Java Desktop System Release 3
Administration Guide

Contents

Preface

The *Java Desktop System Release 3 Administration Guide* provides information about how to administer a system running the following releases:

- Sun Java™ Desktop System Release 3 for the SPARC® platform edition of the Solaris™ Operating System.
- Sun Java Desktop System Release 3 for the x86 platform edition of the Solaris Operating System.

Most of the information in the guide is generic to all releases of the Java Desktop System. Where the information is not generic, the platform is indicated.

Who Should Use This Guide

This guide is for system administrators who are responsible for administering one or more systems that run the Java Desktop System Release 3. For information about how to use the Java Desktop System, see the *Java Desktop System Release 3 User Guide*.

Before You Read This Guide

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

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

How This Guide Is Organized

This guide is organized as follows:

- Chapter 1 describes how to use GConf to manage user preferences.
- Chapter 2 describes the implementation of menus and how to customize menus.
- Chapter 3 describes the types of theme that are available in the Java Desktop System, how to install themes, and how to create a custom theme.
- Chapter 4 describes how to configure fonts for the Java Desktop System.
- Chapter 5 describes how applications detect MIME types, how to register MIME types, and how to add applications to the Java Desktop System.
- Chapter 6 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 7 introduces session management, and describes how to set session defaults. This chapter also contains information about sessions and login scripts.
- Chapter 8 describes the Help system in the Java Desktop System.
- Chapter 9 describes how to improve the performance of the Java Desktop System.
- Chapter 10 describes how to disable particular features of the Java Desktop System.
- Appendix A describes the hidden directories that the Java Desktop System adds to the home directories of users.
- Glossary is a list of terms used in this guide, and definitions of the terms.

Related Documentation

The following manuals are related to this guide:

- *Java Desktop System Release 3 Accessibility Guide*
- *Java Desktop System Release 3 Installation Guide*
- *Java Desktop System Release 3 Release Notes*
- *Java Desktop System Release 3 Troubleshooting Guide*
- *Java Desktop System Release 3 User Guide*

Supported Solaris Systems

The Solaris Operating System (Solaris OS) supports various architectures, processor families, and systems. Table P–1 describes where you can find information about systems that are supported by the Solaris OS, relevant to this product release.

TABLE P–1 Supported Solaris Systems

Architecture	Processor Families	Solaris Systems
SPARC	■ SPARC64 ■ UltraSPARC	See the latest Solaris Hardware Platform Guide at the following location: http://docs.sun.com
x86	■ AMD64 ■ Pentium ■ Xeon EM64T	See the latest Solaris x86 Platform Hardware Compatibility List at the following location: http://www.sun.com/bigadmin/hcl

Accessing Sun Documentation Online

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

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print. For a list of documents and how to order them, see "Buy printed documentation" at http://docs.sun.com.

Typographic Conventions

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

TABLE P–2 Typographic Conventions

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

Shell Prompts in Command Examples

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

TABLE P–3 Shell Prompts

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

Mouse Usage Conventions

The following table lists the conventions for mouse usage in documentation for the Java Desktop System.

Action	Definition
Click	Press and release the left mouse button, without moving the mouse.
Click-and-hold	Press and do not release the left mouse button.
Left-click	Same as *click*. Left-click clarifies the action when there might be confusion with *right-click*.
Middle-click	Press and release the middle mouse button, without moving the mouse.
Right-click	Press and release the right mouse button, without moving the mouse.
Double-click	Press and release the left mouse button twice in rapid succession without moving the mouse.
Drag	Click-and-hold a mouse button, then move an object. For example, you can drag a window or an icon. The left and middle mouse buttons can perform drag actions.
Drag-and-drop	Click-and-hold a mouse button, then move an object. For example, you can drag-and-drop a window or an icon. Release the mouse button to place the object in a new location.
Grab	Point to an item that you can move, and click-and-hold on the mouse button. For example, you can grab the titlebar of a window, then drag the window to a new location.

Using GConf

This chapter describes how to use GConf to manage user preferences.

Introducion to GConf

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

- Set mandatory values for particular preferences for all users. In this way, system administrators can control whether users can update particular preferences.

- Set default values for particular preferences for all users.

- Use suggested values for preferences that are specified in definition files for the preferences.

- Read documentation on each preference.

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

GConf has the following components:

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

GConf Repository

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

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

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

Preference keys typically have simple 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 backend module that is used to read the repository. The following is an example of the /desktop/gnome/interface/font_name preference key when an Extensible Markup Language (XML) backend module is used to read the repository:

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

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

GConf Configuration Sources

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

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

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

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

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

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

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

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

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

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

The `apoc` references in the GConf path file specify a Sun Java Desktop System Configuration Manager backend module. For more information about Sun Java Desktop System Configuration Manager, see the Sun Java Desktop System Configuration Manager documentation at http://docs.sun.com.

GConf Schemas

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

Item	Description
Preference key	An element in the GConf repository that corresponds to an application preference.
Schema key	A key that stores a schema object for a preference key.
Schema object	An element in a configuration source that contains information for a preference key, such as the following: ■ The name of the application that uses the preference key. ■ The type of value required for the preference key, for example integer, boolean, and so on. ■ A default value for the preference key. ■ Brief documentation on the preference key.

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

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

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

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

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

Typically, schemas are stored in the default configuration source.

GConf Schema Definition Files

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

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

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

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

GConf Daemon

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

An instance of the GConf daemon is started for each user. If a user logs in to more than one system, an instance of the GConf daemon is started for each session. You can configure the Java Desktop System to use only one instance of the GConf daemon when a user logs in to more than one system. For information about how to do this, see "Configuring the Java Desktop System to Use One Instance of the GConf Daemon" on page 19.

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

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

1. Search for the value of the preference key in each configuration source, in the order specified in the path file. If the value is found, return the value.

2. If a value is not found, search for the schema key that corresponds to the preference key in each configuration source, in the order specified in the path file.

3. If the schema key is found, check the value of the schema key.

4. If the value of the schema key is a schema object, return the suggested value in the `<default>` element of the schema object.

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

To terminate the GConf daemon, execute the following command:

gconftool-2 --shutdown

Configuring the Java Desktop System to Use One Instance of the GConf Daemon

By default, the Java Desktop System creates multiple instances of the GConf daemon when users log in to more than one system. You can configure the Java Desktop System to use only one instance of the GConf daemon when a user logs in to more than one system. The user must have an *Network File System (NFS)* home directory.

When a user uses one instance of the GConf daemon on more than one system, and the user changes the value of a preference, the change is applied to all of the sessions to which the user is logged in. For example, if the user selects to show only icons in toolbars in the Menus & Toolbars preference tool in one session, the toolbars on all open applications are updated instantly in all sessions to which the user is logged in.

Note – Users can only use a single instance of the GConf daemon for multiple sessions if all of the sessions can access the home directory of the user.

The Java Desktop System uses *Common Object Request Broker Architecture (CORBA)*. CORBA enables application objects to communicate with one another regardless of what programming language the applications are written in, and regardless of what operating system the applications are running on.

In CORBA, an *Object Request Broker (ORB)* communicates between servers and clients. One of the ORBs in the Java Desktop System is ORBit2. GConf uses the TCP protocol in ORBit2 to communicate between the GConf daemon and the sessions to which the user is logged in.

To configure a system to use one instance of the GConf daemon for each user, perform the following steps:

1. For all the systems that the users log in to, ensure that the file `/etc/orbitrc` contains the following line:

   ORBIIOPIPv4=1

2. Set the value of the `GCONF_GLOBAL_LOCKS` environment variable to `1`. This ensures that GConf creates locks in the home directory of the user, and not in a directory on the local system.

3. Restart the GConf daemon.

To configure the system to use multiple instances of the GConf daemon for each user, unset the value of the `GCONF_GLOBAL_LOCKS` environment variable. Then restart the GConf daemon.

Caution – The TCP protocol in ORBit2 is not completely secure because the communications between servers and clients are not encrypted.

GConf Command Line Tool

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

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

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

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

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

TABLE 1–1 gconftool-2 Command Options

Option	Function
--all-dirs	Lists all subdirectories in a directory that you specify.
--all-entries	Displays the values of all keys in a directory that you specify.
--config-source= *configuration-source*	Use this option with the --direct option to specify a configuration source to use. If you do not specify a configuration source with this option, the command runs on all configuration sources in the path file.
--direct	Use this option with the --config-source option to access a configuration source directly. When you use this option, GConf bypasses the server. Ensure that the GConf daemon, gconfd-2, is not running before you use this option.

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

Option	Function
`--dump`	Generates a list that contains all preference keys in a GConf repository directory that you specify. The list contains XML descriptions of all the keys. The list is contained in a `<gconfentryfile>` element.
	For example, you can redirect the output from this option to generate a file that lists all keys that are related to your panel configuration. You can use the `--load` option with this file.
`--get`	Displays the value of a preference key that you specify. Also displays the values of the elements in the schema object for a schema key that you specify.
`--help`	Displays a help message about the `gconftool-2` command, and the options that you can use with the `gconftool-2` command.
`--load=`*filename*	Use this option to set the values of preference keys in the current directory in a configuration source to the values in the file that you specify. The file that you specify must contain XML descriptions of the keys, in a `<gconfentryfile>` element.
`--long-desc=`*description*	Use this option with the `--set-schema` option to specify a long description for a schema key.
`--makefile-install-rule`	Installs schema definition files to applications.
`--owner=`*owner*	Use this option with the `--set-schema` option to specify an owner for a schema key.
`--recursive-list`	Displays the values of all preference keys in all subdirectories in a directory that you specify.
`--recursive-unset`	Resets the values of all preference keys, in all subdirectories in a directory, from the user setting to the setting in the default configuration source.

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

Option	Function

`--set`	Sets the value of a preference key, and writes the value to the user configuration source. Use the `--type` option with the `--set` option to specify the data type of the value that you want to set. For example, the following command sets the value of the `/apps/gnome-terminal/profiles/Default/background_color` key in the user configuration source:
	
	`# gconftool-2 --set`
	`"/apps/gnome-terminal/profiles/Default/background_color"`
	`--type string "#000000"`
	
	You can also use the `--direct` option and the `--config-source` option with the `--set` option to write a value to another configuration source.

`--set-schema`	Sets the value of an attribute in a schema key, and writes the value to the default configuration source.
	
	Use the following options with the `--set-schema` option to specify the attribute that you want to update:
	■ `--type`
	■ `--short-desc`
	■ `--long-desc`
	■ `--owner`
	
	For example, the following command sets the short description in the schema key for the `/apps/gnome-terminal/profiles/Default/background_color` key:
	
	`# gconftool-2 --set-schema`
	`"/schemas/apps/gnome-terminal/profiles/Default/background_color"`
	`--short-desc "Default background color of`
	`terminal"`

`--short-desc=`*description*	Use this option with the `--set-schema` option to specify a short description for a schema key.
`--shutdown`	Terminates the GConf daemon.

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

Option	Function
--type=*data-type*	Use this option to specify the data type when you set a value of a preference key. You can also use this option when you set the value of an attribute in a schema key. The following is a list of valid data types: ■ bool ■ float ■ int ■ list ■ pair ■ string
--unset	Resets the value of a preference key from the user setting to the setting in the default configuration source.
--usage	Displays a brief help message about the gconftool-2 command, and the options that you can use with the gconftool-2 command.

Setting Preference Values

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

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

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

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

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

The user cannot override this preference value.

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

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

The user can override this preference value.

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

Setting General Preferences

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

To Set HTTP Proxy Preferences

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

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

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

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

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

To Set the Number of Workspaces

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

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

To set a default number of workspaces, execute 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.

To Set Keyboard Accessibility Preferences

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

To Set Keyboard Shortcut Preferences

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

Setting Panel and Panel Object Preferences

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

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

The configuration of individual panels and of panel objects is a complex task. To configure individual panels and panel objects, you must first understand the structure of the `panel-default-setup.entries` file. For more information about the `panel-default-setup.entries` file, see "Structure of Panel Configuration File" on page 26.

To set preferences for individual panels and panel objects, you must set the values of many preferences in a configuration source. The easiest way to set the values of panel preferences is to use the `gconftool-2` command with the `--dump` and `--load` options. For more information about how to set preferences for panels and objects on panels, see "To Set Preferences for Individual Panels and Panel Objects" on page 29.

Structure of Panel Configuration File

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

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

1. An `entrylist` element that specifies the base path for all the keys in the file. The following excerpt from `panel-default-setup.entries` specifies `/apps/panel/default_setup` as the base path for keys:

   <entrylist base="/apps/panel/default_setup">

   For example, if the key `general/toplevel_id_list` is referred to in `panel-default-setup.entries`, the full path for the key is `/apps/panel/default_setup/general/toplevel_id_list`.

2. Keys that specify the general structure of panels, panel applications, and other panel objects in the desktop. The following keys specify the number of panels, panel objects, and panel applications that appear in the desktop:

   - `general/toplevel_id_list`
   - `general/object_id_list`
   - `general/applet_id_list`

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

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

In the `panel-default-setup.entries` file, the identifier `bottom_panel` identifies the panel at the bottom of the screen.

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

toplevels/*panel-name*/*panel-property-key*

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

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

```
<entry>
   <key>objects/main_menu/object_type</key>
   <schema_key>/schemas/apps/panel/objects/object_type</schema_key>
   <value>
    <string>menu-object</string>
   </value>
  </entry>
  <entry>
   <key>objects/main_menu/toplevel_id</key>
   <schema_key>/schemas/apps/panel/objects/toplevel_id</schema_key>
   <value>
    <string>bottom_panel</string>
   </value>
  </entry>
  <entry>
   <key>objects/main_menu/position</key>
   <schema_key>/schemas/apps/panel/objects/position</schema_key>
   <value>
```

```
        <int>0</int>
      </value>
    </entry>
```

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

```
<entry>
    <key>applets/window_list/object_type</key>
    <schema_key>/schemas/apps/panel/objects/object_type</schema_key>
    <value>
     <string>bonobo-applet</string>
    </value>
   </entry>
   <entry>
    <key>applets/window_list/toplevel_id</key>
    <schema_key>/schemas/apps/panel/objects/toplevel_id</schema_key>
    <value>
     <string>bottom_panel</string>
    </value>
   </entry>
   <entry>
    <key>applets/window_list/position</key>
    <schema_key>/schemas/apps/panel/objects/position</schema_key>
    <value>
     <int>3</int>
    </value>
   </entry>
 .
 .
 .
   <entry>
    <key>applets/window_list/bonobo_iid</key>
    <schema_key>/schemas/apps/panel/objects/bonobo_iid_type</schema_key>
    <value>
     <string>OAFIID:GNOME_WindowListApplet</string>
    </value>
   </entry>
```

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

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

To Set Preferences for Individual Panels and Panel Objects

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

1. Log in to a session with a test user account.

2. Configure the panels as required.

3. Use the --dump option with the gconftool-2 command line tool to generate a file that contains an XML description of your panel configuration. The --dump option generates a list that contains all preference keys in a GConf repository directory that you specify.

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

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

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

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

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

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

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

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

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

6. Log out of the test user account.

Setting Look-and-Feel Preferences

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

To Set Font Preferences

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

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

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

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

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

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

To Set Background Preferences

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

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

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

To Set Splash Image Preferences

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

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

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

To Restore Default Preference Values

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

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

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

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

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

Customizing Menus

This chapter describes how the Java Desktop System implements menus and how you can customize menus.

Introduction to Menus

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

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

The menus in the Java Desktop System use the following components:

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

File Abstraction Layer

The `gnome-vfs` file abstraction layer provides a simplified and generalized way for applications to interact with files. The file abstraction layer also provides *Uniform Resource Identifier* (URI) locations that map to particular menu configuration files.

To add a menu or a menu item for all users, you must add the menu or menu item to one of the URI locations. Table 2–1 lists the menus to which you can add items, and the URI locations that correspond to the menus.

TABLE 2–1 Menus and URI Locations for System Administrators

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

If a user wants to add a menu or a menu item to their system only, the user must add the menu or menu item to other URI locations. Table 2–2 lists the menus to which users can add items, and the URI locations that correspond to the menus.

TABLE 2–2 Menus and URI Locations for Users

Menu	URI Locations
Applications menu	`applications:///`
Preferences menu	`preferences:///`

For more information about how users can customize menus, see the *Java Desktop System Release 3 User Guide*.

Vfolders and Menus

In general terms, a *vfolder* is a virtual representation of items that reside in a physical location or physical locations on your system. For example, a vfolder might represent the contents of several directories. A vfolder is an abstraction from one or more physical locations. In terms of menus in the Java Desktop System, a vfolder is a representation in a menu of items that might be physically located in several directories.

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

The following is a sample from a vfolder information file:

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

Table 2–3 describes some of the elements in vfolder information files.

TABLE 2–3 Vfolder Information File Elements

Element	Description
`<Folder>`	Contains the elements that define the name, content, and structure of the menu.
`<Name>`	Specifies the name of the menu.
`<Desktop>`	Specifies the name of the directory entry file that specifies the name, comment, and icon for the menu.

TABLE 2–3 Vfolder Information File Elements   *(Continued)*

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

Desktop Entry Files

A *desktop entry file* is a data file that provides information about an item in a menu. The desktop entry file specifies the details for the item such as a name, a command to 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 have a `.desktop` file extension.

The following is a sample desktop entry file:

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

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

TABLE 2–4 Desktop Entry Keys

Desktop Entry Key	Description
`Encoding`	Specifies the encoding of the desktop entry file.

TABLE 2–4 Desktop Entry Keys   *(Continued)*

Desktop Entry Key	Description
Name	Specifies the name of the item. This name is displayed on the item in the menu.
Comment	Specifies a short description of the item. The comment is displayed as a tooltip when you point to the item in the menu.
Exec	Specifies a command to execute when you choose the item from the menu.
Icon	Specifies the filename of an icon that represents the item. Does not specify the path to the filename, or the file extension.
Terminal	Specifies whether the command in the Exec key runs in a terminal window. If the value is true the command runs in a terminal window. If the command does not create a window in which to run, the value of this key must be true.
Type	Specifies the type of item. This value is one of the following: ■ Application: Enter this option for an item that starts an application. ■ Link: Enter this option for an item that links to a file, folder, or URI.
Categories	Specifies the keywords that describe the item. The keywords are separated with semicolons (;). To see a list of the standard category keywords, see the desktop menu specification at the following URL: http://www.freedesktop.org The vfolder information files map the keywords to menus.

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

http://www.freedesktop.org

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

Directory Entry Files

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

The following is a sample directory entry file:

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

Table 2–5 describes the most important keys in directory entry files.

TABLE 2–5 Directory Entry Keys

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

Editing Menus

You use the following Java Desktop System components to edit menus:

- Nautilus file manager
- Menus on panels

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

When you use panels to customize menus for all users, you use the menu item popup menu. For more information, see the *Java Desktop System Release 3 User Guide*.

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

Adding Menus

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

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

To Add a Menu Using the File Manager

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

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

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

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

---

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

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

To Add a Menu Using Menu Files

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

1. Create a directory entry file for the item that you want to add. Create the directory entry file in the `/usr/share/gnome/vfolders` directory. For more information about directory entry files, see "Directory Entry Files" on page 38.

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

3. In the vfolder information file, add a `<Folder>` element for the new menu. For more information about vfolder information files, see "Vfolders and Menus" on page 34.

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

To Add an Item to a Menu

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

1. Create a desktop entry file for the item that you want to add. For more information about desktop entry files, see "Desktop Entry Files" on page 36.

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

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

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

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

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

To Edit the Properties of a Menu

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

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

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

3. Modify the properties of the menu in the **Launcher Properties** dialog. For more information about the elements in the **Launcher Properties** dialog, see the *Java Desktop System Release 3 User Guide*.

4. Click OK.

To Edit a Menu Item

To edit a menu item, perform the following steps:

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

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

3. Modify the properties of the menu item in the **Launcher Properties** dialog. For more information about the elements in the **Launcher Properties** dialog, see the *Java Desktop System Release 3 User Guide*.

4. Click OK.

To Delete an Item from a Menu

Perform the following steps:

1. From a panel, open the menu that contains the item that you want to delete.
2. Right-click on the item that you want to delete, then choose Remove this item.

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

To Configure Menus That Users Cannot Modify

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

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

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

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

Installing Themes

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

Introduction to Themes

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

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

Controls

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.

To select an option for the controls setting, users start the Theme preference tool, click on the Theme Details button, then select from the **Controls** tabbed section in the preference tool.

Window frame

The window frame setting for a theme determines the appearance of the frames around windows only.

To select an option for the window frame setting, users start the Theme preference tool, click on the Theme Details button, then select from the **Window Border** tabbed section in the preference tool.

Icon   The icon setting for a theme determines the appearance of the icons on panels and the desktop background.

To select an option for the icon setting, users start the Theme preference tool, click on the Theme Details button, then select from the **Icons** tabbed section in the preference tool.

Theme Index File

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

The following is a sample theme index file:

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

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

The following table describes the keys in theme index files:

TABLE 3–1 Keys in Theme Index Files

Index File Key	Description
Type	Specifies that this theme determines the appearance of several theme options, such as controls, window frames, and icons.
Name	The name of the theme. This is the name of the theme that is displayed in the Theme preference tool.
Comment	A brief description of the theme. This is the text that is displayed under the name of the theme in the Theme preference tool.

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

Index File Key	Description
GtkTheme	Corresponds to the controls setting in the **Theme Details** dialog in the Theme preference tool. Specifies which controls setting option to apply to windows, panels, and panel applications.
IconTheme	Corresponds to the icons setting in the **Theme Details** dialog in the Theme preference tool. Specifies which icons setting option to apply to panels and the desktop background.
MetacityTheme	Corresponds to the window frame setting in the **Theme Details** dialog in the Theme preference tool. Specifies which window frame setting option to apply to windows.
ApplicationFont	Corresponds to the application font setting in the Font preference tool.

To Install a New Controls Option

You can add a new option for the controls setting. When you add a new option, users can select the option from the **Theme Details** dialog in the Theme preference tool. Controls options reside in the `/usr/share/themes` directory. The typical structure of a controls option in the file system is as follows:

Option file   `/usr/share/themes/`*option-name*`/gtk-2.0/gtkrc`

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

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

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

To Install a New Window Frame Option

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

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

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

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

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

To Install a New Icons Option

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

Option file   `/usr/share/icons/`*option-name*

Image files   `/usr/share/icons/`*option-name*`/`*image-size*`/`*ui-category*`/*.*`

             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.

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

Installing New Icons for Themes

The Java Desktop System contains the following types of icon:

- Icons that are used internally by GTK+ applications, or GTK+ stock icons. Users can use the **Controls** tabbed section in the Theme Details dialog to specify a set of these icons for their desktop.

- Icons that are used to represent applications in the Main Menu, and in the file manager. Users can use the **Icons** tabbed section in the Theme Details dialog to specify a set of these icons for their desktop.

The Java Desktop System provides several themes that are designed for users with special visual needs. For example, some of the themes are designed for users with low vision. Several versions of icons might be required so that the icon can be displayed properly in each theme. You can create the icons in several formats, for example Portable Network Graphic (PNG) format. The suggested size of icons for the Java Desktop System is 48 pixels by 48 pixels. At this size, most themes can rescale the icons.

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

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

Install the icons to the image files location that is specified for the theme in "To Install a New Controls Option" on page 45.

For example, to add GTK+ icons to the HighContrastLargePrint theme, perform the following steps:

- Add the icons to the `/usr/share/themes/HighContrastLargePrint/pixmaps` directory.

- Modify the `/usr/share/themes/HighContrastLargePrint/gtk-2.0/gtkrc` file to associate the icon with a GTK stock icon identifier.

To add Main Menu icons for new applications to the HighContrastLargePrint theme, install the icons to the `/usr/share/icons/HighContrastLargePrint/48x48/apps` directory.

For more information about how to create icons for application launchers and for panels, see the following URL:

http://www.freedesktop.org/Standards/icon-theme-spec

To Create a Custom Controls Option

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

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

   - `/usr/share/themes/SmallPrint`
   - `/usr/share/themes/SmallPrint/gtk-2.0`

2. Locate the `gtkrc` file that is closest to meeting the needs of your users. Copy the file to the `gtk-2.0` directory of your new option.

3. Open the `gtkrc` file in a text editor, and modify the attributes of the interface elements as required.

4. If the new option includes images, install the images for the new option in the `pixmaps` directory of your new option. If the new option uses images from another option, you do not need to create copies of the images for the new option. Instead, ensure that the reference to the images in the `pixmap_path` entry in the `gtkrc` file is correct.

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

For more information about `gtkrc` files, see the following URL:

http://developer.gnome.org/doc/API/2.0/gtk/gtk-Resource-Files.html

Configuring Fonts

This chapter describes how to configure fonts in the Java Desktop System.

Introduction to Configuring Fonts

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

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

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

http://freedesktop.org/software/fontconfig

Font Substitution

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

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

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

To Add a Font for All Users

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

1. Copy the font file to one of the directories in the `/etc/fonts/fonts.conf` file. Typically, additional fonts are stored in the `/usr/share/fonts/` directory.

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

   # **fc-cache**

   Alternatively, to update only from a specific directory, execute the following command:

   # **fc-cache** *directory-name*

   Alternatively, to update only from the directories in `/etc/fonts/fonts.conf`, and to view a list of the updated directories, execute the following command:

   # **fc-cache --system-only -v**

To Add a Font for an Individual User

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

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

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

   # **fc-cache** *directory-name*

For information about other options to use with the `fc-cache` command, see "To Add a Font for All Users" on page 50.

MIME Types

This chapter describes MIME types, and the MIME database, and also describes how to add applications to the desktop.

Introduction to MIME Types

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

The file manager uses MIME types to identify the type of a file. The file manager needs to know the MIME type of a file to perform the following tasks:

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

If you add a new application to the desktop, you must ensure that other applications can recognize the files associated with the application. You must ensure that if the new application requires a new MIME type, that the new application installs the new MIME type.

This chapter describes the MIME database, and how applications are associated with MIME types. This chapter also describes the steps that you must follow to add a new application to the desktop.

MIME Database

All MIME type information is stored in a database. The MIME database is located in the directory `/usr/share/mime/`. The MIME database contains a large number of common MIME types, stored in the file `/usr/share/mime/packages/freedesktop.org.xml`. Applications can add new MIME types to the MIME database.

Applications use the MIME database to detect the MIME type of a file, in the following ways:

- The application checks the contents of the file, then uses the MIME database to identify the appropriate MIME type. The MIME database contains *file content sniffer* information. The file content sniffer information provides details of a specific pattern in the file. The MIME database associates this pattern with a MIME type. The application checks the file for the pattern. If the application finds a match for the pattern in the file content, the MIME type associated with the pattern is the MIME type of the file.

  The `magic` elements in the MIME database specify the file content sniffer information.

- The application checks the filename, then uses the MIME database to identify the appropriate MIME type. The MIME database associates particular file extensions and filename patterns, with particular MIME types. The application checks the filename to search for a particular pattern in the filename. If a match for the filename is found, the MIME type associated with the extension or pattern is the MIME type of the file.

  The `glob` elements in the MIME database specify the filename pattern and file extension pattern information.

The MIME database also contains a text description of each MIME type. The MIME database can also contain a list of applications that you can use to view or edit a MIME type.

Structure of the MIME Database

The following is a sample from the MIME type database:

```
<mime-type type="image/png">
    <comment>PNG image</comment>
    <comment xml:lang="de">PNG-Grafik</comment>
    :
    :
    :
```

```
    <magic priority="50">
       <match offset="0" type="string" value="\x89PNG" />
    </magic>
    <glob pattern="*.png" />
 </mime-type>
```

Table 5–1 describes the elements in the MIME type database.

TABLE 5–1 MIME Type Database Elements

Element	Description
mime-info	This is the document element, or highest level element in the MIME database. This element contains the required information for all MIME types. The namespace URI of this element is as follows: http://www.freedesktop.org/standards/shared-mime-info
mime-type	Describes one MIME type. This element has a type attribute, which specifies the MIME type.
comment	Provides a description of the MIME type that is used in the user interface. The description can be displayed in the file manager and other applications.
	The database might contain comment elements with xml:lang attributes. These elements can provide the descriptive text in other languages.
magic	Contains match elements. The match elements specify a particular pattern to search for in the contents of a file.
	This associates a specific pattern in a file with the MIME type specified in the mime-type element. If an application finds a match for the pattern, the MIME type associated with the pattern is the MIME type of the file.
	The magic element includes a priority attribute. You can specify a priority value for the pattern. If there are identical patterns in the magic elements in the MIME database, the element with the highest priority value takes precedence.
	For more information about the attributes of a match element, see "Attributes in Match Elements" on page 56.
glob	Specifies a filename pattern to match with the MIME type. Any file that matches this filename pattern takes this MIME type.

TABLE 5–1 MIME Type Database Elements　　*(Continued)*

Element	Description
root-XML	If a file is identified as an XML file, you can associate a more specific MIME type than text/xml with the file. To do this, you assign values to the following attributes of the root-XML element: ■ namespaceURI: Type a namespace that uniquely identifies the type of XML file. ■ localName: Type a local name for the XML file. This name uniquely identifies the MIME type within the namespace of the XML file. This element is optional. This element is not present in the example.

Attributes in Match Elements

Table 5–2 describes the attributes in a match element.

TABLE 5–2 Attributes in match Elements

Attribute	Description
offset	Specifies characters to ignore in the file before searching for a pattern. You can specify an integer value. This specifies the number of characters to ignore in the file before searching for a pattern. Alternatively, you can specify a range of characters to search. For example, if you specify offset="1:10", only the first ten characters are searched.
type	Specifies the type of pattern to search for. You can use the following types for this attribute: ■ big16 ■ big32 ■ byte ■ host16 ■ host32 ■ little16 ■ little32 ■ string
value	Specifies the pattern to search for.

TABLE 5–2 Attributes in match Elements　　*(Continued)*

Attribute	Description
mask	Specifies a *pattern mask*. A pattern mask identifies bits in the pattern to ignore when searching for a pattern in a file.
	Masks for strings must be in hexadecimal format, and begin with 0x.
	This attribute is optional. This attribute is not present in the example.

Files Generated From the MIME Database

When a new MIME type is installed, the update-mime-database command is executed to update the MIME database. Table 5–3 describes the files that the update-mime-database command generates:

TABLE 5–3 Files Generated by update-mime-database

File	Description
globs	The globs file is generated from the glob elements in the MIME database, and from the glob elements in any MIME type description files in the packages directory. The globs file contains a simple list of MIME types and the filename patterns that are associated with that file type.
	The following are examples of entries in the globs file:application/x-bzip-compressed-tar:*.tar.bz2 text/html:*.html image/png:*.png
	If there are two identical filename patterns in the glob elements in the MIME database, only the entry from the MIME type description file which comes later alphabetically is added to the globs file. For example, if the MIME type description files abc.xml and def.xml contain identical filename patterns in the glob elements, only the glob element from def.xml is added to the globs file. That is, filename patterns in alphabetically later MIME type description files take precedence.
	If you want a MIME type description to take precedence over other MIME type descriptions, no matter what the alphabetical order is, put the MIME type description in a file called Override.xml in the packages directory.

TABLE 5–3 Files Generated by update-mime-database   *(Continued)*

File	Description
magic	The magic file is generated from the magic elements in the MIME database, and from the magic elements in any MIME type description files in the packages directory. The magic file is stored in binary format.
XMLnamespaces	The magic file is generated from the root-XML elements in the MIME database. This file lists all of the XML namespaces that the XML file types use.
content-type/*content-subtype*.xml	These files are generated from the comment elements in the MIME database.

Adding a MIME Type to the MIME Database

To add a new MIME type to the database, an application must install a *MIME type description file*. A MIME type description file is an XML file that associates MIME types with file content sniffer information, and with file extensions and filename patterns.

The following is a sample MIME type description file:

```
<?xml version="1.0" encoding="UTF-8"?>
<mime-info xmlns='http://www.freedesktop.org/standards/shared-mime-info'>
  <mime-type type="image/png">
    <comment>PNG image</comment>
    <comment xml:lang="de">PNG-Grafik</comment>
    :
    :
    :
    <magic priority="50>
      <match offset="0" type="string" value="\x89PNG" />
    </magic>
    <glob pattern="*.png" />
  </mime-type>
</mime-info>
```

The elements in a MIME type description file are similar to the elements in the MIME database. For information about the elements in a MIME type description file, see "Structure of the MIME Database" on page 54.

After the application installs the MIME type description file, the application must execute the update-mime-database command to update the MIME database.

Adding an Application to the Desktop

To add an application to the desktop, perform the following steps:

1. Add a menu item for the application. For more information about how to add an item to a menu, see Chapter 2.

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

3. If the application uses a new MIME type and does not automatically add the MIME type to the MIME database, add a MIME type description file to `/usr/share/mime/packages/`.

4. If the application installs a new MIME type description file, ensure that the `update-mime-database` command is executed during the installation, to update the MIME database.

5. If the application uses a new MIME type, add icons for the MIME type to `/usr/share/icons/`*theme-name*`/`*icon-size*`/mimetypes`. For more information about icons and themes, see Chapter 3.

6. Log out of the session, then log in again.

Setting Screensavers

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.

Introduction to Screensavers

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

Setting Screensaver Preferences

Default screensaver preferences are stored in a file that is called `XScreenSaver`. This file is stored in different locations, depending on your operating environment, as shown in the following table:

Operating System	File Location
Solaris	`/usr/openwin/lib/app-defaults/XScreenSaver`

To modify screensaver application preferences, users can use the Screensaver preference tool. When a user modifies the screensaver preferences, the preferences are stored in the home directory of the user, in the `$HOME/.xscreensaver` file. For information about screensaver preferences, see the latest version of the *Java Desktop System Release 3 User Guide*.

Users can also run a command to start the Screensaver preference tool. The command that you use depends on your operating environment, as shown in the following table:

Operating System	Command
Solaris	`/usr/openwin/bin/xscreensaver-demo`

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

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

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

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

xscreensaver-command -restart

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

Modifying Screensaver Displays

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

To Add a Screensaver Display on Solaris Systems

To add a new screensaver display, perform the following steps:

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

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

To Disable a Screensaver Display on Solaris Systems

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

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

Some screensaver displays show the contents of the screen of the user. To avoid
security violations, screensaver displays that show the contents of the screen are not
included in the Java Desktop System.

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

Session Management

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

Introduction to Sessions

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

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

- The session manager, `gnome-session`.
- The GConf X settings daemon, `gnome-settings-daemon`.
- The gnome-panel application, which runs the panels on the desktop.
- The Metacity window manager.
- The Nautilus file manager.
- The `gnome-keyring` daemon.

Setting Session Defaults

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

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

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

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

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

Using Login Scripts on Solaris Systems

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

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

number-greater-than-1000`.name`

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

Help System

This chapter describes the Help system in the Java Desktop System.

Introduction to the Help System

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

http://www.oasis-open.org/docbook/xml

The Java Desktop System can display Help in two Help browsers, as follows:

- JDSHelp
- Yelp

JDSHelp Browser

The JDSHelp browser is based on the Javahelp browser. For more information about Javahelp file formats, how to write documents for Javahelp, and how to group documents in hierarchies, see the following URL:

http://java.sun.com/products/javahelp/reference/docs/index.html

Yelp Browser

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

OMF Files

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

When you install an application, the installation process installs an OMF file to a directory of OMF files. If a Help document has an associated OMF file in an OMF file directory, users can access the Help document from the Yelp browser.

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

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

ScrollKeeper Cataloging System

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

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

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

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

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

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

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

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

Improving Performance

This chapter describes how to improve the performance of the Java Desktop System.

Introduction to Improving Performance

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

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

Reducing CPU Usage

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

To Use Control Theme Options Which Require Less CPU Resources

Some control theme options can be slow on systems with limited CPU resources. To reduce CPU usage, use one of the following control options:

- Crux
- Mist
- Simple

To change the control theme option, execute the following command:

```
# gconftool-2 --type string --set
/desktop/gnome/interface/gtk_theme option-name
```

For example, to use Simple, execute the following command:

```
# gconftool-2 --type string --set
/desktop/gnome/interface/gtk_theme Simple
```

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

To Use Window Frame Theme Options Which Require Less CPU Resources

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

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

- Atlanta
- Esco
- Mist
- Simple

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

- AgingGorilla
- Bright
- Metabox

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

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

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

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

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

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

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

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

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

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

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

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

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

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

To Reduce CPU Usage by Turning Off the Splash Screen

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

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

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

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

To Reduce CPU Usage by Turning Off Panel Animation

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

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

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

Improving File Manager Performance

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

To Modify Performance Preferences

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

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

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

Preference	Description
show_icon_text	Specifies when to preview the content of text files in the icon that represents the file. To never preview the content of text files, execute the following command: # **gconftool-2 --type string --set /apps/nautilus/preferences/ show_icon_text never** Alternatively, users can perform the following steps: 1. Choose Edit → Preferences from a file manager window, then choose **Preview**. 2. Select an option for the **Show text in icons** preference.
show_directory_item_counts	Specifies when to show the number of items in folders. To never show the number of items in folders, execute the following command: # **gconftool-2 --type string --set /apps/nautilus/preferences/ show_directory_item_counts never** Alternatively, users can perform the following steps: 1. Choose Edit → Preferences from a file manager window, then choose **Preview**. 2. Select an option for the **Count number of items** preference.
show_image_thumbnails	Specifies when to show thumbnails of image files. To never show thumbnails, execute the following command: # **gconftool-2 --type string --set /apps/nautilus/preferences/ show_image_thumbnails never** Alternatively, users can perform the following steps: 1. Choose Edit → Preferences from a file manager window, then choose **Preview**. 2. Select an option for the **Show thumbnails** preference.

Preference	Description
`preview_sound`	Specifies when to preview the content of sound files. To never preview the content of sound files, execute the following command:

```
# gconftool-2 --type string --set
/apps/nautilus/preferences/preview_sound
never
```

Alternatively, users can perform the following steps:
1. Choose Edit → Preferences from a file manager window, then choose **Preview**.
2. Select an option for the **Preview sound files** preference.

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

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

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

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

To turn off the toolbar, execute the following command:

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

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

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

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

To Turn Off the Desktop Background

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

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

To disable the desktop background, execute the following command:

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

To Reduce CPU Usage by Enabling Reduced Resources Mode

The Metacity window manager has a reduced resources mode which renders windows as wireframes when you move and resize windows. When reduced resources mode is enabled, only the outline of windows is displayed when you move and resize windows. The contents of the window do not need to be updated during the move or resize. The contents of the window are displayed when the move or resize is complete. When reduced resources mode is enabled, windows are not animated when you move and resize the windows.

To enable reduced resources mode, execute the following command:

```
# gconftool-2 --type bool --set
/apps/metacity/general/reduced_resources true
```

To Use the mediaLib Libraries on Solaris Systems

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

For information about how to obtain and install the mediaLib libraries, see the mediaLib website at the following URL:

```
http://www.sun.com/processors/vis/mlib.html
```

Reducing X Window System Network Traffic

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

To Use Window Frame Theme Options Which Create Less Network Traffic

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

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

- Mist
- Simple

For information about how to change theme options, see "To Use Window Frame Theme Options Which Require Less CPU Resources" on page 72.

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

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

For information about how to turn off this feature, see "To Reduce CPU Usage by Turning Off Display of Icons in Menus" on page 73.

To Reduce Network Traffic by Enabling Reduced Resources Mode

If the application that displays the window is running on a remote host, enable reduced resources mode to reduce X Window System network traffic.

For more information about reduced resources mode, see "To Reduce CPU Usage by Enabling Reduced Resources Mode" on page 77.

Reducing Color Usage and Improving Display Quality on Solaris Systems

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

The Java Desktop System uses the *websafe color palette*. This palette is a general-purpose palette of 216 colors, which is designed to optimize the use of color on systems that support 8-bit color. However, some visual components of the Java 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 a dithered approximation. This use of replacement colors causes the grainy appearance.

- Applications that do not use the websafe color palette have less colors available. Color errors might occur. Some colors might not appear in the user interface of the application. Some applications might crash if the application cannot allocate colors.

- Color flashing might occur when users switch between applications that use the websafe color palette, and applications that do not use this palette. The applications that do not use the websafe color palette might use a custom colormap. When the custom colormap is used, other visual components might lose colors, then become unviewable.

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

To Use Theme Options That Use the Websafe Color Palette

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

For information about how to change theme options, see "To Use Window Frame Theme Options Which Require Less CPU Resources" on page 72.

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

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

For information about how to turn off this feature, see "To Reduce CPU Usage by Turning Off Display of Icons in Menus" on page 73.

To Reduce Color Usage by Turning Off the Splash Screen

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

For information about how to turn off the splash screen, see "To Reduce CPU Usage by Turning Off the Splash Screen" on page 73.

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

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

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

```
# gconftool-2 --type string --set
/desktop/gnome/background/picture_options none

# gconftool-2 --type string --set
/desktop/gnome/background/color_shading_type solid

# gconftool-2 --type string --set
/desktop/gnome/background/primary_color \#hexadecimal-color
```

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

Disabling Features in the Java Desktop System

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

Introduction to Disabling Java Desktop System Features

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

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

To Disable 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 item from the Add to Panel → Actions menu. To open this menu, right-click on a vacant space on a panel, then choose Add to Panel → Actions.
- Lock Screen menu item from the Actions menu in the Menu Bar panel application.

Also, any Lock Screen buttons on panels are disabled.

To Disable 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 panel application.

Also, any Log Out buttons on panels are disabled.

To Disable Force Quit Operations

To disable the force quit function, set the `/apps/panel/global/disable_force_quit` key to `true`.

When you disable the force quit function, the Force Quit menu item is removed from the Add to Panel → Actions menu. Also, any Force Quit buttons on panels are disabled.

To Disable Command Line Operations

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

When you disable command line operations, the following items are removed from the panels:

- Any menu items that contain the following commands:

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

  The items are removed from the following menus:

  - Main Menu
  - Add to Panel → Launcher from menu

- Command Line panel application from the Main Menu and the Add to Panel → Utility menu.

- Run Application menu item, from the following menus:

  - Main Menu
  - Actions submenu in the Add to Panel menu
  - Actions menu in the Menu Bar panel application

To Disable Panel Configuration

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

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

- The following items are removed from the panel popup menu, and from the drawer popup menu:

  - Add to Panel
  - Delete This Panel
  - Properties
  - New Panel

- The following items are removed from the popup menus for panel objects:

  - Remove From Panel.
  - Lock.
  - Move.
  - The Properties menu item is removed from the launcher popup menu.

- The Main Menu popup menu is disabled.

- The launcher drag feature is disabled, so that users cannot drag launchers to, or from, panels.

- The panel drag feature is disabled, so that users cannot drag panels to new locations.

To Restrict Applications

To restrict the applications that users can start, use the following keys:

- `/desktop/gnome/lockdown/restrict_application_launching`
- `/desktop/gnome/lockdown/allowed_applications`

To restrict the applications that users can start, set the `/desktop/gnome/lockdown/restrict_application_launching` key to `true`.

When you restrict the applications, the following changes occur in the user interface:

- The Run Application item is removed from the following menus:

  - Main Menu
  - Actions submenu in the Add to Panel menu
  - Actions menu in the Menu Bar panel application

- Any menu items that do not correspond to applications listed in the `/desktop/gnome/lockdown/allowed_applications` key are removed from the following menus:

  - Main Menu
  - Actions submenu in the Add to Panel menu
  - Launcher from menu submenu in the Add to Panel menu
  - Actions menu in the Menu Bar panel application

- When users double-click on a launcher on a panel, on a drawer, or on the desktop, a warning dialog is displayed if the application is not listed in the `/desktop/gnome/lockdown/allowed_applications` key. The application is not started.

- When users open a file whose MIME type is associated with an application that is not listed in the `/desktop/gnome/lockdown/allowed_applications` key, a warning dialog is displayed. The application is not started.

- If particular commands are not listed in the `/desktop/gnome/lockdown/allowed_applications` key, the corresponding menu items are removed from menus, as shown in the following table:

Command	Menu Item	Menu
`/usr/bin/gnome-search-tool`	Find Files	Main Menu
		Actions menu in Menu Bar panel application
		Add to Panel → Actions
`/usr/bin/yelp`	Help	Main Menu

Command	Menu Item	Menu
`/usr/bin/gnome-panel-screenshot`	Screenshot	Add to Panel → Actions

Hidden Directories

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

A hidden directory is a directory that has a name that begins with a period (.). Table A–1 describes the hidden directories that the GNOME Desktop adds to the home directories of users.

TABLE A–1 Hidden Directories in User Home Directories

Directory	Description
`.esd_auth`	Contains the authentication cookie for the GNOME sound daemon. The GNOME sound daemon is the Enlightened Sound Daemon (ESD).
`.gconf`	Contains the GConf configuration source for the user. When the user sets a preference, the new preference information is added to this location.
`.gconfd`	Contains the following GConf daemon details: ■ Configuration information. ■ Lock information for objects that are referenced by an *Interoperable Object Reference (IOR)*. ■ State information for objects that are referenced by an IOR.
`.gnome`	Contains user-specific application data that is not stored in the GConf repository. For example, this directory contains MIME type information and session information for the user.

87

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

Directory	Description
`.gnome2`	Contains user-specific application data that is not stored in the GConf repository. For example, this directory contains the following: ■ Keyboard shortcut information. ■ Window location information. ■ Desktop entry files for panel launchers. This directory also contains user-specific menu data. If a user modifies menus, the details are stored here.
`.gnome2-private`	Ignore this directory. This directory has no function at the time of publication of this guide.
`.metacity`	Contains session data for the Metacity window manager.
`.nautilus`	Contains file manager data that is specific to the user. For example, this directory can contain the following: ■ Metadata for the directories with which the user works. ■ Nautilus emblems that the user adds. ■ Nautilus desktop background images.
`.themes`	Contains controls theme options, window frame theme options, and icons theme options that the user adds. The user can add themes from the Theme preference tool.
`.thumbnails`	Contains image thumbnails for the user. The image thumbnails are used in the file manager. The file manager contains a preference that the user can select to stop generation of thumbnail images.
`.xscreensaver`	Contains screensaver configuration data and screensaver preference data.

Glossary

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

GConf path file	A file that lists the GConf configuration sources, and the order in which to search the sources.
GConf schema	A collective term for a schema key and a schema object.
GConf schema definition file	A GConf schema definition file lists the keys in a particular application, and defines the characteristics of the keys. GConf schemas are generated from schema definition files. Schema definition files have a `.schemas` 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 on the preference key.
Interoperable Object Reference	An Interoperable Object Reference (IOR) is a string reference to a CORBA object. An IOR encodes a hostname and port to which messages can be sent to control the object. The IOR also contains an object key to identify the object.
MIME	Multipurpose Internet Mail Extension.
MIME 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, and with file extensions and filename patterns.
NFS	Network File System.
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.
PAM	Pluggable Authentication Modules.
pattern mask	A pattern mask is a series of hexadecimal characters in a file content sniffer. The pattern mask identifies bits in the pattern to ignore when searching for a pattern in a file.
screensaver	A screensaver is an application that replaces the image on a screen when the screen is not in use. The screensaver application for the Java Desktop System is XScreenSaver.

screensaver display	A screensaver display is an application that displays images on the screen of the user when the screen is not in use.
SMB	Server Message Block.
UDP	User Datagram Protocol.
Uniform Resource Identifier	A Uniform Resource Identifier (URI) is a string that identifies a particular location in a file system or on the web. For example, the address of a web page is a URI.
vfolder	A virtual representation of items that reside in a physical location or physical locations on your system. For example, a vfolder might represent the contents of several directories. In terms of menus, a vfolder is a representation in a menu of items that might be physically located in several directories.
vfolder information file	An XML file that describes a vfolder. Vfolder information files specify the structure of your menus.
websafe color palette	The websafe color palette is a general-purpose palette of 216 colors. The websafe color palette is designed to optimize the use of color on systems that support 8-bit color. The websafe color palette is also called the *Netscape color palette* and the *Netscape color cube*.

Index