

## Using hidden Gnome settings

# Safety-Locks Off

“Where have all the options gone?

Long time passing”. This is what many Gnome users may be asking themselves, echoing the old Pete Seeger classic. Most options are hidden away in the GConf editor.

This article shows you where.

BY ANDREA MÜLLER

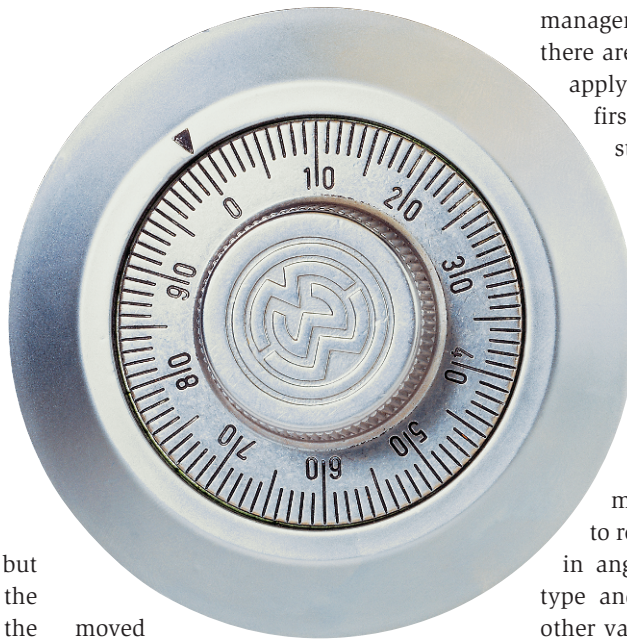
A bit chaotic and disorganized, but highly configurable – this is the constellation that made the Gnome desktop so popular with hackers and geeks in the past. Version 2.0 saw a big turn-around. The Gnome project set off to tackle the myriad of options and implement the Human Interface Guidelines (see Box 1) to improve usability.

## Collateral Damage

Although their goals may be laudable, many experienced users feel the Gnome developers have gone too far. If you use the Nautilus file manager on another desktop, you will know the issue that the file manager insists on drawing the desktop. Users could disable the *Use Nautilus to draw the desktop* option. Today, if you launch Nautilus without paying attention, you get beamed to Gnome.

Even if you stipulate the `--no-desktop` option in the command line, you may still be in for a nasty surprise. Nautilus is often registered as the *Help Viewer* for Gnome programs, and an innocent call to *Help / Content* in a Gnome application running on KDE, may be all it takes to trip up the unsuspecting user.

Fortunately, the developers have not discarded the feature. It has simply been



moved out of the menu and into the GConf editor. This is where to look for any configuration options.

Red Hat Linux is the only distro to install the practical tool by default. Mandrake 9.0 and 9.1 at least install the tool along with the *gnome2* metapackage. Suse Linux users have no option but to install manually, even though they have Gnome on their hard disks.

## Almost Like Regedit

When you launch the configuration tool, by entering `gconf-editor &`, don't let the warning shown in Figure 2 discourage you. The program is stable, and if you modify an option by mistake, you can always revert to the default setting.

The tree structure on the left-hand side contains so-called keys and subkeys. The values are shown on the right. Any changes are applied immediately, and written to the user's configuration database. The database is XML formatted and stored below `~/gconf`.

Keys that contain values will be stored in individual files. `~/gconf/apps/nautilus/preferences/%gconf.xml` contains the current settings for the Gnome file

manager. In addition to user databases, there are system global preferences that apply for users running Gnome the first time. The global defaults are stored in `/etc/gconf/gconf.xml.defaults`. Suse version 9 does its own thing, and stores the defaults in `/etc/opt/gnome/gconf/gconf.xml.defaults`. Suse 8.2 used the `/etc/opt/gnome2/gconf` directory. If you compiled Gnome yourself, look for the defaults in `/usr/local/etc/gconf`.

Although XML is a text format, XML documents are difficult to read. Each setting comprises a tag in angled brackets that includes the type and value of the setting, among other values (see Figure 3). The path in the XML file always reflects the path used by the GConf editor. For example, the graphical interface to the Nautilus preferences is stored in the `/apps/nautilus/preferences` key.

The item responsible for drawing the desktop is called `show_desktop`. You can remove the checkmark, to tell the file manager to stop drawing this. Close the GConf editor to apply your modifications to the appropriate XML file.

## Reinstating the Tools Menu

The `apps` and `desktop` keys are typical starting points from which to locate hidden keys. `apps` is where you configure the way Gnome applications behave. In addition to the core applications that come with the desktop environment, there are a number of optional Gnome programs, such as the HTML editor, screem. The `desktop` key is a kind of catchment basin for global (application independent) desktop defaults.

It is often difficult to match up keys to features. The key description can be a big help. The `desktop/gnome/background/picture_filename` key specifies the background image that Gnome uses.

In contrast to the `show_desktop` option, this is not a simple yes/no flip-

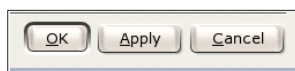


Figure 1: Two of the three buttons in this dialog box are redundant.

flop; instead, you are expected to specify the pathname to the required image file. Click *picture\_filename* in the *Value* column and type the filename when prompted (see Figure 4). The alternative is to click an entry. This tells the GConf editor to open a settings dialog box where you can modify the value of an entry (see Figure 5).

If the option name is not self-explanatory, it may have a docstring assigned, a short description of what the option does. If there is a docstring, the GConf editor will display it on the lower right-hand side of the screen (see Figure 6).

You can browse the tree structure to access options that are not available in the application menus. Table 1 contains a selection. You can assign bookmarks to values that often change by selecting *Bookmark / Add bookmark*. You can access it in future via *Bookmark / Keyname*.

### Three Make a Team

*gconf-editor* does not write the options to the configuration database itself. Two helpers take care of this. One of these is the command line-based *gconftool-2*, the other the configuration server, *gconfd*. The editor provides a graphical front-end for *gconftool-2* and passes your changes to *gconfd*. *gconfd* takes your changes and informs the active programs of them. For example, the desktop drawn by Nautilus disappears immediately, without you having to re-launch the file manager.

### Box 1: Improved Usability with the Human Interface Guidelines

The Human Interface Guidelines at <http://developer.gnome.org/projects/gup/hig/specify-software-ergonomics-requirements> that all Gnome applications are expected to fulfill. These include consistent control layouts, well-organized menus, translations into as many languages as possible, and doing without features that the majority of users will never need, or even understand.

The HIG define usability rules stating that software should choose language appropriate to the typical user's vocabulary, and – if the current operation will take a while – provide

feedback to tell the user what the program is doing. The guidelines provide details on the layout of dialog boxes and menus; this is why you are unlikely to see Gnome applications with dialog boxes that include *OK*, *Apply/Use* and *Close* buttons.

When a user makes a change, as shown in Figure 1, one can assume that the user will want to apply the change immediately. This makes separate *Apply* or *Use* buttons redundant. Who needs an *OK* function? Users will expect the *Close* button to do exactly the same thing.

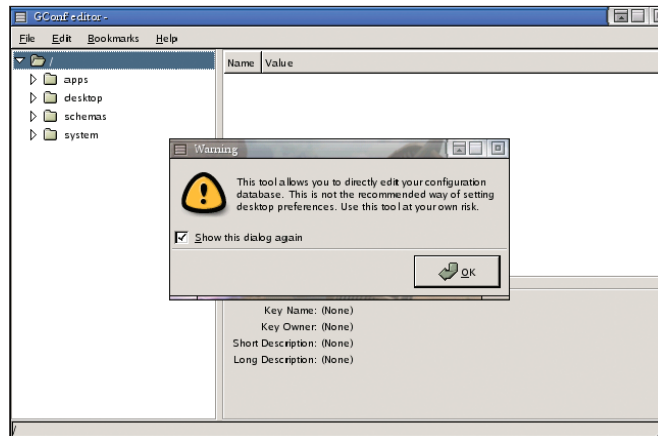


Figure 2: Use at your own peril! Or so the GConf editor says.



Figure 3: Gnome uses XML files of this kind to store its configuration data.

You can use *gconftool-2* to change options in the command line. The tool runs on systems where the GUI-based front-end is not installed. It can also be your rescue tool in cases where the configuration database has been destroyed by a power failure or system crash and is preventing Gnome from launching.

### Shell-Based Configuration

*gconftool-2* provides such a multitude of options that it is not intuitive to use. Table 2, and a few examples, should help demonstrate *gconftool-2*'s capabilities.

Calls follow this pattern:

```
gconftool-2 option key
```

where *key* is the path you would access with Gconf editor, but without a terminating forward slash /. If you need to change, rather than just display a key, you will need to supply a value. Type the key just like you would type a path (in fact, keys are just pathnames in the configuration database), such as */apps/nautilus/preferences/show-desktop*. You can type *gconftool-2 --all-dirs* to display a list of paths:

```
/desktop
/apps
/system
/schemas
```

The *-a* parameter shows you the settings below a key. *gconftool-2 -a /apps/file-roller/listing* will show the current display settings for the *file\_roller* archiving program. If you need to export the subkeys and settings for an application branch to a file for reference purposes, use the following:

```
gconftool-2 -R /apps >
> apps-optionstxt
```

You can then print the *apps-options.txt* file created by this example, or simply browse the file on screen (see Figure 7).

The *--long-docs*, *--short-docs*, and *-T* options display the details of a specific

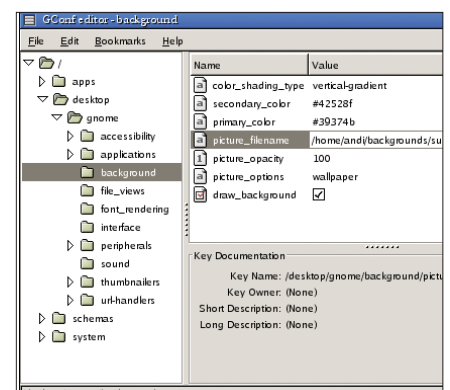


Figure 4: Checkboxes and input fields allow you to modify values.

setting. The first two print help texts for a key, if available. The *-T* flag tells you the type of a specific value; you will need this information if you want to edit a value.

As a rule of thumb, you can assume that pathnames will be *string* types, while true/false values will be *bool* types. We looked at an example earlier: the option that specifies whether Nautilus should draw the desktop. There are also *int* (integers), *float* (floating point numbers) and *list* types.

`/apps/panel/default_setup/general/toplevel_id_list` is an example of a list. In this case it tells you the panels that Gnome will launch.

## Ring in the Changes

After discovering the name and type of a key, and you know exactly what it does, it's time for a change. For example, you can enter the following to tell Nautilus to stop drawing the desktop:

```
gconftool-2 -t bool -s /apps
/nautilus/preferences
/show_desktop false
```

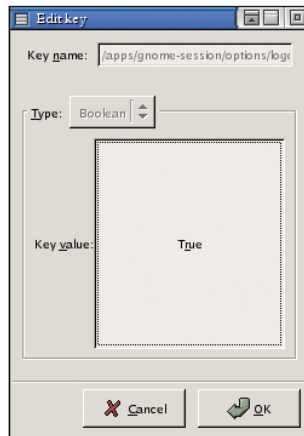


Figure 5: Click to toggle between “true” and “false”.

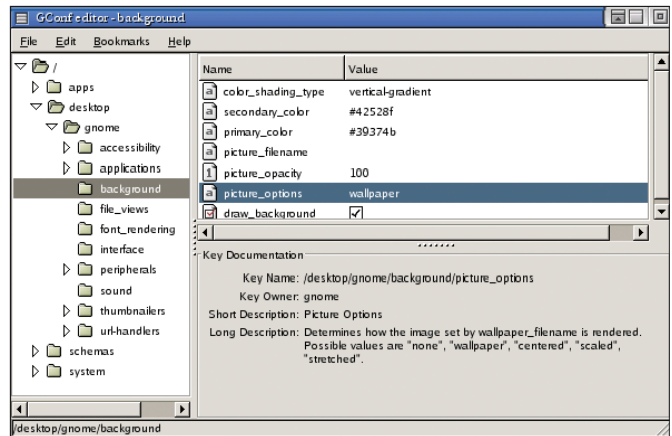


Figure 6: The docstring provides a list of the values for integrating a background image.

The *-t bool* parameter tells *gconftool-2* what kind of value this is. *-s* indicates that you want to set an option, and the appropriate value is *false* in this case. Specify *true* instead of *false* to reinstate this feature. The following command:

```
gconftool-2 -t int -s /apps
/nautilus/preferences
/directory_limit 10000
```

increases the maximum number of files that Nautilus can display to 10000 (Gnome 2.4 does not apply a restriction

by default, Gnome 2.2 has a maximum limit of 4000 files). The next command:

```
gconftool-2 -t string /apps
/gnome-session/options
/splash_image/home/andi
/mysplash.png
```

makes Gnome use another splashscreen.

If it is even simpler to reset an option to the default, as you do not need to specify a value or type in this case. *gconftool-2 -u* and the appropriate key will do the trick.

If you have made a mess of the settings for a program, there is no need to repeat the command for each option. The *--recursive-unset* parameter resets all the keys in a branch to the defaults.

```
gconftool-2 --recursive-unset
/apps/nautilus
```

will reset all the file manager values to the defaults. This also works for the desktop properties:

### Table 1: Secret Gnome Settings

Keyname in GConf editor	Effect
<code>/apps/file-roller/listing/show_mime_icons</code>	Enable/disable file type icons in the File Roller archiving program. When this is disabled, each file in the archive is assigned a plain white symbol.
<code>/apps/file-roller/ui/history_len</code>	Number of files in the File Roller history menu.
<code>/apps/gedit-2/preferences/editor/save/save_encoding</code>	Specifies the codeset to be used by the gedit editor. Defaults to UTF-8. The <code>GEDIT_SAVE_ORIGINAL_FILE_ENCODING_IF_POSSIBLE</code> option tells gedit to save only new files in UTF-8, and to keep the original encoding in all other cases. This is useful if users with whom you exchange files complain about illegible characters in your files.
<code>/apps/gedit-2/preferences/ui/recents/max_recents</code>	Number of recent files shown in the editor menu.
<code>/apps/gnome-search-tool/show_additional_options</code>	Displays extended search options when the search function is launched (Gnome 2.4 or later).
<code>/apps/nautilus/desktop/home_icon_visible</code>	Specifies whether to display a home directory icon on the desktop (Gnome 2.4 or later).
<code>/apps/nautilus/desktop/trash_icon_visible</code>	Enable or disable the trashcan display (Gnome 2.4 or later).
<code>/apps/panel/global/confirm_panel_remove</code>	Disable confirm prompt when removing a panel.
<code>/apps/gnome-session/options/show_splash_screen</code>	Specifies whether Gnome should display a splash screen on starting up.
<code>/apps/gnome-session/options/splash_image</code>	Requires the path to an image file as a value. Gnome will display this image as the splash screen.
<code>/desktop/gnome/file-views/show_backup_files</code>	In contrast to the menu, this displays only files with filenames ending in a tilde (~). Many programs use files of this kind as backup copies. Files with names starting with a dot will remain hidden.
<code>/desktop/gnome/file-views/show_hidden_files</code>	The same feature for hidden configuration files that start with a dot.

## GLOSSARY

**UTF-8:** Character code that includes the majority of alphabets used worldwide. Applications that can handle Unicode encoding are capable of displaying Cyrillic, Asian and accented characters without the user having to specify a character set specific to a certain alphabet. The application simply needs the right font to be able to display a UTF-8 encoded text.

**Splash screen:** A welcome screen shown by an application on launching. The screen typically includes a logo and version information. Apart from GNOME, KDE, Quanta, OpenOffice, and the nVidia drivers include splash screens.

```
gconftool-2 --recursive-unset ?
/desktop
```

Note that this will remove all your preferences. Before you take this step, make sure that you copy your `/home/username/.gconf` to a safe place. This will allow you to restore it, if needed.

## On a Short Leash

The `--config-source` and `--direct` parameters, which we have not looked at so far, make life a lot easier for system administrators. They allow you to modify the global configuration database in the `/etc/gconf/gconf.xml.defaults/` directory. This makes sense if you use KDE as your desktop environment and want to avoid unsuspecting users wrecking their desktops, simply by launching Nautilus.

Make sure that `gconfd` is not running while modifying the default database, as this step involves editing the XML files directly without using the server. Terminate any Gnome applications and use `top`, or `ps -e | grep gconfd`, to double-check that the server is not running. It can take a while for the server to quit.

You can then modify a global setting:

```
gconftool-2 --direct -config?
-source xml:readwrite:/etc?
/gconf/gconf.xml.defaults/ -t ?
bool -s /apps/nautilus?
/preferences/show_desktop false
```

Specifying the `xml:readwrite:` parameter for the configuration directory tells `gcon-`

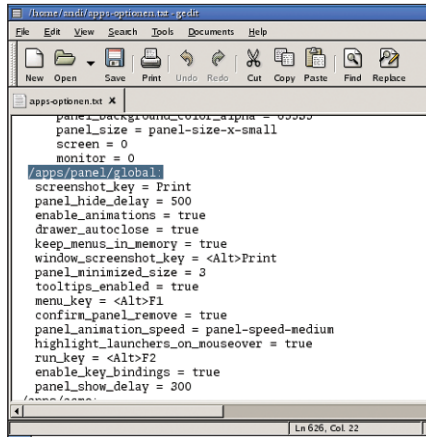


Figure 7: The full set of application options at a glance.

`ftool-2` to open the XML database for read/write access. To simply query a value, type the following command instead:

```
gconftool-2 --direct -config?
-source xml:readonly:/etc/gconf?
/gconf.xml.defaults/ -g /apps?
/nautilus/preferences?
/show_desktop
```

This is a command that unprivileged users can run. On Suse Linux 9, the path to the configuration database is `/etc/opt/gnome/gconf/gconf.xml.defaults`. Users of Suse Linux 8.2 will need to replace `gnome` with `gnome2`.

Unprivileged users can make the same changes to their local database in their `home` directories as the root user makes globally. If you want to disable this privi-

lege for a specific option, to avoid the help calls typically provoked by changing defaults, there is a way to enforce this. It involves a third database, `/etc/gconf/gconf.xml.mandatory/`, which stores mandatory values that users cannot change in their personal preferences.

If you use Gnome as your standard desktop, and want to prevent users from changing Nautilus's draw background function, try the following command:

```
gconftool-2 --direct -config?
-source xml:readwrite:/etc?
/gconf/gconf.xml.mandatory?
/ -t bool -s /apps/nautilus?
/preferences/show_desktop true
```

The GConf editor will still display the option, but unprivileged users will no longer be able to modify it. Multiple clicks on checkboxes caused the configuration tool to crash while working on Gnome 2.2.

In the case of options available in application menus, these restrictions only apply if the user has not yet modified the value – you might like to reset the user preferences beforehand, using the `-u` parameter. You can then remove the feature that displays a true remove command in the Nautilus dropdown menu, as follows:

```
gconftool-2 --direct -config?
-source xml:readwrite:/etc?
/gconf/gconf.xml.mandatory?
/ -t bool -s /apps/nautilus?
/preferences/enable_delete false
```

Users who attempt to enable this feature will be treated to an error message (see Figure 8).

If this has whetted your appetite, you might like to check out the examples and the detailed reference on the Gnome configuration system in the Desktop / System Administration Guide / Using GConf. ■

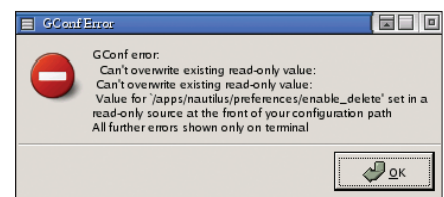


Figure 8: Users are not allowed to remove files but can still trash them.

Table 2: Command line parameters for `gconftool-2`

Option	Argument	Effect
<code>-s</code>	Key name Value	Sets a value; requires <code>-t</code> parameter.
<code>-g</code>	Key name	Queries a value.
<code>-u</code>	Key name	Resets an entry.
<code>--recursive-unset</code>	Key	Resets all the values below this key.
<code>-a</code>	Key	Outputs all the key/value pairs below this key.
<code>--all-dirs</code>	Key	Lists the folders below this key.
<code>-R</code>	Key	Displays a recursive list of all folders and entries below this key.
<code>-t</code>	Type	Sets the file type to the required value – <i>int, string, bool, float, list, or pair</i> .
<code>-T</code>	Key name	Queries the type of a value.
<code>--short-docs, --long-docs</code>	Key name	Outputs the short or detailed documentation for an option.
<code>--config-source</code>	Configuration database	Specifies the configuration database that the query or set operation applies to. Normal users only have read access to global configurations.
<code>--direct</code>		Modifies the database without using the server. This is required to modify any configuration apart from your own configuration below <code>~/.gconf</code> ; avoid running <code>gconfd</code> while making these changes.