

GNOME Display Manager Reference Manual

1. [Terms and Conventions Used in This Manual](#)
2. [Overview](#)
3. [Security](#)
4. [Support for ConsoleKit](#)
5. [Configuration](#)
6. [GDM Commands](#)
7. [Troubleshooting](#)
8. [Licence](#)

1. Terms and Conventions Used in This Manual

This manual describes version 2.26.0 of the GNOME Display Manager. It was last updated on 02/10/2009.

Chooser - A program used to select a remote host for managing a display remotely on the attached display (`gdm-host-chooser`).

FreeDesktop - The organisation providing desktop standards, such as the Desktop Entry Specification used by GDM. <http://www.freedesktop.org>.

GDM - GNOME Display Manager. Used to describe the software package as a whole.

Greeter - The graphical login window (`gdm-simple-greeter`).

PAM - Pluggable Authentication Mechanism

XDMCP - X Display Manage Protocol

Xserver - An implementation of the X Window System. For example the Xorg webserver provided by the X.org Foundation <http://www.x.org>.

Paths that start with a word in angle brackets are relative to the installation prefix. I.e. `<share>/pixmaps/` refers to `/usr/share/pixmaps` if GDM was configured with `--prefix=/usr`.

2. Overview

- 2.1. [Introduction](#)
- 2.2. [Interface Stability](#)
- 2.3. [Functional Description](#)
- 2.4. [Greeter Panel](#)
- 2.5. [Accessibility](#)
- 2.6. [The GDM Face Browser](#)
- 2.7. [XDMCP](#)
- 2.8. [Logging](#)
- 2.9. [Fast User Switching](#)

2.1. Introduction

The GNOME Display Manager (GDM) is a display manager that implements all significant features required for managing attached and remote displays. GDM was written from scratch and does not contain any XDM or X Consortium code.

Note that GDM is configurable, and many configuration settings have an impact on security. Issues to be aware of are highlighted in this document.

Please note that some Operating Systems configure GDM to behave differently than the default values as described in this document. If GDM does not seem to behave as documented, then check to see if any related configuration may be different than described here.

For further information about GDM, refer to the project website at <http://www.gnome.org/projects/gdm> and the project Wiki <http://live.gnome.org/GDM>.

For discussion or queries about GDM, refer to the

[<gdmlist@gnome.org>](mailto:gdmlist@gnome.org)

mail list. This list is archived, and is a good resource to check to seek answers to common questions. This list is archived at <http://mail.gnome.org/archives/gdm-list/> and has a search facility to look for messages with keywords.

Please submit any bug reports or enhancement requests to the "gdm" category in <http://bugzilla.gnome.org>.

2.2. Interface Stability

GDM 2.20 and earlier supported stable configuration interfaces. However, the codebase was completely rewritten for GDM 2.22, and is not completely backward compatible with older releases. This is in part because things work differently, so some options just don't make sense, in part because some options never made sense, and in part because some functionality has not been reimplemented yet.

Interfaces which continue to be supported in a stable fashion include the Init, PreSession, PostSession, PostLogin, and Xsession scripts. Some daemon configuration options in the `<etc>/gdm/custom.conf` file continue to be supported. Also, the `~/.dmrc`, and face browser image locations are still supported.

GDM 2.20 and earlier supported the ability to manage multiple displays with separate graphics cards, such as used in terminal server environments, login in a window via a program like Xnest or Xephyr, the `gdmsetup` program, XML-based greeter themes, and the ability to run the XDMCP chooser from the login screen. These features were not added back during the 2.22 rewrite.

2.3. Functional Description

GDM is responsible for managing displays on the system. This includes authenticating users, starting the user session, and terminating the user session. GDM is configurable and the ways it can be configured are described in the "Configuring GDM" section of this document. GDM is also accessible for users with disabilities.

GDM provides the ability to manage the main console display, and displays launched via VT. It is integrated with other programs, such as the Fast User Switch Applet (FUSA) and `gnome-`

screensaver to manage multiple displays on the console via the Xserver Virtual Terminal (VT) interface. It also can manage XDMCP displays.

Regardless of the display type, GDM will do the following when it manages the display. It will start an Xserver process, then run the `Init` script as the root user, and start the greeter program on the display.

The greeter program is run as the unprivileged "gdm" user/group. This user and group are described in the "Security" section of this document. The main function of the greeter program is to authenticate the user. The authentication process is driven by Pluggable Authentication Modules (PAM). The PAM modules determine what prompts (if any) are shown to the user to authenticate. On the average system, the greeter program will request a username and password for authentication. However some systems may be configured to use alternative mechanisms such as a fingerprint or SmartCard reader. GDM and PAM can be configured to not require any input, which will cause GDM to automatically log in and simply start a session, which can be useful for some environments, such as for kiosks.

In addition to authentication, the greeter program allows the user to select which session to start and which language to use. Sessions are defined by files that end in the `.desktop` suffix and more information about these files can be found in the "asking for pConfiguration" section of this document. By default, GDM is configured to display a face browser so the user can select their user account by clicking on an image instead of having to type their username. GDM keeps track of the user's default session and language in the user's `~/.dmmrc` and will use these defaults if the user did not pick a session or language in the login GUI.

After authenticating a user, the daemon runs the `PostLogin` script as root, then runs the `PreSession` script as root. After running these scripts, the user session is started. When the user exits their session, the `PostSession` script is run as root. These scripts are provided as hooks for distributions and end-users to customise how sessions are managed. For example, using these hooks you could set up a machine which creates the user's `$HOME` directory on the fly, and erases it on logout. The difference between the `PostLogin` and `PreSession` scripts is that `PostLogin` is run before the `pam_open_session` call so is the right place to do anything which should be run before the user session is initialised. The `PreSession` script is called after session initialisation.

2.4. Greeter Panel

The GDM greeter program displays a panel docked at the bottom of the screen which provides additional functionality. When a user is selected, the panel allows the user to select which session, language, and keyboard layout to use after logging in. The keyboard layout selector also changes the keyboard layout used when typing your password. The panel also contains an area for login services to leave status icons. Some example status icons include a battery icon for current battery usage, and an icon for enabling accessibility features. The greeter program also provides buttons which allow the user to shutdown or restart the system. It is possible to configure GDM to not provide the shutdown and restart buttons, if desired. GDM can also be configured via PolicyKit (or via RBAC on Solaris) to require the user have appropriate authorisation before accepting the shutdown or restart request.

Note that keyboard layout features are only available on systems that support libxklavier.

2.5. Accessibility

GDM supports "Accessible Login", allowing users to log into their desktop session even if they cannot easily use the screen, mouse, or keyboard in the usual way. Accessible Technology (AT) features such as an on-screen keyboard, screen reader, screen magnifier, and Xserver AccessX keyboard accessibility are available. It is also possible to enable large text or high contrast icons and controls, if needed. Refer to the "Accessibility Configuration" section of the document for more information how various accessibility features can be configured.

On some Operating Systems, it is necessary to make sure that the GDM user is a member of the "audio" group for AT programs that require audio output (such as text-to-speech) to be functional.

2.6. The GDM Face Browser

The Face Browser is the interface which allows users to select their username by clicking on an image. This feature can be enabled or disabled via the `/apps/gdm/simple-greeter/disable_user_list` GConf key and is on by default. When disabled, users must type their complete username by hand. When enabled, it displays all local users which are available for login on the system (all user accounts defined in the `/etc/passwd` file that have a valid shell and sufficiently high UID) and remote users that have recently logged in. The face browser in GDM 2.20 and earlier would attempt to display all remote users, which caused performance problems in large, enterprise deployments.

The Face Browser is configured to display the users who log in most frequently at the top of the list. This helps to ensure that users who log in frequently can quickly find their login image.

The Face Browser supports "type-ahead search" which dynamically moves the face selection as the user types to the corresponding username in the list. This means that a user with a long username will only have to type the first few characters of the username before the correct item in the list gets selected.

The icons used by GDM can be installed globally by the sysadmin or can be located in the user's home directories. If installed globally they should be in the `<share>/pixmap/faces/` directory and the filename should be the name of the user. Face image files should be a standard image that GTK+ can read, such as PNG or JPEG. Face icons placed in the global face directory must be readable to the GDM user.

If there is no global icon for the user, GDM will look in the user's `$HOME` directory for the image file. GDM will first look for the user's face image in `~/ .face`. If not found, it will try `~/ .face.icon`. If still not found, it will use the value defined for "face/picture=" in the `~/ .gnome2/gdm` file.

If a user has no defined face image, GDM will use the "stock_person" icon defined in the current GTK+ theme. If no such image is defined, it will fallback to a generic face image.

Please note that loading and scaling face icons located in remote user home directories can be a very time-consuming task. Since it not practical to load images over NIS or NFS, GDM does not attempt to load face images from remote home directories.

When the browser is turned on, valid usernames on the computer are exposed for everyone to see. If XDMCP is enabled, then the usernames are exposed to remote users. This, of course, limits security somewhat since a malicious user does not need to guess valid usernames. In some very restrictive environments the face browser may not be appropriate.

2.7. XDMCP

The GDM daemon can be configured to listen for and manage X Display Manage Protocol (XDMCP) requests from remote displays. By default XDMCP support is turned off, but can be enabled if desired. If GDM is built with TCP Wrapper support, then the daemon will only grant access to hosts specified in the GDM service section in the TCP Wrappers configuration file.

GDM includes several measures making it more resistant to denial of service attacks on the XDMCP service. A lot of the protocol parameters, handshaking timeouts, etc. can be fine tuned. The default configuration should work reasonably on most systems.

GDM by default listens for XDMCP requests on the normal UDP port used for XDMCP, port 177, and will respond to QUERY and BROADCAST_QUERY requests by sending a WILLING packet to the originator.

GDM can also be configured to honour INDIRECT queries and present a host chooser to the remote display. GDM will remember the user's choice and forward subsequent requests to the chosen manager. GDM also supports an extension to the protocol which will make it forget the redirection once the user's connection succeeds. This extension is only supported if both daemons are GDM. It is transparent and will be ignored by XDM or other daemons that implement XDMCP.

If XDMCP seems to not be working, make sure that all machines are specified in `/etc/hosts`.

Refer to the "Security" section for information about security concerns when using XDMCP.

2.8. Logging

GDM uses syslog to log errors and status. It can also log debugging information, which can be useful for tracking down problems if GDM is not working properly. This can be enabled by starting the GDM daemon with the "--debug" option.

Output from the various Xservers is stored in the GDM log directory, which is normally `<var>/log/gdm/`. Any Xserver messages are saved to a file associated with the display value, `<display>.log`.

The session output is piped through the GDM daemon to the `~/.xsession-errors` file. The file is overwritten on each login, so logging out and logging back into the same user via GDM will cause any messages from the previous session to be lost.

Note that if GDM can not create this file for some reason, then a fallback file will be created named `~/.xsession-errors.XXXXXXXX` where the `XXXXXXX` are some random characters.

2.9. Fast User Switching

GDM allows multiple users to be logged in at the same time. After one user is logged in, additional users can log in via the User Switcher on the GNOME Panel, or from the "Switch User" button in Lock Screen dialog of GNOME Screensaver. The active session can be changed back and forth using the same mechanism. Note that some distributions may not add the User Switcher to the default panel configuration. It can be added using the panel context menu.

Note this feature is available on systems that support Virtual Terminals. This feature will not function if Virtual Terminals is not available.

3. Security

- 3.1. [The GDM User and Group](#)
- 3.2. [PAM](#)
- 3.3. [utmp and wtmp](#)
- 3.4. [Xserver Authentication Scheme](#)
- 3.5. [XDMCP Security](#)
- 3.6. [XDMCP Access Control](#)
- 3.7. [Firewall Security](#)
- 3.8. [PolicyKit](#)
- 3.9. [RBAC \(Role Based Access Control\)](#)

3.1. The GDM User and Group

For security reasons a dedicated user and group ID are recommended for proper operation. This user and group are normally "gdm" on most systems, but can be configured to any user or group. All GDM GUI programs are run as this user, so that the programs which interact with the user are run in a sandbox. This user and group should have limited privileges.

The only special privilege the "gdm" user requires is the ability to read and write Xauth files to the `<var>/run/gdm` directory. The `<var>/run/gdm` directory should have root:gdm ownership and 1777 permissions.

You should not, under any circumstances, configure the GDM user/group to a user which a user could easily gain access to, such as the user `nobody`. Any user who gains access to an Xauth key can snoop on and control running GUI programs running in the associated session or perform a denial-of-service attack on it. It is important to ensure that the system is configured properly so that only the "gdm" user has access to these files and that it is not easy to login to this account. For example, the account should be setup to not have a password or allow non-root users to login to the account.

The GDM greeter configuration is stored in GConf. To allow the GDM user to be able to write configuration, it is necessary for the "gdm" user to have a writable \$HOME directory. Users may configure the default GConf configuration as desired to avoid the need to provide the "gdm" user with a writable \$HOME directory. However, some features of GDM may be disabled if it is unable to write state information to GConf configuration.

3.2. PAM

GDM uses PAM for login authentication. PAM stands for Pluggable Authentication Module, and is used by most programs that request authentication on your computer. It allows the administrator to configure specific authentication behavior for different login programs (such as SSH, login GUI, screensaver, etc.)

PAM is complicated and highly configurable, and this documentation does not intend to explain this in detail. Instead, it is intended to give an overview of how PAM configuration relates with GDM, how PAM is commonly configured with GDM, and known issues. It is expected that a person needing to do PAM configuration would need to do further reading of PAM documentation to understand how to configure PAM and to understand terms used in this section.

PAM configuration has different, but similar, interfaces on different Operating Systems, so check the [pam.d](#) or [pam.conf](#) man page for details. Be sure you read the PAM documentation and are

comfortable with the security implications of any changes you intend to make to your configuration.

Note that, by default, GDM uses the "gdm" PAM service name for normal login and the "gdm-autologin" PAM service name for automatic login. These services may not be defined in your `pam.d` or `pam.conf` configured file. If there is no entry, then GDM will use the default PAM behavior. On most systems this should work fine. However, the automatic login feature may not work if the `gdm-autologin` service is not defined.

The `PostLogin` script is run before `pam_open_session` is called, and the `PreSession` script is called after. This allows the system administrator to add any scripting to the login process either before or after PAM initialises the session.

If you wish to make GDM work with other types of authentication mechanisms (such as a fingerprint or SmartCard reader), then you should implement this by using a PAM service module for the desired authentication type rather than by trying to modify the GDM code directly. Refer to the PAM documentation on your system. How to do this is frequently discussed on the

[<gdmlist@gnome.org>](mailto:gdmlist@gnome.org)

mail list, so you can refer to the list archives for more information.

PAM does have some limitations regarding being able to work with multiple types of authentication at the same time, like supporting the ability to accept either SmartCard and the ability to type the username and password into the login program. There are techniques that are used to make this work, and it is best to research how this problem is commonly solved when setting up such a configuration.

If automatic login does not work on a system, check to see if the "gdm-autologin" PAM stack is defined in the PAM configuration. For this to work, it is necessary to use a PAM module that simply does no authentication, or which simply returns `PAM_SUCCESS` from all of its public interfaces. Assuming your system has a `pam_allow.so` PAM module which does this, a PAM configuration to enable "gdm-autologin" would look like this:

```
gdm-autologin auth    required    pam_unix_cred.so.1
gdm-autologin auth    sufficient  pam_allow.so.1
gdm-autologin account sufficient  pam_allow.so.1
gdm-autologin session sufficient  pam_allow.so.1
gdm-autologin password sufficient  pam_allow.so.1
```

The above setup will cause no `lastlog` entry to be generated. If a `lastlog` entry is desired, then use the following for the session:

```
gdm-autologin session required pam_unix_session.so.1
```

If the computer is used by several people, which makes automatic login unsuitable, you may want to allow some users to log in without entering their password. This feature can be enabled as a per-user option in the `users-admin` tool from the `gnome-system-tools`; it is achieved by checking that the user is member a Unix group called "nopasswdlogin" before asking for a password. For this to work, the PAM configuration file for the "gdm" service must include a line such as:

```
gdm auth    sufficient  pam_succeed_if.so  user ingroup nopasswdlogin
```

3.3. utmp and wtmp

GDM generates utmp and wtmp User Accounting Database entries upon session login and logout. The utmp database contains user access and accounting information that is accessed by commands such as `finger`, `last`, `login`, and `who`. The wtmp database contains the history of user access and accounting information for the utmp database. Refer to the [utmp](#) and [wtmp](#) man pages on your system for more information.

3.4. Xserver Authentication Scheme

Xserver authorization files are stored in a newly created subdirectory of `<var>/run/gdm` at start up. These files are used to store and share a "password" between X clients and the Xserver. This "password" is unique for each session logged in, so users from one session can't snoop on users from another.

GDM only supports the MIT-MAGIC-COOKIE-1 Xserver authentication scheme. Normally little is gained from the other schemes, and no effort has been made to implement them so far. Be especially careful about using XDMCP because the Xserver authentication cookie goes over the wire as clear text. If snooping is possible, then an attacker could simply snoop your authentication password as you log in, regardless of the authentication scheme being used. If snooping is possible and undesirable, then you should use ssh for tunneling an X connection rather than using XDMCP. You could think of XDMCP as a sort of graphical telnet, having the same security issues. In most cases, ssh -Y should be preferred over GDM's XDMCP features.

3.5. XDMCP Security

Even though your display is protected by cookies, XEvents and keystrokes typed when entering passwords will still go over the network in clear text. It is trivial to capture these.

XDMCP is primarily useful for running thin clients such as in terminal labs. Those thin clients will only ever need the network to access the server, and so it seems like the best security policy to have those thin clients on a separate network that cannot be accessed by the outside world, and can only connect to the server. The only point from which you need to access outside is the server. This type of set up should never use an unmanaged hub or other sniffable network.

3.6. XDMCP Access Control

XDMCP access control is done using TCP wrappers. It is possible to compile GDM without TCP wrapper support, so this feature may not be supported on some Operating Systems.

You should use the daemon name `gdm` in the `<etc>/hosts.allow` and `<etc>/hosts.deny` files. For example, to deny computers from `.evil.domain` from logging in, then add

```
gdm: .evil.domain
```

to `<etc>/hosts.deny`. You may also need to add

```
gdm: .your.domain
```

to your `<etc>/hosts.allow` if you normally disallow all services from all hosts. See the [hosts.allow\(5\)](#) man page for details.

3.7. Firewall Security

Even though GDM tries to outsmart potential attackers trying to take advantage of XDMCP, it is still advised that you block the XDMCP port (normally UDP port 177) on your firewall unless really needed. GDM guards against denial of service attacks, but the X protocol is still inherently insecure and should only be used in controlled environments. Also each remote connection takes up lots of resources, so it is much easier to do a denial of service attack via XDMCP than attacking a webserver.

It is also wise to block all of the Xserver ports. These are TCP ports 6000+ (one for each display number) on your firewall. Note that GDM will use display numbers 20 and higher for flexible on-demand servers.

X is not a very safe protocol when using it over the Internet, and XDMCP is even less safe.

3.8. PolicyKit

GDM may be configured to use PolicyKit to allow the system administrator to control whether the login screen should provide the shutdown and restart buttons on the greeter screen.

These buttons are controlled by the `org.freedesktop.consolekit.system.stop-multiple-users` and `org.freedesktop.consolekit.system.restart-multiple-users` actions respectively. Policy for these actions can be set up using the `polkit-gnome` authorization tool, or the `polkit-auth` command line program.

3.9. RBAC (Role Based Access Control)

GDM may be configured to use RBAC instead of PolicyKit. In this case the RBAC configuration is used to control whether the login screen should provide the shutdown and restart buttons on the greeter screen.

For example, on Solaris, the `"solaris.system.shutdown"` authorisation is used to control this. Simply modify the `/etc/user_attr` file so that the `"gdm"` user has this authorisation.

4. Support for ConsoleKit

GDM includes support for publishing user login information with the user and login session accounting framework known as ConsoleKit. ConsoleKit is able to keep track of all the users currently logged in. In this respect, it can be used as a replacement for the `utmp` or `utmpx` files that are available on most Unix-like Operating Systems.

When GDM is about to create a new login process for a user it will call a privileged method of ConsoleKit in order to open a new session for this user. At this time GDM also provides ConsoleKit with information about this user session such as: the user ID, the X11 Display name that will be associated with the session, the host-name from which the session originates (useful in the case of an XDMCP session), whether or not this session is attached, etc. As the entity that initiates the user process, GDM is in a unique position to know about the user session and to be trusted to provide these bits of information. The use of this privileged method is restricted by the use of the D-Bus system message bus security policy.

In case a user with an existing session has authenticated at GDM and requests to resume that existing session, GDM calls a privileged method of ConsoleKit to unlock that session. The exact

details of what happens when the session receives this unlock signal are undefined and session-specific. However, most sessions will unlock a screensaver in response.

When the user chooses to log out, or if GDM or the session quit unexpectedly the user session will be unregistered from ConsoleKit.

If support for ConsoleKit is not desired it can be disabled at build time using the "--with-console-kit=no" option when running configure.

5. Configuration

GDM has a number of configuration interfaces. These include scripting integration points, daemon configuration, greeter configuration, general session settings, integration with gnome-settings-daemon configuration, and session configuration. These types of integration are described in detail below.

- 5.1. [Scripting Integration Points](#)
- 5.2. [Autostart Configuration](#)
- 5.3. [Xsession Script](#)
- 5.4. [Daemon Configuration](#)
- 5.5. [Simple Greeter Configuration](#)
- 5.6. [Accessibility Configuration](#)
- 5.7. [General Session Settings](#)
- 5.8. [GNOME Settings Daemon](#)
- 5.9. [GDM Session Configuration](#)
- 5.10. [GDM User Session and Language Configuration](#)

5.1. Scripting Integration Points

The GDM script integration points can be found in the `<etc>/gdm/` directory:

```
Xsession
Init/
PostLogin/
PreSession/
PostSession/
```

The `Init`, `PostLogin`, `PreSession`, and `PostSession` scripts all work as described below.

For each type of script, the default one which will be executed is called "Default" and is stored in a directory associated with the script type. So the default `Init` script is `<etc>/gdm/Init/Default`. A per-display script can be provided, and if it exists it will be run instead of the default script. Such scripts are stored in the same directory as the default script and have the same name as the Xserver `DISPLAY` value for that display. For example, if the `<Init>/:0` script exists, it will be run for `DISPLAY ":0"`.

All of these scripts are run with root privilege and return 0 if run successfully, and a non-zero return code if there was any failure that should cause the login session to be aborted. Also note that GDM will block until the scripts finish, so if any of these scripts hang, this will cause the login process to also hang.

When the Xserver for the login GUI has been successfully started, but before the login GUI is actually displayed, GDM will run the `Init` script. This script is useful for starting programs that should be run while the login screen is showing, or for doing any special initialization if required.

After the user has been successfully authenticated GDM will run the `PostLogin` script. This is done before any session setup has been done, including before the `pam_open_session` call. This script is useful for doing any session initialisation that needs to happen before the session starts. For example, you might set up the user's `$HOME` directory if needed.

After the user session has been initialised, GDM will run the `PreSession` script. This script is useful for doing any session initialisation that needs to happen after the session has been initialised. It can be used for session management or accounting, for example.

When a user terminates their session, GDM will run the `PostSession` script. Note that the Xserver will have been stopped by the time this script is run, so it should not be accessed.

Note that the `PostSession` script will be run even when the display fails to respond due to an I/O error or similar. Thus, there is no guarantee that X applications will work during script execution.

All of the above scripts will set the `$RUNNING_UNDER_GDM` environment variable to `yes`. If the scripts are also shared with other display managers, this allows you to identify when GDM is calling these scripts, so you can run specific code when GDM is used.

5.2. Autostart Configuration

The `<share>/gdm/autostart/LoginWindow` directory contains files in the format specified by the "FreeDesktop.org Desktop Application Autostart Specification". Standard features in the specification may be used to specify programs that should auto-restart or only be launched if a GConf configuration value is set, etc.

Any `.desktop` files in this directory will cause the associated program to automatically start with the login GUI greeter. By default, GDM is shipped with files which will autostart the `gdm-simple-greeter` login GUI greeter itself, the `gnome-power-manager` application, the `gnome-settings-daemon`, and the `Metacity` window manager. These programs are needed for the greeter program to work. In addition, desktop files are provided for starting various AT programs if the configuration values specified in the Accessibility Configuration section below are set.

5.3. Xsession Script

There is also an `Xsession` script located at `<etc>/gdm/Xsession` which is called between the `PreSession` and the `PostSession` scripts. This script does not support per-display like the other scripts. This script is used for actually starting the user session. This script is run as the user, and it will run whatever session was specified by the Desktop session file the user selected to start.

5.4. Daemon Configuration

The GDM daemon is configured using the `<etc>/gdm/custom.conf` file. Default values are stored in GConf in the `gdm.schemas` file. It is recommended that end-users modify the `/etc/gdm/custom.conf` file because the `schemas` file may be overwritten when the user updates their system to have a newer version of GDM.

Note that older versions of GDM supported additional configuration options which are no longer supported in the latest versions of GDM.

The `<etc>/gdm/custom.conf` file is in the `keyfile` format. Keywords in brackets define group sections, strings before an equal sign (=) are keys and the data after equal sign represents their value. Empty lines or lines starting with the hash mark (#) are ignored.

The file `/etc/gdm/custom.conf` supports the "[daemon]", "[security]", and "[xdmcp]" group sections. Within each group, there are particular key/value pairs that can be specified to modify how GDM behaves. For example, to enable timed login and specify the timed login user to be a user named "you", you would modify the file so it contains the following lines:

```
[daemon]
TimedLoginEnable=true
TimedLogin=you
```

A full list of supported configuration keys follow:

- 5.4.1. [\[chooser\]](#)
- 5.4.2. [\[daemon\]](#)
- 5.4.3. [Security Options](#)
- 5.4.4. [XDCMP Support](#)

5.4.1. [chooser]

Multicast

```
Multicast=false
```

If true and IPv6 is enabled, the chooser will send a multicast query to the local network and collect responses from the hosts who have joined the multicast group.

MulticastAddr

```
MulticastAddr=ff02::1
```

This is the Link-local Multicast address.

5.4.2. [daemon]

Group

```
Group=gdm
```

The group name under which the greeter and other GUI programs are run. Refer to the `User` configuration key and to the "Security->GDM User And Group" section of this document for more information.

TimedLoginEnable

```
TimedLoginEnable=false
```

If the user given in `TimedLogin` should be logged in after a number of seconds (set with `TimedLoginDelay`) of inactivity on the login screen. This is useful for public access terminals or perhaps even home use. If the user uses the keyboard or browses the menus, the timeout will be reset to `TimedLoginDelay` or 30 seconds, whichever is higher. If the user does not enter a username but just hits the ENTER key while the login program is requesting the username, then GDM will assume the user wants to log in immediately as the timed user. Note that no password will be asked for this user so you should be careful, although if using PAM it can be configured to require password entry before allowing login. Refer to the "Security->PAM" section of the manual for more information, or for help if this feature does not seem to work.

TimedLogin

```
TimedLogin=
```

This is the user that should be logged in after a specified number of seconds of inactivity.

TimedLoginDelay

```
TimedLoginDelay=30
```

Delay in seconds before the `TimedLogin` user will be logged in.

AutomaticLoginEnable

```
AutomaticLoginEnable=false
```

If true, the user given in `AutomaticLogin` should be logged in immediately. This feature is like timed login with a delay of 0 seconds.

AutomaticLogin

```
AutomaticLogin=
```

This is the user that should be logged in immediately if `AutomaticLoginEnable` is true.

User

```
User=gdm
```

The username under which the greeter and other GUI programs are run. Refer to the `Group` configuration key and to the "Security->GDM User And Group" section of this document for more information.

5.4.3. Security Options

[security]

DisallowTCP

```
DisallowTCP=true
```

If true, then always append `-nolisten tcp` to the command line when starting attached Xservers, thus disallowing TCP connection. This is a more secure configuration if you are not using remote connections.

5.4.4. XDMCP Support

[xdmcp]

DisplaysPerHost

```
DisplaysPerHost=1
```

To prevent attackers from filling up the pending queue, GDM will only allow one connection for each remote computer. If you want to provide display services to computers with more than one screen, you should increase this value.

Note that the number of attached DISPLAYS allowed is not limited. Only remote connections via XDMCP are limited by this configuration option.

Enable

```
Enable=false
```

Setting this to true enables XDMCP support allowing remote displays/X terminals to be managed by GDM.

`gdm` listens for requests on UDP port 177. See the Port option for more information.

If GDM is compiled to support it, access from remote displays can be controlled using the TCP Wrappers library. The service name is `gdm`. You should add

```
gdm:.my.domain
```

to your `<etc>/hosts.allow`, depending on your TCP Wrappers configuration. See the [hosts.allow](#) man page for details.

Please note that XDMCP is not a particularly secure protocol and that it is a good idea to block UDP port 177 on your firewall unless you really need it.

HonorIndirect

```
HonorIndirect=true
```

Enables XDMCP INDIRECT choosing (i.e. remote execution of `gdmchooser`) for X-terminals which do not supply their own display browser.

MaxPending


```
MaxPending=4
```

To avoid denial of service attacks, GDM has fixed size queue of pending connections. Only `MaxPending` displays can start at the same time. Please note that this parameter does not limit the number of remote displays which can be managed. It only limits the number of displays initiating a connection simultaneously.

MaxSessions

```
MaxSessions=16
```

Determines the maximum number of remote display connections which will be managed simultaneously. I.e. the total number of remote displays that can use your host.

MaxWait

```
MaxWait=30
```

When GDM is ready to manage a display an `ACCEPT` packet is sent to it containing a unique session id which will be used in future XDMCP conversations.

GDM will then place the session id in the pending queue waiting for the display to respond with a `MANAGE` request.

If no response is received within `MaxWait` seconds, GDM will declare the display dead and erase it from the pending queue freeing up the slot for other displays.

MaxWaitIndirect

```
MaxWaitIndirect=30
```

The `MaxWaitIndirect` parameter determines the maximum number of seconds between the time where a user chooses a host and the subsequent indirect query where the user is connected to the host. When the timeout is exceeded, the information about the chosen host is forgotten and the indirect slot freed up for other displays. The information may be forgotten earlier if there are more hosts trying to send indirect queries than `MaxPendingIndirect`.

PingIntervalSeconds

```
PingIntervalSeconds=15
```

Interval in which to ping the Xserver in seconds. If the Xserver does not respond before the next time we ping it, the connection is stopped and the session ended. This is a combination of the XDM `PingInterval` and `PingTimeout`, but in seconds.

Note that GDM in the past used to have a `PingInterval` configuration key which was also in minutes. For most purposes you'd want this setting to be lower than one minute. However since in most cases where XDMCP would be used (such as terminal labs), a lag of more than 15 or so seconds would

really mean that the terminal was turned off or restarted and you would want to end the session.

Port

```
Port=177
```

The UDP port number `gdm` should listen to for XDMCP requests. Do not change this unless you know what you are doing.

Willing

```
Willing=<etc>/gdm/Xwilling
```

When the machine sends a WILLING packet back after a QUERY it sends a string that gives the current status of this server. The default message is the system ID, but it is possible to create a script that displays customised message. If this script does not exist or this key is empty the default message is sent. If this script succeeds and produces some output, the first line of its output is sent (and only the first line). It runs at most once every 3 seconds to prevent possible denial of service by flooding the machine with QUERY packets.

5.5. Simple Greeter Configuration

The GDM default greeter is called the simple Greeter and is configured via GConf. Default values are stored in GConf in the `gdm-simple-greeter.schemas` file. These defaults can be overridden if the "gdm" user has a writable `$HOME` directory to store GConf settings. These values can be edited using the `gconftool-2` or `gconf-editor` programs. The following configuration options are supported:

Greeter Configuration Keys

`/apps/gdm/simple-greeter/banner_message_enable`

```
false (boolean)
```

Controls whether the banner message text is displayed.

`/apps/gdm/simple-greeter/banner_message_text`

```
NULL (string)
```

Specifies the text banner message to show on the greeter window.

`/apps/gdm/simple-greeter/debug`

```
false (boolean)
```

If true, then debugging mode is enabled for the greeter.

`/apps/gdm/simple-greeter/disable_restart_buttons`

```
false (boolean)
```

Controls whether to show the restart buttons in the login window.

/apps/gdm/simple-greeter/disable_user_list

```
false (boolean)
```

If true, then the face browser with known users is not shown in the login window.

/apps/gdm/simple-greeter/logo_icon_name

```
computer (string)
```

Set to the themed icon name to use for the greeter logo.

/apps/gdm/simple-greeter/recent-languages

```
[] (string list)
```

Set to a list of languages to be shown by default in the login window. Default value is []. With the default setting only the system default language is shown and the option "Other..." which pops up a dialogue box showing a full list of available languages which the user can select.

Users are not intended to change this setting by hand. Instead GDM keeps track of any languages selected in this configuration key, and will show them in the language combo box along with the "Other..." choice. This way, commonly-selected languages are easier to select.

/apps/gdm/simple-greeter/recent-layouts

```
[] (string list)
```

Set to a list of keyboard layouts to be shown by default in the login panel. Default value is []. With the default setting only the system default keyboard layout is shown and the option "Other..." which pops up a dialogue box showing a full list of available keyboard layouts which the user can select.

Users are not intended to change this setting by hand. Instead GDM keeps track of any keyboard layouts selected in this configuration key, and will show them in the keyboard layout combo box along with the "Other..." choice. This way, commonly-selected keyboard layouts are easier to select.

/apps/gdm/simple-greeter/wm_use_compiz

```
false (boolean)
```

Controls whether Compiz is used as the window manager instead of Metacity.

5.6. Accessibility Configuration

This section describes the accessibility configuration options available in GDM.

- 5.6.1. [GDM Accessibility Dialog And Gconf Keys](#)
- 5.6.2. [Accessibility GConf Keys](#)
- 5.6.3. [Linking GConf Keys to Accessibility Tools](#)
- 5.6.4. [Example Of Modifying Accessibility Tool Configuration](#)

5.6.1. GDM Accessibility Dialog And Gconf Keys

The GDM greeter panel at the login screen displays an accessibility icon. Clicking on that icon opens the GDM Accessibility Dialog. In the GDM Accessibility Dialog, there is a list of checkboxes, so the user can enable or disable the associated assistive tools.

The checkboxes that correspond to the on-screen keyboard, screen magnifier and screen reader assistive tools act on the three GConf keys that are described in the next section of this document. By enabling or disabling these checkboxes, the associated GConf key is set to "true" or "false". When the key is set to true, the assistive tools linked to this GConf key are launched. When the key is set to "false", any running assistive tool linked to this GConf key are terminated. Also these keys could be set to "true" by default so that the assistive programs are always launched with the GDM login GUI.

The other checkboxes in the GDM Accessibility Dialog do not have corresponding GConf keys because no additional program is launched to provide the accessibility features that they offer. These other options correspond to accessibility features that are provided by the Xserver, which is always running during the GDM session.

5.6.2. Accessibility GConf Keys

GDM offers the following GConf keys to control its accessibility features:

GDM Configuration Keys

`/desktop/gnome/interface/accessibility`

```
false (boolean)
```

Controls whether the Accessibility infrastructure will be started with the GDM GUI. This is needed for many accessibility technology programs to work.

`/desktop/gnome/applications/at/screen_magnifier_enabled`

```
false (boolean)
```

If set, then the screen magnifier will be started with the GDM GUI programs.

`/desktop/gnome/applications/at/screen_keyboard_enabled`

```
false (boolean)
```

If set, then the GNOME on-screen keyboard will be started with the GDM GUI programs.

/desktop/gnome/applications/at/screen_reader_enabled

false (boolean)

If set, then the GNOME screen reader will be started with the GDM GUI programs.

5.6.3. Linking GConf Keys to Accessibility Tools

For magnification, on-screen keyboard, and screen reader; the assistive tool which gets launched depends on the desktop files located in the GDM autostart directory as described in the "Autostart Configuration" section of this manual. Any desktop file in the GDM autostart directory can be linked to a GDM Accessibility GConf key via specifying that GConf key in the AutostartCondition value in the desktop file. For example, the value could be one of the following:

```
AutostartCondition=GNOME /desktop/gnome/applications/at/screen_keyboard_enabled
AutostartCondition=GNOME /desktop/gnome/applications/at/screen_magnifier_enabled
AutostartCondition=GNOME /desktop/gnome/applications/at/screen_reader_enabled
```

When an accessibility key is true, then any program which is linked to that key in a GDM autostart desktop file will be launched (unless the Hidden key is set to true in that desktop file). A single GConf key can even start multiple assistive tools if there are multiple desktop files with this AutostartCondition in the GDM autostart directory.

5.6.4. Example Of Modifying Accessibility Tool Configuration

For example, if GNOME is distributed with GOK as the default on-screen keyboard, then this could be replaced with a different program if desired. To replace GOK with the on-screen keyboard application "onboard" and additionally activate the assistive tool "mousetweaks" for dwelling support, then the following configuration is needed.

Create a desktop file for onboard and a second one for mousetweaks; for example, onboard.desktop and mousetweaks.desktop. These files must be placed in the GDM autostart directory and be in the format as explained in the "Autostart Configuration" section of this document. The following is an example onboard.desktop file:

```
[Desktop Entry]
Encoding=UTF-8
Name=Onboard Onscreen Keyboard
Comment=Use an on-screen keyboard
TryExec=onboard
Exec=onboard --size 500x180 -x 20 -y 10
Terminal=false
Type=Application
StartupNotify=true
Categories=GNOME;GTK;Accessibility;
AutostartCondition=GNOME /desktop/gnome/applications/at/screen_keyboard_enabled
```

The following is an example mousetweaks.desktop file:

```
[Desktop Entry]
Encoding=UTF-8
Name=Software Mouse-Clicks
Comment=Perform clicks by dwelling with the pointer
TryExec=mousetweaks
Exec=mousetweaks --enable-dwell -m window -c -x 20 -y 240
Terminal=false
Type=Application
StartupNotify=true
Categories=GNOME;GTK;Accessibility;
AutostartCondition=GNOME /desktop/gnome/applications/at/screen_keyboard_enabled
```

Note the line with the `AutostartCondition` that links both desktop files to the GConf key for the on-screen keyboard.

To disable GOK from starting, the desktop file for the GOK on-screen keyboard must be removed or deactivated. Otherwise onboard and GOK would simultaneously be started. This can be done by removing the `gok.desktop` file from the GDM autostart directory, or by adding the `"Hidden=true"` key setting to the `gok.desktop` file.

After making these changes, GOK will no longer be started when the user activates the on-screen keyboard in the GDM session; but onboard and mousetweaks will instead be launched instead.

5.7. General Session Settings

The GDM Greeter uses some of the same framework that your desktop session will use. And so, it is influenced by a number of the same GConf settings. For each of these settings the Greeter will use the default value unless it is specifically overridden by a) GDM's installed mandatory policy b) system mandatory policy. GDM installs its own mandatory policy to lock down some settings for security.

5.8. GNOME Settings Daemon

GDM enables the following `gnome-settings-daemon` plugins: `a11y-keyboard`, `background`, `sound`, `xsettings`.

These are responsible for things like the background image, font and theme settings, sound events, etc.

Plugins can also be disabled using GConf. For example, if you want to disable the sound plugin then unset the following key: `/apps/gdm/simple-greeter/settings-manager-plugins/sound/active`.

5.9. GDM Session Configuration

GDM sessions are specified using the FreeDesktop.org Desktop Entry Specification, which can be referenced at the following URL: <http://www.freedesktop.org/wiki/Specifications/desktop-entry-spec>.

These files are installed, by default, to `<etc>/X11/sessions/`. For backwards compatibility any desktop files in the `<etc>/Sessions`, `<share>/xsessions`, and `<share/gdm/BuiltInSessions` directories are also recognised by GDM.

A session can be disabled by editing the desktop file and adding a line that says `Hidden=true`.

5.10. GDM User Session and Language Configuration

The user's default session and language choices are stored in the `~/.dmrc` file. When a user logs in for the first time, this file is created with the user's initial choices. The user can change these default values by simply changing to a different value when logging in. GDM will remember this change for subsequent logins.

The `~/.dmrc` file is in the standard INI format. It has one section called `[Desktop]` which has two keys: `Session` and `Language`.

The `Session` key specifies the basename of the session `.desktop` file that the user wishes to normally use without the `.desktop` extension. The `Language` key specifies the language that the user wishes to use by default. If either of these keys is missing, the system default is used. The file would normally look as follows:

```
[Desktop]
Session=gnome
Language=cs_CZ.UTF-8
```

6. GDM Commands

6.1. GDM Root User Commands

6.1. GDM Root User Commands

The GDM package provides the following commands in `sbindir` intended to be run by the root user:

- 6.1.1. `gdm` and `gdm-binary` Command Line Options
- 6.1.2. `gdm-restart` Command Line Options
- 6.1.3. `gdm-safe-restart` Command Line Options
- 6.1.4. `gdm-stop` Command Line Options

6.1.1. `gdm` and `gdm-binary` Command Line Options

The `gdm` command is really just a script which runs the `gdm-binary`, passing along any options. Before launching `gdm-binary`, the `gdm` wrapper script will source the `<etc>/profile` file to set the standard system environment variables. In order to better support internationalisation, it will also set the `LC_MESSAGES` environment variable to `LANG` if neither `LC_MESSAGES` or `LC_ALL` are set. The `gdm-binary` is the actual GDM daemon.

`gdm` and `gdm-binary` Command Line Options

`-?, --help`

Gives a brief overview of the command line options.

`--debug`

Print debug output to the syslog. This is typically `<var>/log/messages` or `<var>/adm/messages` depending on your Operating System.

`--fatal-warnings`

Make all warnings cause GDM to exit.

`--timed-exit`

Exit after 30 seconds. Useful for debugging.

`--version`

Print the version of the GDM daemon.

6.1.2. `gdm-restart` Command Line Options

`gdm-restart` stops and restarts GDM by sending the GDM daemon a HUP signal. This command will immediately terminate all sessions and log out users currently logged in with GDM.

6.1.3. `gdm-safe-restart` Command Line Options

`gdm-safe-restart` stops and restarts GDM by sending the GDM daemon a USR1 signal. GDM will be restarted as soon as all users log out.

6.1.4. `gdm-stop` Command Line Options

`gdm-stop` stops GDM by sending the GDM daemon a TERM signal.

7. Troubleshooting

This section discusses helpful tips for getting GDM working. In general, if you have a problem using GDM, you can submit a bug or send an e-mail to the `gdm-list` mailing list. Information about how to do this is in the Introduction section of the document.

If GDM is failing to work properly, it is always a good idea to include debug information. To turn on debug, launch `gdm` with the `--debug` option. Then use GDM to the point where it fails, and debug output will be sent to your system log (`<var>/log/messages` or `<var>/adm/messages` depending on your Operating System). If you share this output with the GDM community via a bug report or e-mail, please only include the GDM-related debug information and not the entire file since it can be large. If you do not see any GDM syslog output, you may need to configure syslog (refer to the [syslog](#) man page).

7.1. [GDM Will Not Start](#)

7.1. GDM Will Not Start

There are a many problems that can cause GDM to fail to start, but this section will discuss a few common problems and how to approach tracking down a problem with GDM starting. Some problems will cause GDM to respond with an error message or dialogue when it tries to start, but it can be difficult to track down problems when GDM fails silently.

First make sure that the Xserver is configured properly. The GDM configuration file contains a command in the [server-Standard] section that is used for starting the Xserver. Verify that this command works on your system. Running this command from the console should start the Xserver. If it fails, then the problem is likely with your Xserver configuration. Refer to your Xserver error log for an idea of what the problem may be. The problem may also be that your Xserver requires different command-line options. If so, then modify the Xserver command in the GDM configuration file so that it is correct for your system.

Also make sure that the `/tmp` directory has reasonable ownership and permissions, and that the machine's file system is not full. These problems will cause GDM to fail to start.

8. Licence

This program is free software; you can redistribute it and/or modify it under the terms of the [GNU General Public Licence](#) as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *GNU General Public Licence* for more details.

A copy of the *GNU General Public Licence* is included as an appendix to the *GNOME Users Guide*. You may also obtain a copy of the *GNU General Public Licence* from the Free Software Foundation by visiting [their Web site](#) or by writing to

Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor Boston, MA 02110-1301 USA