG menu specification.

Merging Menus

By default, user menu configuration files take precedence over system menu configuration files of the same name. For example, if an applications.menu file is present in the user's $XDG_CONFIG_HOME/menus directory, then that file will be used instead of the system applications.menu file to configure the Applications menu for that user. This replacement might result in the user not having access to all the menu items they would expect. To avoid this situation, a menu configuration file can specify that it should be merged with other menu configuration files, rather than replacing them. This process is described in the following sections.

Merging User and System Menus

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

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

Note – Older specifications did not include the type attribute and simply require the location of the menu file to be merged as the content of the <MergeFile> element. As a result, you might still see a location specified in the contents of <MergeFile>, even when type="parent".
The menu merging is performed as follows:

1. The children of the root `<Menu>` element in the merged menu file are substituted for the `<MergeFile>` element in the base menu file.

   **Note** – “Merged menu file” refers to the next `applications.menu` in the `$XDG_CONFIG_DIRS/menus` directory.

2. All child `<Menu>` elements with the same name are consolidated into a single `<Menu>` element by appending all child elements of each `<Menu>` element with the same name into the *last* occurrence of the menu element.

The following example shows a user menu file explicitly merging the system menu file.

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

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

This chapter includes the following information:

- “Themes Overview” on page 43
- “Theme Index File” on page 44
- “Installing a Controls Option” on page 45
- “Installing a Window Border Option” on page 45
- “Installing an Icons Option” on page 46
- “Installing Icons for Themes” on page 46
- “Creating a Custom Controls Option” on page 47

Themes Overview

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

A theme contains settings that affect different parts of the Oracle Solaris Desktop. The settings are described in the following table.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls</td>
<td>The controls setting for a theme determines the visual appearance of windows, panels, and panel applications. The controls setting also determines the visual appearance of the GNOME-compliant interface items that appear on windows, panels, and panel applications, such as menus, icons, and buttons. Some of the controls setting options that are available are designed for special accessibility needs. You can choose an option for the controls setting from the Controls tabbed section in the Theme preference tool.</td>
</tr>
</tbody>
</table>
Setting Description
---
Window border The window border setting for a theme determines the appearance of the borders around windows only. You can choose an option for the window border setting from the Window Border tabbed section in the Theme preference tool.
Icon The icon setting for a theme determines the appearance of the icons on panels and the desktop background. You can choose an option for the icon setting from the Icons tabbed section in the Theme preference tool.

---

### Theme Index File

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

The following example shows a sample theme index file.

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

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

The following table describes the keys in theme index files.

<table>
<thead>
<tr>
<th>Index File Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Specifies that this theme determines the appearance of several theme options such as controls, window border, and icons.</td>
</tr>
<tr>
<td>Name</td>
<td>Specifies the name of the theme that is displayed in the Theme preference tool.</td>
</tr>
<tr>
<td>Comment</td>
<td>Specifies the text that is displayed under the name of the theme in the Theme preference tool.</td>
</tr>
<tr>
<td>GtkTheme</td>
<td>Corresponds to the controls setting in the Theme Details dialog in the Theme preference tool. Specifies which controls setting option to apply to windows, panels, and panel applications.</td>
</tr>
</tbody>
</table>
TABLE 4–1 Keys in Theme Index Files (Continued)

<table>
<thead>
<tr>
<th>Index File Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IconTheme</td>
<td>Corresponds to the icons setting in the Theme Details dialog in the Theme preference tool. Specifies the icons setting option to apply to panels and the desktop background.</td>
</tr>
<tr>
<td>MetacityTheme</td>
<td>Corresponds to the window border setting in the Theme Details dialog in the Theme preference tool. Specifies the window border setting option to apply to windows.</td>
</tr>
<tr>
<td>ApplicationFont</td>
<td>Corresponds to the application font setting in the Font preference tool.</td>
</tr>
</tbody>
</table>

**Installing a Controls Option**

You can install a new option for the controls setting in the Theme preference tool. Controls options are in the /usr/share/themes directory.

The option file for a controls option is located at /usr/share/themes/option-name/gtk-2.0/gtkrc.

The image files for a controls option in the file system are located at /usr/share/themes/option-name/pixmaps/*. *

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

You can install your own options for the controls setting. If you install an option for the controls setting, the option is stored in the $HOME/.themes directory.

**Installing a Window Border Option**

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

The option file for a window border option is located at /usr/share/themes/option-name/metacity-1/metacity-theme-1.xml.

The image files for a controls option in the file system are located at /usr/share/themes/option-name/metacity-1/*. *

Typically, a new option for the window border setting is supplied as a .tar.gz file. To install the new window border option, unzip the .tar.gz file, then, untar the .tar file into the /usr/share/themes directory.
You can install your own options for the window border setting. If you install an option for the window border setting, the option is stored in the $HOME/.themes directory.

**Installing an Icons Option**

You can add a new option for the icons setting. Icons options reside in the /usr/share/icons/option-name directory.

The option file for an icons option is located at /usr/share/icons/option-name.

The image files for an icons option are located at /usr/share/icons/option-name/icons/*.*.

The ui-category directories are apps, devices, emblems, filesystems, or mimetypes.

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

You can install your own options for the icons setting. If you install an option for the icons setting, the option is stored in the $HOME/.icons/option-name directory.

**Installing Icons for Themes**

When you install a new icon for an application, you must create several versions of the icon so that the icon displays correctly in the themes.

You must create several versions of the following types of icon:

- Icons that are used within applications in the Oracle Solaris Desktop
- Icons that are used internally by GTK+ applications or GTK+ stock icons

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 × 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
- High contrast icon
- Inverse high contrast icon

If possible, you should also create 16 × 16 pixel versions of each of the listed icons for themes that do not require large print.

The Oracle Solaris Desktop provides several themes that are designed for users with special visual needs. For example, some of the themes are designed for users with low vision.
For more information about how to create icons for application launchers and for panels, see http://www.freedesktop.org/Standards/icon-theme-spec.

▼ How to Install an Icon for a Theme

1 Install the icons to the image files location that is specified for the theme in “Installing a Controls Option” on page 45 or “Installing a Window Border Option” on page 45.  
   For example, to add icons to the HighContrastLargePrint theme, add the icons to the /usr/share/themes/HighContrastLargePrint/pixmaps directory.

2 Add references to the icons to the relevant theme files.  
   For example, to add icons to the HighContrastLargePrint theme, add references to the icons to the /usr/share/themes/HighContrastLargePrint/gtk-2.0/gtkrc file.

3 Modify the gtkrc file for the theme to associate the icon with a GTK+ stock icon identifier.

Creating a Custom Controls Option

If the options for the controls setting are not suitable, you can create a custom controls option.

▼ How to Create a Custom Controls Option

1 Create a directory structure for the option in the /usr/share/themes directory.  
   Use the same directory structure that other options use.  
   For example, to create an option that is called SmallPrint, create the following directories:
   - /usr/share/themes/SmallPrint
   - /usr/share/themes/SmallPrint/gtk-2.0

2 Locate the closest gtkrc file that suits your requirement.

3 Copy the file to the gtk-2.0 directory of the new option.

4 Modify the attributes of the interface elements in the gtkrc file as required.

5 If the new option includes images, install the images for the new option in the pixmaps directory of the new option.
Note – If the new option uses images only from another option, you do not need to create copies of the images for the new option. Ensure that the reference to the images in the pixmap_path entry in the gtkrc file is correct.

Users can now choose the new option for the controls setting.
Applications in the Oracle Solaris Desktop use one of the following font systems to find and use fonts for drawing text on the screen and printouts:

- fontconfig Library
- Legacy X11 Font System

Each application is designed to use one of these systems. Few applications in the Oracle Solaris Desktop support both systems or switch between them.

This chapter describes the available font systems in the Oracle Solaris Desktop in detail. It also describes how to customize fonts in the Oracle Solaris Desktop.

This chapter includes the following information:

- “Fonts Overview” on page 49
- “Substituting Fonts” on page 52

Fonts Overview

This section describes the fontconfig library and explains how to add fonts to it. It also describes the legacy X11 font system and how to use it to add bitmap fonts.

fontconfig Library

The Oracle Solaris Desktop uses the fontconfig configuration and customization library. The fontconfig library can use all types of fonts, including PostScript Type 1 fonts and TrueType fonts.

Many applications in the Oracle Solaris Desktop, including those that are part of the GNOME Desktop Environment, use the fontconfig system for finding fonts. The fontconfig library provides a list of all the fonts available on the Oracle Solaris Desktop. To compile this list, fontconfig searches the directories listed in the /etc/fonts/fonts.conf file.
Installing fonts to be accessed by the fontconfig library is as simple as copying them to the "HOME/ .fonts" subdirectory. Fonts accessed through this library are addressed with simple names such as DejaVu Sans or Liberation Mono.

The fc-list command provides a list of all fonts known to this library. You can specify specific sizes or styles by using the patterns described in the fonts.conf(4) man page. For example, to start an xterm using the bold variant of the DejaVu Mono font at a 12-point size, you would type the following command:

```
$ xterm -fa "DejaVu Mono-12:style=Bold"
```

For more information about the fontconfig library, see http://freedesktop.org/software/fontconfig.

For information about adjusting font configuration, see “Adjusting Font Configuration” in International Language Environments Guide for Oracle Solaris 11.1.

## Adding Fonts

This section describes how to add fonts for all users or for an individual user.

### How to Add a Font for All Users

1. Copy the font file to one of the directories in the /etc/fonts/fonts.conf file. Typically, fonts are stored in the /usr/share/fonts/ directory. The fontconfig library updates the list of fonts automatically.

2. If the list of fonts is not updated, type the following command:
   ```
   # fc-cache directory-name
   ```

### How to Add a Font for an Individual User

1. Copy the font file to the $HOME/.fonts directory of the user. The fontconfig library updates the list of fonts automatically.

2. If the list of fonts is not updated, type the following command:
   ```
   # fc-cache directory-name
   ```

## Legacy X11 Font System

Some applications still use the original X Window System font mechanisms. They allow less styling choices, have a more complex font naming scheme, and do not include anti-aliasing or LCD font smoothing. For more information about the X11 font naming scheme, see the X Logical Font Description specification. Some of the X11 font commands include:
Fonts Overview

- `xlsfonts` – Provides a list of all fonts known to the system
- `xfontsel` – Provides a simple font selection application for the fonts known to the system

You can specify specific styles and sizes by using fields in the X Logical Font Description (XLFD) name. For example, to start an `xterm` using the bold variant of the DejaVu Mono font at a 12-point size, you would type the following command:

```
$ xterm -fn '-misc-dejavu sans mono-bold-r-normal--12-120-72-72-m-*-iso10646-1'
```

To install fonts by using the legacy X11 font system, you must create metadata files with the `mkfontdir` or `mkfontscale` commands and add the directory to the X server font path with the `xset` command. The font path changes made with the `xset` command are reset to the default setting for every new session.

Note – Font paths can be permanently added to the default X11 font path for all sessions by adding links to the font directories in the `/etc/X11/fontpath.d` file. For more information, see the `FONTS` and `FONTPATH.D` sections in the `Xorg` manpage.

Because the `/etc/X11/fontpath.d` directories are included in the default `fonts.conf` files in the Oracle Solaris OS, fonts added in this way are automatically available to the applications by using the fontconfig library.

For more information about the legacy X11 font system and installing the bitmap and scalable fonts, see [http://www.x.org/releases/X11R7.6/doc/xorg-docs/fonts/fonts.html](http://www.x.org/releases/X11R7.6/doc/xorg-docs/fonts/fonts.html).

▼ How to Install Bitmap Fonts by Using the Legacy X11 Font System

When installing fonts, you first need to create a font directory that contains all the relevant font files as well as some index files. You then need to inform the X server of the existence of this new directory by including it in the font path.

1 **(Optional) Convert BDF format font files to the PCF format.**

   Although bitmap fonts are normally distributed in the BDF format, the binary PCF format is more efficient.

   a. **Use the `bftopcf` command to convert BDF format files.**

      For example, to convert a `courier12.bdf` file, you would type:

      ```
      $ bftopcf courier12.bdf
      ```

   b. **Compress the resulting PCF format files.**

      For example:
      ```
      $ gzip courier12.pdf
      ```
2 Copy all font files that you want to make available into a directory.
   For example, to use the /usr/local/share/fonts/bitmap/ directory:
   $ mkdir /usr/local/share/fonts/bitmap/
   $ cp *.pcf.gz /usr/local/share/fonts/bitmap/

3 Create the fonts.dir index file.
   $ mkfontdir /usr/local/share/fonts/bitmap/

4 Set the font path to let the X server know about the new font directory.
   ■ To set the font path for the current session only, use the fp option.
     Putting a plus sign (+) before the option adds the directory to the beginning of the font path.
     Putting it after the option adds the directory to the end of the font path.
     For example:
     $ xset +fp /usr/local/fonts/Type1
     $ xset fp+ /usr/local/fonts/bitmap
     For more information, see the xset man page.

   ■ To set the font path permanently, specify it in the X server’s xorg.conf file.
     The path is computed by appending all the directories mentioned in the FontPath entries of
     the Files section of the file in the order in which they appear
     For example:
     FontPath "/usr/local/fonts/Type1"
     ...FontPath "/usr/local/fonts/bitmap"
     For more information see the xorg.conf man page.

Substituting Fonts

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

If the system needs to display a character that is not present in the selected font, fontconfig
attempts to display the character in a similar font. For example, you might select Bitstream Vera
Sans as the font for the Text Editor application. The Bitstream Vera font family does not include
Cyrillic characters. If you open a document which contains a Cyrillic character, Text Editor uses
a similar font that includes Cyrillic characters to display the character.
The fontconfig library also defines aliases for fonts, for example, serif, sans-serif, and monospace. When you select one of the aliases for a font, the system attempts to use the first font that is defined for that alias in the /etc/fonts/fonts.conf file.
This chapter provides a general overview of the MIME system and describes how applications detect MIME types, how to register MIME types, and how to add applications to the desktop. This chapter also describes how to configure different parts of the MIME database.

This chapter includes the following information:

- “MIME Types Overview” on page 55
- “About the MIME Database” on page 56
- “Understanding MIME Type XML Files” on page 59
- “Modifying MIME Types” on page 61
- "Registering Applications for MIME Types” on page 63

For the detailed information about MIME types, see the XDG shared mime info specification.

**MIME Types Overview**

A Multipurpose Internet Mail Extension (MIME) type identifies the format of a file. Applications such as Internet browsers and email applications use the MIME type of a file to determine the actions to perform on the file.

For example, 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

MIME types were originally proposed as a standard for identifying the message body of an email message. Many systems use MIME types to identify the format of arbitrary files on the file system.
MIME types are composed of a top-level media type followed by a subtype identifier, separated by a forward slash character (/). An example of a MIME type is image/jpeg. The media type in this example is image and the subtype identifier is jpeg. The top-level media type is a general categorization about the content of the file, while the subtype identifier specifically identifies the format of the file. For more information about the supported media types and the corresponding subtypes, see MIME Media Types at the IANA website.

The implementation of MIME types in the Oracle Solaris Desktop follows the XDG shared mime info specification. This specification provides the following advantages:

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

The following sections in this chapter describe the files and directories that make up the MIME database, details about source XML files, how to create or modify MIME types, how to register applications as handlers for certain MIME types, and how to add an application to the Oracle Solaris Desktop.

### About the MIME Database

The MIME database is a collection of files that include the following information:

- The set of known MIME types
- The method for determining the MIME type of a file
- Meta information regarding a MIME type

The MIME database is created from the set of files located in the $XDG_DATA_HOME/mime and $XDG_DATA_DIRS/mime directories. If the environment variables are not set, then the default values are ~/.local/share and /usr/local/share:/usr/share respectively.

This guide uses <MIME> to refer collectively to these directories. If conflicting information for the same MIME type is found, the information in the file that is found first takes precedence.

For example, assuming default paths for the environment variables, an instruction to load the <MIME>/text/plain.xml file, loads the following files:

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

**Note** – The XDG shared mime info specification was drafted by the X Desktop Group, and the specification makes use of the XDG base directory specification.
MIME Database Contents

The MIME database contains the following directories and files:

- <MIME>/packages/
- <MIME>/media/subtype.xml
- <MIME>/globs
- <MIME>/magic
- <MIME>/XMLnamespaces
- <MIME>/aliases
- <MIME>/subclasses

The following is a list of MIME directories and files in the MIME database with a brief description:

- <MIME>/packages/ – This directory can contain any number of XML files. Each XML file describes the collection of MIME types. The freedesktop.org.xml file located in the <MIME>/packages/ directory contains all the default MIME types that are widely used and recognized. By default, the freedesktop.org.xml file is installed in the /usr/share/mime/packages directory. Applications that provide information about new MIME types install a single new XML file in this directory. Depending on the prefix where the application is installed, the file is created in the /mime/package subdirectory. For example, an application installed in the /usr/bin directory should install a new source XML file in the /usr/share/mime/packages directory. For more information about the XML files in the packages directory, see "Understanding MIME Type XML Files" on page 59.

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

The following example is an excerpt from the file generated by the update-mime-database application, using the freedesktop.org.xml default source XML file.

```xml
<?xml version='1.0' encoding='utf-8'?>
mime-type
xmlns="http://www.freedesktop.org/standards/shared-mime-info"
type="text/plain">
<!--Created automatically by update-mime-database.
DO NOT EDIT!-->
  <comment>
    plain text document
  </comment>
  <!-- possibly more translations-->
```
About the MIME Database

- **<MIME>/globs** – Each line in this file contains a MIME type and a glob pattern, separated by a colon. The files that match the glob pattern are resolved to the MIME type specified before the colon. There are special rules about how filenames are matched by the glob pattern. For more information, see the XDG shared mime specification.

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

- **<MIME>/magic** – This is a binary file that contains information on how to resolve MIME types by content sniffing (known as "magic rules"). A magic rule is a set of one or more rules that define a file's MIME type by specifying text or binary data to search for at the beginning of the file. For example, you can set a rule to check for the `%PDF-` string at byte offset 0 in the file. If the string is found, you can assign the file to the `application/pdf` MIME type.

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

- **<MIME>/XMLnamespaces** – This file contains a mapping of XML namespaces to MIME types. Each line contains three fields:
  - namespace
  - localName
  - MIME type

  Each field is separated by a space. If the `localName` is empty, then there are two spaces between the namespace and the MIME type fields.

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

- **<MIME>/aliases** – This file contains a list of aliases for each MIME type. An alias is another type of a MIME type. Each line in this file contains two fields:
  - alias name
  - MIME type

  The fields are separated by a space. This file is also generated by the `update-mime-database` application.

- **<MIME>/subclasses** – This file contains a list of subclassed MIME types and their parent MIME type. According to the XDG shared mime specification, a type is a subclass of another type if any instance of the first type is also an instance of the second type. For example, all
image/svg files are also text/xml, text/plain, and application/octet-stream files. Subclassing is about the format, rather than the category of the data. For example, all spreadsheets do not inherit from a generic spreadsheet class.

The format of this file is similar to the aliases file. Each line contains two fields:

- subclassed MIME type
- parent MIME type

Each field is separated by a space. This file is also generated by the update-mime-database application.

**Refreshing the MIME Database**

To add new MIME types to the system or to modify information about a MIME type, you need to understand how to refresh the MIME database. To refresh the MIME database, use the update-mime-database application. For example, if an application installs information about a new MIME type to the /usr/share/mime/packages/diff.xml file, the update-mime-database application must be invoked with the /usr/share/mime parameter.

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

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

**Understanding MIME Type XML Files**

The MIME XML files provide all the information regarding MIME types that are installed into the MIME database by the update-mime-database application. The MIME XML files are located in the <MIME>/packages directory. A few rules about the MIME XML files:

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

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

The following table gives a brief description of each element that can occur as children to the mime-type element.
### TABLE 6–1  Child elements of `<mime-type>`

<table>
<thead>
<tr>
<th>Element (and attributes)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;glob pattern=&quot;*.xyz&quot;&gt;</code></td>
<td>This element specifies a glob pattern for the filenames. If the filename matches, then the file is assigned the MIME type of the parent <code>&lt;mime-type&gt;</code> element. The <code>pattern</code> attribute is mandatory.</td>
</tr>
</tbody>
</table>
| `<magic priority="50">` | This element contains a list of match elements as its children. The `priority` attribute is optional, and specifies a priority between 0 and 100, with 100 being the highest matching priority. Each child match element has three required attributes and an optional attribute:  
  - `type`
  - `offset`
  - `value`
  - `mask` (optional attribute)  
  For details about these attributes, see the [XDG shared mime info specification](http://www.freedesktop.org/standards/shared-mime-info). |
| `<alias type="media/subtype">` | This element defines an alias for the parent `<mime-type>` element. For example, `application/x-pdf` is an alias for the `application/pdf` MIME type. |
| `<sub-class-of type="media/subtype">` | This element defines the parent `<mime-type>` element as a subclass of the MIME type specified in the `type` attribute. For example, `image/svg` is a subclass of the `text/xml`, `text/plain`, and `application/octet-stream` MIME types. |
| `<comment xml:lang="locale">` | This element provides a readable description for the MIME type. There can be zero or more occurrences of this element as long as each one contains a unique value for the `xml:lang` attribute. |
| `<root-XML namespaceURI="namespace" localName="">` | If a file is determined to be an XML file, then this element helps to further classify the file type by using the `namespaceURI` and `localName` attributes. The `namespaceURI` attribute is the namespace of the document, and `localName` is the name of the root element for the document. If `localName` is present but the value is empty, then the root element might have any name but the namespace must still match. |

The following example defines the `text/x-diff` MIME type.

**EXAMPLE 6–1**  Example of a `diff.xml` source XML file:

```xml
<?xml version='1.0'?>
<mime-info xmlns='http://www.freedesktop.org/standards/shared-mime-info'>
```
EXAMPLE 6-1 Example of a `diff.xml` source XML file: (Continued)

```xml
<mime-type type="text/x-diff">
    <comment>Differences between files</comment>
    <comment xml:lang="af">verskille tussen lêers</comment>
    <!-- more translated comment elements -->
    <magic priority="50">
        <match type="string" offset="0" value="diff\t"/>
        <match type="string" offset="0" value="***\t"/>
        <match type="string" offset="0" value="Common subdirectories: "/>
    </magic>
    <glob pattern="*.diff"/>
    <glob pattern="*.patch"/>
</mime-type>
</mime-info>
```

In this example, multiple comment elements give a readable name to the MIME type in a number of different languages. The text/x-diff MIME type has rules for matching both through glob patterns and through the use of content sniffing (known as magic rules). Any file with the .diff or .patch extension will resolve to this MIME type. Additionally, any file whose contents start with the strings specified in the value attributes of the match element will resolve to the text/x-diff MIME type. For more details about the glob patterns and magic rules, see the XDG shared mime info specification.

Modifying MIME Types

You should never directly modify the source XML files that are installed by applications in the `<MIME>/packages` directory. Instead, modify the `Overrides.xml` file. This file has precedence over all other source XML files installed into the same packages directory. If you are an application author, then this rule does not apply. Application authors should create a new source XML file and place the file in the `<MIME>/packages` directory.

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

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

▼ How to Add or Modify MIME Types

1. Create the `Overrides.xml` source XML file containing the definitions for the MIME types.
   - To set the definitions for all users, put the file in the `/usr/share/mime/packages` directory.
To set the definitions for an individual user, put the file in the
~/.local/share/mime/packages directory.

If the file already exists, open it.

2 Update the MIME database by running the update-mime-database command.

   - If you are setting the definitions for all users:
     
     ```
     # update-mime-database /usr/share/mime
     ```

   - If you are setting the definitions for an individual user:
     
     ```
     # update-mime-database ~/.local/share/mime/packages
     ```

3 Update the MIME database by running the update-mime-database command.

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

4 Use the gnomevfs-info command to verify your changes.

   For example, the gnomevfs-info command displays the following output when you run the
   command for a SVG file. Note that the default application for this MIME type is eog.desktop.

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

   $```

   For more information about default applications, see “Registering Applications for MIME
   Types” on page 63.
Creating an application/x-newtype MIME Type

1. Create a new file, test.xyz, in your home directory.
2. Use the gnomevfs-info command to find the file’s MIME type.

```
$ gnomevfs-info test.xyz
```

The MIME type for this file should be detected as text/plain because no glob patterns or magic rules match the file.

Note – When no glob patterns or magic rules match a file, then the file is resolved to the text/plain MIME type if the file contains textual data or application/octet-stream for binary data. If the file is empty, then the type is identified as text/plain MIME type.

3. Create the Overrides.xml file or if the file already exists, modify the file.

The sample XML file is as follows:

```xml
<?xml version='1.0' encoding='utf-8'?>
<mime-info xmlns="http://www.freedesktop.org/standards/shared-mime-info">
  <mime-type type="application/x-newtype">
    <comment>new mime type</comment>
    <glob pattern="*.xyz"/>
  </mime-type>
</mime-info>
```

4. Update the MIME database by using the update-mime-database command.

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

5. Use the gnomevfs-info command to verify that your change has taken effect.

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

You should see that the MIME type for the testing.xyz file is resolved as application/x-newtype.

Registering Applications for MIME Types

Applications are registered by creating a MimeType key in the .desktop entry file and listing each MIME type separated by a semicolon. The MimeType key should only be used in the .desktop files whose Type key has Application as value. For more information about .desktop files, see “How to Add or Modify MIME Types” on page 61.
How to Register Applications for MIME Types

1. Create a .desktop file for the application in the $XDG_DATA_HOME/applications or $XDG_DATA_DIRS/applications directory.
   The .desktop file must include a MimeType key whose value comprises each MIME type to be registered, separated by semicolons. For example:
   MimeType=text/html;text/css;text/x-javascript

   Note – The MimeType key should only be used in the .desktop files whose Type key has Application as the value. For more information about .desktop files, see “Working With Desktop Entry Files” on page 35.

2. Update the application database.
   # update-desktop-database
   This command creates or updates the mimeinfo.cache file in the same directory as the .desktop file. This cache file is used to facilitate MIME type searches.

3. Create or update the defaults.list file in the same directory as the .desktop file.
   The defaults.list file is a plain text file that specifies the default application to use for specific MIME types. Each line in the file consists of the MIME type, the = symbol and the Desktop File ID, which is the filename for the desktop entry file. For more information, see Example 6–3.

   Note – Registering MIME types for applications is part of the XDG desktop entry specification rather than the shared mime info specification. For more information, see http://www.freedesktop.org/wiki/Standards/desktop-entry-spec.

Example 6–3  A User's defaults.list File

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

How to Associate a MIME Type With an Application

1. Add a menu item for the application.
   For more information about how to add an item to a menu, see “How to Add a Menu” on page 37.
2 Add an icon for the application to the `/usr/share/icons/theme-name/icon-size/apps` directory.
   For more information about installing icons and themes, see “Installing Icons for Themes” on page 46.

3 If the application uses a new MIME type, perform the following steps:
   a. Add a source XML file to the MIME database.
      For more information, see “Modifying MIME Types” on page 61.
   b. Add an icon for the MIME type to `/usr/share/icons/theme-name/icon-size/mimetypes`.
      For more information on icons and themes, see Chapter 4, “Installing Themes.”

4 To associate the application with a MIME type, include a `MimeType` key in your `.desktop` file.
   For more information, see “Registering Applications for MIME Types” on page 63.
Managing 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 Oracle Solaris Desktop is XScreenSaver. The following sections describe how to set preferences for the XScreenSaver application, and how to modify displays that are available for the screensaver.

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

This chapter includes the following information:

- “Setting Screensaver Preferences” on page 67
- “Modifying Look and Feel of Your Screensaver” on page 68

Setting Screensaver Preferences

The default screensaver preferences are stored in the /usr/share/X11/app-defaults/XScreenSaver file. To start the Screensaver preference tool, run the xscreensaver-demo command from the /usr/bin/ directory. To start the Screensaver preference tool from System menu, choose System → Preferences → Screensaver.

Setting Default Screensaver Preferences for All Users

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

Restoring Default Screensaver Settings for a User

To restore the default settings for a user, delete the $HOME/.xscreensaver file from the user’s home directory. If no $HOME/.xscreensaver file is present, the system uses the default preferences in the XScreenSaver file.
Note – The default display behavior of the XScreenSaver application is to display a blank screen.

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

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

Modifying Screensaver Preferences

To modify screensaver application preferences, you can use the Screensaver preference tool. When you modify the screensaver preferences, the preferences are stored in the user’s home directory, in the $HOME/.xscreensaver file.

Modifying Look and Feel of Your Screensaver

The screensaver displays are listed in the XScreenSaver file and in the $HOME/.xscreensaver file. The screensaver application allows users to choose one or more screensaver displays.

▼ How to Add a Screensaver Display

1. Copy the executable file for the display to the /usr/lib/xscreensaver/hacks/ directory.

2. Add the command for the screensaver display to the XScreenSaver file or the $HOME/.xscreensaver file.

   To run the screensaver display on the whole screen rather than in a window, include appropriate arguments. For example, you might want to include the -root option to display the screensaver display on the whole screen.

Disabling a Screensaver Display

To disable the screensaver displays for all users on a Sun Ray client, type the following command:

```
# pkg uninstall 'desktop/xscreensaver/hacks*'
```
Note – The Pluggable Authentication Modules (PAM) service name for the XScreenSaver application is dtsession. This name is used for compatibility with previous applications.
A session is the period of time that you spend while using the desktop between logging in and logging out. During a session, you might use your applications, and print, or browse the web. Logging in to the desktop begins your session.

- “Session Manager Overview” on page 71
- “Setting Session Defaults” on page 71

### Session Manager Overview

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. The following applications run in a session:

- Session manager, `gnome-session`
- GConfX settings daemon, `gnome-settings-daemon`
- `gnome-panel` application
- `metacity` or `compiz` window manager, depending on graphics card capability

### Setting Session Defaults

Certain applications might be configured to start automatically when you log in. Applications to be started are specified by placing a `.desktop` file for each application in the appropriate directory, as defined in the Freedesktop Application Autostart Specification. When you log in, the system searches the following directories for the `.desktop` files in the following order:

1. The `~/.config/gnome-session/saved-session` directory.
2. The `$XDG_CONFIG_HOME/autostart` directory. If the `$XDG_CONFIG_HOME` environment variable is not set, searches in the `~/.config/autostart` default directory.
3. The autostart subdirectory of each directory in the $XDG_CONFIG_DIRS. If the $XDG_CONFIG_DIRS environment variable is not set, searches in the /usr/share/gnome/autostart:/etc/xdg/autostart default directory list.

If conflicting .desktop files for the same application are found, the .desktop file that is found first takes precedence.

The .desktop file of an application must have the format defined in the “Working With Desktop Entry Files” on page 35. All keys should be interpreted as defined, with the following exceptions. These exceptions take into account that the .desktop files in an autostart directory are not shown in a menu.

- **Hidden key** – If the Hidden key is set in the .desktop file, then the application described in the .desktop file will not be shown on the Applications menu, even though it is installed. The Hidden key is a quick way to remove infrequently used applications from the menu, without deleting the .desktop file or uninstalling the application.

  When the .desktop file has the Hidden key set to true, the .desktop file must be ignored. When multiple .desktop files with the same name exist in multiple directories, then only the Hidden key in the .desktop file that is found first must be considered. If that Hidden key is set to true, all .desktop files with the same name in the other directories are also ignored.

- **OnlyShowIn and NotShowIn keys** – The OnlyShowIn entry contains a list of strings identifying the desktop environments that will autostart this application. All other desktop environments will not autostart this application. For example, OnlyShowIn=GNOME;KDE.

  The NotShowIn entry contains a list of strings that must not autostart this application. All other desktop applications must autostart this application.

---

**Note** – Only one of these keys, either OnlyShowIn or NotShowIn, may appear in a single .desktop file.

- **TryExec key** – The value of the TryExec key must match an installed executable program, or the program will not autostart. The value of the TryExec field may either be an absolute path or the name of an executable without any path components. If the name of an executable is specified without any path components, the $PATH environment is searched to find a matching executable program.

To set default session applications for all users, place the appropriate .desktop files in the /etc/xdg/autostart directory.
Restoring the Default Session Settings

To restore the default session settings for a user, delete the following directories:

- ~/.config/gnome-session/saved-session
- $XDG_CONFIG_HOME/autostart
- ~/.config/autostart (if $XDG_CONFIG_HOME is not set)

Saving the Current Session as the Default Session

To save the current session as the default session for a user, perform one of the following steps:

- Use the `gnome-session-save` command.
  
  `$ gnome-session-save`

- In the Preferences menu, choose Startup Applications. Select the Remember Currently Running Applications option in the Options tab.
  
  A .desktop file in the ~/config/gnome-session/saved-session directory for each currently running application is created.
Overview of the Yelp Help Browser

The Oracle Solaris Desktop displays help in the Yelp help browser. For more information, see “Overview of the Yelp Help Browser” in Oracle Solaris 11.1 Desktop User’s Guide.

This chapter describes the Yelp help browser in the Oracle Solaris Desktop system. It includes the following information:

- “Online Help Source Documents” on page 75
- “Open Source Metadata Framework Files” on page 75
- “Rarian Cataloging System” on page 76

Online Help Source Documents

The source documents for the online help are XML files. The XML files are written in the DocBook XML Version 4.1.2 document type definition (DTD). The DocBook XML files are converted to HTML, using stylesheets installed with the gnome-doc-utils utility and the HTML is displayed in the help browser. For more information about DocBook XML, see http://www.oasis-open.org/docbook/xml/.

The help system uses Open Source Metadata Framework (OMF) files, and a cataloging system.

Open Source Metadata Framework Files

The XML file for each help system manual has an associated Open Source Metadata Framework (OMF) file. The OMF file contains information about the manual that is used by the help browser. OMF files have a .omf extension.
When you install an application, the OMF file is copied to the
/usr/share/omf/application-name directory. If a help document has an associated OMF file, users can access the help document from the help browser. OMF files contain the following information about help documents:

- Location of the XML file for the help document
- Title of the help document
- Subject category to which the help document belongs

Rarian Cataloging System

Rarian is a document cataloging system. The primary function of Rarian is to manage the information in OMF files for the help browser. When you install an application, the OMF file is copied to the /usr/share/omf/application-name directory.

Rarian contains a hierarchy of subject categories to which a help document can belong. The subject category of the document in the OMF file determines the location of the document in the table of contents presented by the help browser.

The hierarchy of subject categories is defined in the /usr/share/librarian/Template/lang/scrollkeeper_cl.xml file. The subject category of a document must correspond to a category specified in this file. For more information about Rarian, see the documentation installed locally at /usr/share/librarian/. Also, see the community site at http://rarian.freedesktop.org/.

Note – If the help XML file is moved to a new location, then the location in the OMF file needs to be updated.
Improving the Performance of the Oracle Solaris Desktop System

This chapter describes how to improve the performance of the Oracle Solaris Desktop system. This chapter lists several preferences whose settings can be changed to improve the performance of the Oracle Solaris Desktop system. You can use the `gconf-tool` command to set values for user preferences. The example commands in this chapter show how to set values in the user configuration source.

For information about the `gconf-tool` command and the options that are available with the command, see Chapter 2, “Managing User Preferences With GConf.”

For information about optimizing the desktop environment, see *Optimizing the Oracle Solaris Desktop for a Multi-User Environment*.

This chapter includes the following information:

- “Reducing CPU Usage” on page 77
- “Reducing X Window System Network Traffic” on page 82
- “Reducing Color Usage and Improving Display Quality” on page 82

### Reducing CPU Usage

This section describes preferences that you can set to reduce CPU usage by the Oracle Solaris Desktop system. These preferences are described in the following sections:

- “Using Theme Options” on page 78
- “Turning Off the Display of Icons in Menus” on page 78
- “Turning Off Panel Animation” on page 79
- “Using a Solid Color for the Desktop Background” on page 79
- “Improving File Manager Performance” on page 79
- “Using the Appropriate Window Manager” on page 81
Using Theme Options

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

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

- Atlanta
- Esco
- AgingGorilla
- Bright
- Metabox

*Note* – Metabox does not work well with inverse controls options such as HighContrastInverse. Use Atlanta with inverse controls options.

To change the window border theme option, type the following command:

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

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

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

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

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

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

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

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

Turning Off the Display of Icons in Menus

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

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

## Turning Off Panel Animation

Panels can be shown or hidden in an animated style. Panel animation can be enabled or disabled by using the Properties dialog.

If the panel has hide buttons at both ends, then clicking one of the hide buttons will cause it to slide in or out of view. If the panel is set to Autohide, then the panel slides into view when you move your mouse to the edge of the screen, and slide out of view again when you move the mouse away. If panel animations are disabled, then there will be no sliding effect. Panels just abruptly appear or disappear from view.

To turn off panel animation, type the following command:

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

## Using a Solid Color for the Desktop Background

Using a solid color for the desktop background reduces the number of colors used by the Oracle Solaris Desktop system.

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

```bash
# 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, you can use the Background preference tool to choose a solid color for the background.

## Improving File Manager Performance

The file manager includes performance-related preferences. Each performance preference can take one of the three values described in the following table.
Value Description
always Performs the action for both local files and files on other file systems.
local_only Performs the action for local files only. When you set a performance preference to local_only, the CPU usage is reduced.
ever Never performs the action. When you set a performance preference to never, the CPU usage and the network traffic are reduced.

To set the performance preference, use the following command:

```
# gconftool-2 --type string \n--set /apps/nautilus/preferences/show_icon_text value
```

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

<table>
<thead>
<tr>
<th>Preference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show_icon_text</td>
<td>Preview the content of text files in the icon that represents the file.</td>
</tr>
<tr>
<td>show_directory_item_counts</td>
<td>Show the number of items in folders.</td>
</tr>
<tr>
<td>show_image_thumbnails</td>
<td>Show thumbnails of image files.</td>
</tr>
<tr>
<td>preview_sound</td>
<td>Preview the content of sound files.</td>
</tr>
</tbody>
</table>

**Note** – You can also set performance preferences for the file manager by performing the following steps:

1. Choose Edit → Preferences from a file manager window.
2. Select Preview.
3. Select the preference you want to set.
   
   For example, to set the show_image_thumbnails preference, select the Show Thumbnails preference option.

**Turning Off the Side Pane**

To improve file manager performance, turn off the side pane by typing the following command:

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

**Turning Off the Toolbar**

To improve file manager performance, turn off the toolbar by typing the following command:
# gconftool-2 --type bool \
--set /apps/nautilus/preferences/start_with_toolbar false

**Turning Off the Location Bar**
To improve file manager performance, turn off the location bar by typing the following command:

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

**Tip** – You can use the Ctrl + L keyboard shortcut to display a location bar when required.

**Locking the Desktop Background and Hiding Desktop Icons**
The file manager contains a preference that enables users to lock the desktop background and hide desktop icons. To improve performance, disable the desktop background and hide desktop icons by typing the following command:

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

Any existing colors or wallpapers that were applied to your desktop background remain unchanged. For example, if you had a solid green background before you turn this option off, you will still have a solid green background later. However, once the desktop background is turned off, you cannot make further changes to background colors or wallpapers until you turn on this option back again.

Note that if you disable the desktop background, you cannot do the following:
- Use the file manager to change the pattern or color of the desktop background.
- Use desktop objects, such as the Trash. The desktop objects are not displayed on the desktop.

**Using the Appropriate Window Manager**

By default, the Oracle Solaris Desktop selects one of two available window managers when you log in, depending on the capabilities of your graphics hardware:
- **compiz** – A compositing window manager that supports advanced graphical effects such as shading and translucency. When you log in, the system automatically checks whether your graphics card supports hardware acceleration with Oracle Solaris 11.
- **metacity** – A basic standards-compliant window manager. When you log in, the system automatically checks whether your graphics card supports hardware acceleration with Oracle Solaris 11.
To minimize CPU usage, you should manually select the Metacity Window Manager. You can do this in one of the following ways:

- Use the `metacity` command with the `--replace` option.
  ```bash
  $ metacity --replace &
  ```
- In the Visual Effects tab of the Appearance Preference tool, select None.
  This selection takes effect immediately, and is in effect the next time you log in.

# Reducing X Window System Network Traffic

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

- Using theme options – 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 that uses solid colors, use one of the following window border options:

- Atlanta
- Esco

For information, see “Using Theme Options” on page 78.

- Turning off the 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 when the panels are displayed on a remote host.

For information, see “Turning Off the Display of Icons in Menus” on page 78.

For information about the X Window System, see Chapter 12, “Working With the X Window System.”

# Reducing Color Usage and Improving Display Quality

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

The Oracle Solaris Desktop system uses the websafe color palette. This palette is a general-purpose palette of 216 colors that is designed to optimize the use of color on systems that support 8-bit color. However, some visual components of the Oracle Solaris Desktop system 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 an approximate value. This use of replacement colors causes the grainy appearance.

- Applications that do not use the websafe color palette have fewer colors available. If you do not use the websafe color palette, color errors might occur and some colors might not appear in the user interface of the application. Some applications 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 and then become not viewable.

To optimize the appearance of the Oracle Solaris Desktop system for systems that support 8-bit color:

- Use theme options that use the websafe color palette – Some window border theme options use colors that are in the websafe color palette, therefore, do not have the grainy appearance of other window border options on 8-bit color displays. Use the Bright or Esco themes for the best color display on 8-bit visual modes. For information about how to change theme options, see “Using Theme Options” on page 78.

- Turn 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 about how to turn off this feature, see “Turning Off the Display of Icons in Menus” on page 78.
Disabling Features in the Oracle Solaris Desktop System

The Oracle Solaris Desktop includes features that you can use to restrict access to certain functions. The features are useful in scenarios where you want to restrict the actions that users can perform on a computer. These features are also known as lock down features.

For information about optimizing the desktop environment, see Optimizing the Oracle Solaris Desktop for a Multi-User Environment.

This chapter describes how to disable specific features of the Oracle Solaris Desktop system. This chapter includes the following information:
- “Setting Lockdown Preferences” on page 85
- “Disabling Command-Line Operations” on page 86
- “Disabling Panel Configuration” on page 87

Setting Lockdown Preferences

This section describes how to manually set lockdown preferences. You can also set GConf keys to disable features. You can access the GConf editor by issuing the gconf-editor command or edit GConf keys at the command line. For information about how to set GConf keys, see Chapter 2, “Managing User Preferences With GConf.”

Disabling Lock Screen

To disable the lock screen function, set the /apps/panel/global/disable_lock_screen key to true. When you disable the lock screen function, the following items are removed from the panels:
- Lock Screen menu item from the Main Menu
- Lock menu items from the Add to Panel → Actions menu
- Lock Screen menu item from the Actions menu in the Menu Bar applet
Also, any Lock Screen buttons on panels are disabled.

**Disabling Log Out**

To disable the log out function, set the `/apps/panel/global/disable_log_out` key to `true`. When you disable the log out function, the following items are removed from the panels:

- Log Out user menu item from the Main Menu
- Log Out menu item from the Add to Panel → Actions menu
- Log Out user menu item from the Actions menu in the Menu Bar applet

Also, any Lock Screen buttons on panels are disabled.

**Disabling Command-Line Operations**

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

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

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

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

- `/usr/bin/gnome-terminal`
- `/usr/bin/xterm`
- `/usr/bin/setterm`

The items are removed from the following menus:

- Main Menu
- Add to Panel → Launcher from menu

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

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

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

■ The following items are removed from the panel pop-up menu and from the drawer pop-up menu:
  ■ Add to Panel
  ■ Delete This Panel
  ■ Properties
  ■ New Panel

■ The following items are removed from the pop-up menus for panel objects:
  ■ Remove From Panel
  ■ Lock
  ■ Move

■ The launcher pop-up menu is disabled.
■ The Main Menu pop-up menu is disabled.
■ The launcher drag feature is disabled so that users cannot drag launchers to or from panels.
■ The panel drag feature is disabled so that users cannot drag panels to new locations.
Working With the X Window System

This chapter provides an overview of the X Window System that is available in the Oracle Solaris OS. The X Window System available in Oracle Solaris is based on Xorg. For more information about Xorg, refer to the community web site at http://www.x.org/wiki/.

This chapter includes the following information:
- “X Window System Overview” on page 89
- “Understanding the X Server Process” on page 90
- “Working With X Clients” on page 91
- “Configuring an X Server in Oracle Solaris” on page 92
- “Accessing X11 Display” on page 94

X Window System Overview

The X Window System, commonly referred to as X, is a network-based graphical window system. The X Window System uses a client-server architecture. It enables multiple programs to share and access a common set of hardware. This hardware includes both input and display devices such as mouse, keyboards, video adapters, and monitors that are connected to the server.

The X Window System consists of X server and X clients. The X clients are application programs that do not have direct access to the display. They communicate with the X server which renders the display.

The X architecture allows clients and servers to work on the same system or on different systems with different architectures. The X architecture defines a stream protocol for clients-server communication. This protocol can be exposed over a network to allow clients to connect to a server on a different machine. Therefore, you can set up your system so that the X server is running on the laptop in front of you, and the display is generated by an X client that is running in a remote system.
Understanding the X Server Process

An X server is a single process that controls access to the hardware. The X server performs the following functions:

- Listens for incoming connections from new clients
- Tracks inputs from input devices
- Manages display access of the clients

Depending on the configuration of your Oracle Solaris system, the X server is started from one of the following programs:

- `xdm`, the X Display Manager
- `gdm`, the GNOME Display Manager

The X client can connect to the X server by using TCP/IP, UNIX domain sockets, and several varieties of SVR4 local connections. For information about specifying the transport type for clients, see the DISPLAY NAMES section of the X(5) man page.

Working With X Servers

Oracle Solaris currently includes several X servers designed for different usage types. The following table lists some of the X servers included in the Oracle Solaris OS.

<table>
<thead>
<tr>
<th>X Server</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xorg</td>
<td>Used with physical hardware (input and output) or in virtualization environments that simulate physical hardware. For example, an Xorg server is used when running an Oracle VM VirtualBox guest or when using the keyboard, video, and mouse (KVM) functionality of the Oracle ILOM service processors. For detailed information, see the Xorg(1) man page.</td>
</tr>
<tr>
<td>Xvfb</td>
<td>Used to allocate system virtual memory as a virtual frame buffer. It does not receive input from any input devices and no output is shown. Therefore, the system on which the Xvfb server is running need not have a screen or any input device. For detailed information, see the Xvfb(1) man page.</td>
</tr>
<tr>
<td>Xephyr</td>
<td>Runs as a window displayed on another X server (host X server). It uses the host X server’s input and display devices. For detailed information, see the Xephyr(1) man page.</td>
</tr>
</tbody>
</table>
**X Server Description**

**Xvnc**
Run a VNC session that can be connected by using a VNC client. In a shared connection, multiple VNC clients can connect to the same VNC session. The vncviewer is the default VNC client that is available in the Oracle Solaris 11 OS.
For detailed information, see the vncserver(1) and vncviewer(1) man pages.
For instructions on setting up remote desktop access using VNC, see “Setting Up Remote Desktop Access Using VNC” on page 96.

**Xdmx**
Used as a proxy X server that uses one or more X servers for displaying X clients. It provides X functionality for displays that might be located on different machines.
For detailed information, see the Xdmx(1) man page.

**Xnewt**
Interacts with Sun Ray clients such as Oracle Sun Ray 3 series DTU hardware units or the Oracle Virtual Desktop client for Windows, MacOS, or iPad.
For more information, see [http://docs.oracle.com/cd/E25749_01/index.html](http://docs.oracle.com/cd/E25749_01/index.html).

For more information about the X servers included in the Oracle Solaris OS, see the Xserver(1) man page.

---

### Working With X Clients

X clients are applications that are displayed on an X server. X clients consist of one or more window managers and other applications. The X clients can run on the same system as an X server or run on a different system. The X Window System allows you to run any number of clients simultaneously. The following table lists some of the X clients.

<table>
<thead>
<tr>
<th>X Clients</th>
<th>Description</th>
</tr>
</thead>
</table>
| twm       | The Window Manager is responsible for the look and feel of the X Window System. The Window Manager provides the title bar, shaped windows, icon management, user-defined macro functions, click-to-type and pointer-driven keyboard focus. Some of the common Window Managers in the Oracle Solaris Desktop are:  
  - Metacity Window Manager  
  - Compiz Window Manager  
  - Tab Window Manager  
  For more information, see the metacity(1) and twm(1) man pages. |
| xterm     | Standard terminal emulator for the X Window System. You can open multiple terminal windows and run multiple applications simultaneously.  
  For more information, see the xterm(1) man page. |

---

Chapter 12 • Working With the X Window System 91
### Configuring an X Server in Oracle Solaris

The Xorg server is designed to configure automatically and can run in most situations without the need to edit configuration files. When configuration is needed, the Xorg server gathers configuration details from the following sources:

- `xorg.conf` and `xorg.conf.d` files
- Oracle Solaris SMF configuration properties
- HAL fdi files

The following sections describe the configuration sources in detail.

### Working With Xorg Configuration Files

The `xorg.conf` file is the Xorg server configuration file and is located in the `/etc/X11/` directory.

**Note** – By default, the `xorg.conf` configuration file does not exist until a system administrator creates it in order to change the default configurations.

The following excerpt from the `xorg.conf` configuration file shows the input device and monitor sections:
When an \texttt{xorg.conf} file is needed, you need to mention the sections that are being changed. Unspecified sections continue to use default values. Additional contents are read from any files that exist in directories such as \texttt{/etc/X11/xorg.conf.d} and are merged with the base configuration file. For information about the syntax, search paths, and available options, see the \texttt{xorg.conf(4)} man page.

For sample configuration snippets and examples, see the \texttt{/etc/X11/xorg.conf.avail} file. For example, to stop the X server by using the Ctrl+Alt+Backspace keystroke, see the \texttt{/etc/X11/xorg.conf.avail/90-zap.conf} directory. To use this option, copy or link to the \texttt{/etc/X11/xorg.conf.d} directory:

\verbatim
# ln -s ../xorg.conf.avail/90-zap.conf /etc/X11/xorg.conf.d/90-zap.conf
\endverbatim

Restart the X server to see the changes.

\verbatim
# svcadm restart gdm
\endverbatim

\section*{Working With SMF Configuration Properties}

You can configure the properties of the X server by using the \texttt{scc:/application/x11/x11-server} SMF service. For example, you can disable TCP connections and set the depth of the default visual of the X server. For information about the SMF configuration properties, see the \texttt{SMF PROPERTIES} section of the \texttt{Xserver(1)} man page.

The following example shows how to set the value of the \texttt{options/config_file} property.

\verbatim
EXAMPLE 12-1  Setting a Configuration Property
# svccfg -s scc:/application/x11/x11-server setprop options/config_file=xorg.conf
\endverbatim
Working With HAL fdi Files

You can configure input devices for the Xorg server by using the fdi files read by the Hardware Abstraction Layer (HAL) system daemon, hald.

For example, to force the Emulate3Buttons option on mouse devices to enable pressing left and right buttons together to act as a middle button, include the following XML excerpt in the /etc/hal/fdi/policy/30user/10-x11-3button.fdi file:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<deviceinfo version="0.2">
  <device>
    <!--Default X.org input configuration is defined in: /etc/hal/fdi/policy/30user/10-x11-input.fdi Settings here modify or override the default configuration. See comment in the file above for more information. To see the currently active hal X.org input configuration run lshal or hal-device(1m) and search for "input.x11" keys. Hal and X must be restarted for changes here to take any effect -->
    <match key="info.capabilities" contains="input.mouse">
      <merge key="input.x11_options.Emulate3Buttons" type="string">on</merge>
    </match>
  </device>
</deviceinfo>
```

The lshal command can be used to check the available input devices recognized by HAL and see the options that are set on them. For more information, see lshal(1M), hal(5), and fdi(4) man pages.

Accessing X11 Display

The X server controls which clients can access the server. Clients with access to the server can display windows or images on your screen, receive keyboard input, monitor mouse movement, and interact with the other clients on the system. The Trusted Extensions feature of Oracle Solaris adds security features to prevent labeled clients from accessing the X11 display beyond their security range. For more information, see the Trusted Extensions Label Administration.

By default, the Oracle Solaris Desktop is configured to grant access to the X server and clients run by the user by storing a randomly generated cookie value in a file. The client applications require this value to open a new connection to the X server. This value is stored in the file referenced by the $XAUTHORITY environment variable, which is set automatically by the GNOME Display Manager when you start your session. If you start applications from other display environments, you might need to copy the $XAUTHORITY variable to that environment before clients can be started to authenticate.

For detailed information about the X11 display access, see the xhost(1), xauth(1), and Xsecurity(5) man pages.
Providing Other User Accounts With Access to Your Display

To allow another user ID on the local system to access your display, use the `si:localuser` option of the `xhost` command.

The following example shows how to configure your user account with the correct rights profile and run the SMF Services Visual Panels user interface.

**EXAMPLE 12-2  Allowing the root Account to Display GUIs**

```
user@host:~$ xhost +si:localuser:root
user@host:~$ su
Password:

    /*Visual Panel GUI*/
#
#/usr/bin/vp svcs
```

**Note** – If you use the `su` command to reset the environment variables, you will need to set the `DISPLAY` environment variable in the `su` session to match the `DISPLAY` variable set in the original user’s environment.

```
user@host:~$ xhost +si:localuser:root
    user@host:~$ echo $DISPLAY
:11
user@host:~$ su
Password:

    # export DISPLAY=:11
    # /usr/bin/vp svcs
```

The :11 value will vary depending on each Sun Ray user session. Workstations, typically, have a :0 value on the main console display, but can have additional displays as well.

For more information, see the `Xhost(1)` man page.

Displaying a Client From a Remote Machine

By default, Oracle Solaris 11 is configured to be secure and does not accept direct X11 connections from other machines by using the TCP protocol. To display a client from another machine, use the X11 Tunneling options of the `ssh` command.

For instructions describing how to change the configuration of the `svc:/application/x11/x11-server` SMF service to set up direct and non-tunneled X11 connections over TCP, see the `Xserver(1)` man page.
Note – When you make manual connections, you will need to set up the DISPLAY and
XAUTHORITY environment variables appropriately. When you use the SSH X11 tunnelling
options, these environment variables are automatically set for you.

The following example shows how to display on your desktop, host1, the Package Manager
GUI from host2.

EXAMPLE 12-3  Displaying a Client From a Remote Machine
user@host1:~$ ssh -X admin@host2
Password:
Last login: Fri Apr  6 19:20:18 2012
Oracle Corporation SunOS 5.11 11.1 March 2012
admin@host2:~$ /usr/bin/packagemanager

For more information, see the ssh(1), Xsecurity(5), and xauth(1) man pages.

Setting Up Remote Desktop Access Using VNC

Virtual Network Computing (VNC) is a remote software application that enables you to view
and interact with one computer desktop, the Xvnc server, by using the VNC viewer on another
computer desktop. The two computers do not have to be running the same type of operating
system. Xvnc provides a guest domain graphical login.

This section describes how you can set up your VNC server to provide a guest graphical login.
This section also includes procedures on how you can start VNC manually and at system boot.

Use the vncpasswd command to set the password used to access VNC desktops. The password
is stored on the server. For more information, see the vncpasswd(1) man page.

▼ How to Set Up VNC to Provide a Guest Graphical Login

1  Become superuser or assume the root role.

2  Enable XDMCP connections by editing the /etc/gdm/custom.conf file.
   [xdmcp]
   Enable=true

3  Restart the GDM service.
   # svcadm restart gdm
Note – Restarting the GDM service terminates all corresponding GDM sessions.

4 Enable the Xvnc inetd service.
   # inetadm -e xvnc-inetd

5 Connect from another machine and verify that you see the login screen and can log in to a desktop session.
   # vncviewer hostname:port number
   hostname Name of the host where you ran the service.
   port number Denotes the port number on the host to connect.
   For example:
   # vncviewer myhost:2

▼ How to Start VNC Manually

A user can start VNC manually. A VNC password is required.

Caution – A VNC password should be different from the user’s UNIX login password. A VNC password is not securely encoded on disk or across the network.

1 Become superuser or assume the root role.

2 Start the VNC server.
   # /usr/bin/vncserver

3 Enter the VNC server password.
   Password:
   Verify:
   
   New ’myhost:2 ()’ desktop is myhost:2
   
   Creating default startup script /home/user1/.vnc/xstartup
   Starting applications specified in /home/user1/.vnc/xstartup
   Log file is /home/user1/.vnc/myhost:2.log

4 From another machine, run the vncviewer command with the address reported by the vncserver command.
   # vncviewer hostname:portnumber
For example:

```
# vncviewer myhost:2
```

5 Type the password you provided to the vncserver script.
Verify that you see the login screen and can log in to a desktop session.
This appendix describes the hidden directories that the Oracle Solaris Desktop system adds to the home directories of users.

Hidden directories are commonly used for storing user preferences or preserving the state of a utility. Hidden directory names begin with a period (.). The following table describes hidden directories that the Oracle Solaris Desktop system adds to the home directories of users.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.config</td>
<td>Contains user configuration information. The .config directory is the default location for the XDG_CONFIG_HOME environment variable if the variable is not set. User-defined menus can be created at menus/applications.menu.</td>
</tr>
<tr>
<td>.fonts</td>
<td>Contains fonts specific for a particular user.</td>
</tr>
<tr>
<td>.esd_auth</td>
<td>Contains the authentication cookie for the sound daemon. The sound daemon is the Enlightened Sound Daemon (ESD).</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> – This is a file, not a directory.</td>
</tr>
<tr>
<td>.gconf</td>
<td>Contains the GConf configuration source for the user. When the user sets a preference, the new preference information is added to this location.</td>
</tr>
<tr>
<td>.gconfd</td>
<td>Contains the following GConf daemon details:</td>
</tr>
<tr>
<td></td>
<td>- Configuration information</td>
</tr>
<tr>
<td></td>
<td>- Lock information for objects that are referenced by an Interoperable Object Reference (IOR)</td>
</tr>
<tr>
<td></td>
<td>- State information for objects that are referenced by an IOR</td>
</tr>
</tbody>
</table>
### Hidden Directories in User Home Directories

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.gnome2</td>
<td>Contains user-specific application data that is not stored in the GConf repository. For example, this directory contains the following:</td>
</tr>
<tr>
<td></td>
<td>‣ Keyboard shortcut information</td>
</tr>
<tr>
<td></td>
<td>‣ Window location information</td>
</tr>
<tr>
<td></td>
<td>‣ Desktop entry files for panel launchers</td>
</tr>
<tr>
<td></td>
<td>This directory can also contain subdirectories that store application-specific data.</td>
</tr>
<tr>
<td>.local/share</td>
<td>Specifies the default location for the XDG_DATA_HOME environment variable if the value is not set. User-defined desktop entries can be created in the applications subdirectory. User-defined MIME types can be created in the mime/packages/Overrides.xml file.</td>
</tr>
<tr>
<td>.nautilus</td>
<td>Contains file manager data that is specific to the user. For example, this directory can contain the following information:</td>
</tr>
<tr>
<td></td>
<td>‣ Metadata for the directories with which the user works</td>
</tr>
<tr>
<td></td>
<td>‣ Nautilus emblems that the user adds</td>
</tr>
<tr>
<td></td>
<td>‣ Nautilus desktop background images</td>
</tr>
<tr>
<td>.themes</td>
<td>Contains the controls theme options, window border theme options, and icons theme options that the user adds. The user can add themes from the Theme preference tool.</td>
</tr>
<tr>
<td>.thumbnails</td>
<td>Contains image thumbnails for the user. Image thumbnails are used in the file manager. The file manager contains a preference that the user can select to stop the generation of thumbnail images.</td>
</tr>
</tbody>
</table>
Glossary

application registry
A location that contains text files that register applications. The application registry for the desktop is located in the /usr/share/gnome/application-registry directory.

CLI
Command-line interface.

desktop entry file
A data file that provides information about an item in a menu. The desktop entry file specifies the name, command to run, and icon. Desktop entry files have a .desktop file extension.

directory entry file
A data file that provides information about a menu. The directory entry file specifies details such as a name for the menu, a tool tip for the menu, and an icon to represent the menu. Directory entry files have a .directory file extension.

file content sniffer
A file content sniffer specifies a pattern to search in a file. A file content sniffer associates the pattern with a MIME type. If a match for the pattern is found, the MIME type associated with the pattern is the MIME type of the file.

GConf configuration source
A storage location in the GConf repository. For example, xml:readonly:/etc/gconf/gconf.xml.defaults.

GConf path file
A file that lists the GConf configuration sources and the order to search the sources.

GConf preference key
An element in the GConf repository that corresponds to an application preference. For example, the /apps/gnome-session/options/show_splash_screen key corresponds to the Show Splash Screen on Login option in the Sessions preference tool.

GConf schema
A collective term for a schema key and a schema object.

GConf schema definition file
A file that lists the keys in a particular application and defines the characteristics of the keys. GConf schemas are generated from schema definition files. Schema definition files have a .schemas file extension.

GConf schema key
A key that stores a schema object for a preference key. For example, /schemas/desktop/gnome/interface/font_name is a schema key for the /desktop/gnome/interface/font_name preference key.

GConf schema object
An element in a configuration source that contains information about a preference key. The schema object contains information such as a default value for the preference key and documentation about the preference key.

glob pattern
An instance of pattern matching behavior.
### IOR

Interoperable Object Reference. 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 type

A MIME type identifies the format of a file. The MIME type enables applications to read the file. For example, an email application can use the `image/png` MIME type to detect that a Portable Networks Graphic (PNG) file is attached to an email.

### MIME type description file

An XML file that associates MIME types with file content sniffer information, file extensions, and filename patterns.

### OMF file

Open Source Metadata Framework file. This is a file that is associated with the XML file for a manual. The OMF file contains information about the manual that is used by the Help browser. OMF files have a `.omf` file extension.

### ORB

Object Request Broker.

### pattern mask

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

An application that replaces the image on a screen when the screen is not in use. The screensaver application for the Oracle Solaris Desktop system is XScreenSaver.

### screensaver display

An application that displays images on the screen of the user when the screen is not in use.

### websafe color palette

The websafe color palette is a general-purpose palette of 216 colors. The websafe color palette is designed to optimize the use of color on systems that support 8-bit color. The websafe color palette is also called the *Netscape color palette* and the *Netscape color cube*.
# Index

## A
Allowing other user accounts, access to display, 95
applications
- adding, 64–65
- registry, 63–65

## B
background
- preferences, setting with GConf, 28
- using solid color, 79

## C
color usage, reducing, 82–83
CPU usage, reducing, 77–82

## D
desktop, font preferences, setting with GConf, 28
desktop background, turning off, 81
directory entry files, 34–37
disabling features
- command-line, 86
- panel configuration, 87
disabling lock screen, 85–86
disabling log out, 86
display quality, improving, 82–83
Displaying a client, 95–96

## E
editing menus, 37–40

## F
file manager
- adding menus with, 37–38
desktop background, turning off, 81
location bar, turning off, 81
modifying performance preferences, 79–81
side pane, turning off, 80
toolbar, turning off, 80–81
fonts
- adding fonts, 50
- adding for all users, 50
- adding for individual users, 50
- customizing fonts, 49–53
- fontconfig, 49–50
- introduction, 49–52
- setting with GConf, 27–28
- substitution, 52–53
- X11 font system, 50–52

## G
GConf
- background, setting, 28
- command-line tool, 18–21
- configuration sources, 15–16
- daemon, 18
GConf (Continued)
  fonts, setting, 27–28
  HTTP proxy, setting, 22
  introduction, 13–14
  keyboard accessibility, setting, 23
  keyboard shortcuts, setting, 23
  number of workspaces, setting, 22–23
  panel object preferences, setting, 24–27
  panel preferences, setting individual, 24–27
  preference values, setting, 21–29
  repository, 14–17
  restoring default preference values, 29
  schema definition files
    introduction, 17
  schema keys, 16–17
  schema objects, 16–17
  schemas, 16–17
  splash image, setting, 28–29

H
Help
  OMF files, 75–76
  source documents, 75
  hidden directories, 99–100
  HTTP proxy, setting with GConf, 22

I
individual panels, panels structure, 24–26

K
keyboard
  setting accessibility preferences with GConf, 23
  setting shortcut preferences with GConf, 23

L
lock screen, 85–86
lockdown preferences, 85–86

M
menu definition files, 32–34
.menu files, See menu definition files
menu icons
  and color usage, 83
  and CPU usage, 78–79
  turning off, 78–79
menus
  adding items to, 38
  adding using file manager, 37–38
  deleting menu items, 39
  editing, 37–39
  editing menu items, 39
  editing properties of, 38
  editing user menus, 40
  introduction, 31–32
  merging menus, 40–41
  merging the system menu, 40–41
mime types, 55–65
MIME types
  adding applications, 64–65
  database, 56–59
mime types
  database contents, 57–59
MIME types
  introduction, 55–56
mime types
  mime database
    refreshing mime database, 59
MIME types
  modifying, 61–63
  registering applications for, 63–65
  source XML files, 59–61
  verifying changes, 61–63

N
network traffic, reducing, 82

P
panel objects, panel objects structure, 24–26
panels
  animation, turning off, 79
  preferences, individual, setting with GConf, 24–27
performance, improving, reducing CPU usage, 77–82
preference values
  restoring to default with GConf, 29
  setting with GConf, 21–29
Provide access to your display, 95

R
Rarian, overview, 76
reducing
  color usage, 82–83
  CPU usage, 77–82
  X Window System network traffic, 82
remote desktop access, 96
Remote machine, display, 95–96

S
schemadescription, 16–17
schema definition files
  introduction, 17
schema keys, 16–17
schema objects, 16–17
screensavers
  disable display, 68–69
  introduction, 67–69
  modifying preferences, 68
  modifying screensaver displays, 68–69
  restoring preferences for a user, 67–68
  setting preferences, 67–68
  setting preferences for all users, 67–68
sessions
  default session settings, 73
  introduction, 71
  saving current session, 73
  setting defaults, 71–73
SMF, configuration properties, 93
splash screen, image, setting with GConf, 28–29

T
theme, index file, 44–45
themes
  and color usage, 83
  and CPU usage, 78
  and network traffic, 82
controls options
  creating custom, 47–48
  installing, 45
icons options
  installing, 46
installing new icons, 46–47
installing themes, 43–48
introduction, 43–44
window border options
  and color usage, 83
  and CPU usage, 78
  and network traffic, 82
  changing, 78
  installing, 45–46

V
VNC, 96
  guest graphical login, 96–97
  remote desktop access, 96
  start manually, 97–98

W
window manager, 81–82
workspaces, setting number with GConf, 22–23

X
X Clients, types, 91–92
X server, 90–91
X Server
  configuring X Server, 92–94
  process, 90–91
X Servers, types, 90–91
Index

X Window System
  introduction, 89
  X Clients, 91–92
  X servers, 90–91
X Window System network traffic, reducing, 82
X11 display, accessing X11 display, 94–98
Xorg
  configuration files, 92–93
  HAL
    files, 94

Y
Yelp, introduction, 75–76