

# The kdm Handbook

Oswald Buddenhagen



# The kdm Handbook

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Quick Start Guide</b>	<b>2</b>
<b>3</b>	<b>The Login Window</b>	<b>4</b>
<b>4</b>	<b>Configuring kdm</b>	<b>6</b>
4.1	The Login Manager KDE Control Center Module . . . . .	7
4.1.1	Appearance . . . . .	7
4.1.2	Font . . . . .	8
4.1.3	Background . . . . .	8
4.1.4	Shutdown . . . . .	8
4.1.5	Users . . . . .	9
4.1.6	Convenience . . . . .	10
4.1.6.1	Automatic Login . . . . .	10
4.1.6.2	Password-Less Login . . . . .	10
<b>5</b>	<b>The Files kdm Uses for Configuration</b>	<b>11</b>
5.1	kdmrc - The kdm master configuration file . . . . .	11
5.1.1	The [General] section of kdmrc . . . . .	13
5.1.2	The [Xdmcp] section of kdmrc . . . . .	15
5.1.3	The [Shutdown] section of kdmrc . . . . .	16
5.1.4	The [X-*-Core] section class of kdmrc . . . . .	16
5.1.5	The [X-*-Greeter] section class of kdmrc . . . . .	22
5.2	Specifying permanent X-Servers . . . . .	27
5.3	XDMCP access control . . . . .	28
5.4	Supplementary programs . . . . .	29

## The kdm Handbook

5.4.1	Setup program . . . . .	29
5.4.2	Startup program . . . . .	30
5.4.3	Session program . . . . .	30
5.4.4	Reset program . . . . .	31
<b>6</b>	<b>Creating themes for the kdm greeter</b>	<b>32</b>
6.1	Theme Overview . . . . .	32
6.2	Detailed Description of Theme XML format . . . . .	33
6.2.1	Toplevel Node . . . . .	33
6.2.2	Item Nodes . . . . .	33
6.2.3	Layout Container Nodes . . . . .	35
6.2.3.1	Box Nodes . . . . .	36
6.2.3.2	Fixed Nodes . . . . .	36
6.2.4	Position Nodes . . . . .	36
6.2.5	Show Nodes . . . . .	37
6.2.6	Normal/Active/Prelight Nodes . . . . .	37
6.2.7	Widget Style Nodes . . . . .	39
6.2.8	Face Browser Color Nodes . . . . .	39
6.2.9	Text Nodes . . . . .	40
6.2.10	Stock Nodes . . . . .	40
6.2.11	Buddy Nodes . . . . .	41
<b>7</b>	<b>Configuring your system to use kdm</b>	<b>42</b>
7.1	Setting up kdm . . . . .	42
<b>8</b>	<b>Supporting multiple window managers</b>	<b>46</b>
<b>9</b>	<b>Using kdm for Remote Logins (XDMCP)</b>	<b>48</b>
<b>10</b>	<b>Advanced Topics</b>	<b>49</b>
10.1	Command Sockets . . . . .	49
<b>11</b>	<b>Other sources of information</b>	<b>53</b>
<b>12</b>	<b>Credits and License</b>	<b>54</b>
<b>13</b>	<b>Glossary</b>	<b>55</b>

### **Abstract**

This document describes kdm the KDE Display Manager. kdm is also known as the 'Login Manager'.

# Chapter 1

## Introduction

kdm provides a graphical interface that allows you to log in to a system. It prompts for login (username) and password, authenticates the user and starts a 'session'. kdm is superior to xdm, the X Display Manager, in a number of ways.

## Chapter 2

# Quick Start Guide

This is a quick start guide for users who fit the following pattern:

- X is configured and works with the command **startx** from the commandline.
- Each user will generally only use a single window manager or desktop environment, and does not change this choice very often, or is comfortable editing a single text file in order to change their choice.

This scenario will be sufficient for many environments where a single user or several users normally boot the computer and log into their preferred environment.

1. Create or open the file `/.xinitrc` If you already have a working `/.xinitrc`, go to the next step
2. If one does not already exist, add a line to the `/.xinitrc` to start your preferred window manager or desktop environment. For KDE you should enter:

```
startkde
```

For other window managers or desktop environments, you should look in their documentation for the correct command.

3. Make a link as follows:

```
ln -s ~/.xinitrc ~/.xsession
```

At this point, typing **startx** on the commandline should start X, with a KDE session. The next task is to try kdm.

As **root**, type **kdm** at the prompt.

## The kdm Handbook

You should see a login window, which is described more fully in chapter 3.

Typing your normal username and password in the fields provided, and leaving `default` selected as the session type should now open a KDE session for your user.

If you have other users to configure, you should repeat the procedure above for each of them.

### NOTE

This is a quick guide to getting up and running only. You probably will want to customize kdm further, for example, to hide the names of the system accounts, to allow further sessions, and much more. Please read through the rest of this manual to find out how to do these things.

## Chapter 3

# The Login Window

The user interface to kdm consists of dialog boxes. The main dialog box contains:

- Widgets allowing you to authenticate. When the "classic" authentication method is chosen, these are:
  - A Username: field for you to enter your username.
  - A Password: field for you to enter your password.
- (Optionally) a list containing entries with the name and graphical image of each user (for example, a digitized photograph). Clicking a list entry is equivalent to typing the associated username into the Username: field.
- (Optionally) a region to the right of or above the authentication area which can be used to display either a static image or an analog clock.
- A Login button that validates the username/password combination and attempts to start a session of the selected type.
- A Menu button that opens an action menu with the following items:
  - (Optionally) A Session Type item to choose the type of session (desktop environment, window manager) to start. See [chapter 8](#) to find out how to configure different session types.
  - (Optionally) A Authentication Method item to switch between different authentication methods like the classical username+password, smartcard, biometry, etc.. The actual authenticators are combinations of PAM modules and matching frontend modules ('conversation plugins'). See [\[?term\]](#).
  - (Optionally on local displays) A Switch User... item to switch between local sessions running on different virtual terminals of this computer.
  - (Optionally on local displays) A Restart X Server item that terminates the currently running X-Server, starts a new one and displays the login dialog again. You can use this if the display content seems to be broken somehow.

## The kdm Handbook

- (Optionally on remote displays) A Close Connection item that closes the connection to the XDMCP server you are currently connected to. If you got to this server through a host chooser, this will bring you back to the chooser, otherwise it will only reset the X-Server and bring up the login dialog again.
- (Optionally on local displays) A Remote Login item that displays a host chooser dialog with XDMCP servers one can log into remotely.
- (Optionally on local displays) A Console Mode item that terminates graphical login and leaves you at the console. See [?term] and [?term].
- (Optionally) A Shutdown... item that displays the Shutdown dialog box.

The Shutdown dialog box presents a set of buttons that allow one of these actions to be executed:

**Turn Off Computer** Shut the system down in a controlled manner, ready for power-down.

**Restart Computer** Shut the system down and reboot. For systems that use Lilo or Grub, an optional drop down box allows you to select a particular operating system to be used for the reboot.

**Schedule...** If this option is enabled, you may use it to enter a more complex shutdown dialog. See [?term] for details.

Pressing the Cancel button returns to the main kdm dialog box.

## Chapter 4

# Configuring kdm

This chapter assumes that kdm is already up and running on your system, and that you simply want to change its behavior in some way.

When kdm starts up, it reads its configuration from the folder `$KDEDIR/share/config/kdm/` (this may be `/etc/kde4/kdm/` or something else on your system).

The main configuration file is `kdmrc`; all other files are referenced from there and could be stored under any name anywhere on the system - but usually that would not make much sense for obvious reasons (one particular exception is referencing configuration files of an already installed xdm - however when a new kdm is installed, it will import settings from those files if it finds an already installed xdm).

Since kdm must run before any user is logged in, it is not associated with any particular user. Therefore, it is not possible to have user-specific configuration files; all users share the common `kdmrc`. It follows from this that the configuration of kdm can only be altered by those users that have write access to `$KDEDIR/share/config/kdm/kdmrc` (normally restricted to system administrators logged in as `root`).

You can view the `kdmrc` file currently in use on your system, and you can configure kdm by editing this file. Alternatively, you can use the graphical configuration tool provided by the KDE Control Center (under System Administration → Login Manager).

The remainder of this chapter describes configuration of kdm via the KDE Control Center module, and the [next chapter](#) describes the options available in `kd-mrc` itself. If you only need to configure for local users, the KDE Control Center module should be sufficient for your needs. If you need to configure remote logins, or have multiple kdm sessions running, you will need to read on.

## 4.1 The Login Manager KDE Control Center Module

Using this module, you can configure the KDE graphical login manager, kdm. You can change how the login screen looks, who has access using the login manager and who can shutdown the computer.

### NOTE

All settings will be written to the configuration file `kdmrc`, which in its original state has many comments to help you configure kdm. Using this KDE Control Center module will strip these comments from the file. All available options in `kdmrc` are covered in chapter 5.

The options listed in this chapter are cross referenced with their equivalents in `kdmrc`. All options available in the KControl module are also available directly in `kdmrc` but the reverse is not true.

In order to organize all of these options, this module is divided into several sections: [Appearance](#), [Font](#), [Background](#), [Shutdown](#), [Users](#) and [Convenience](#).

You can switch between the sections using the tabs at the top of the window.

### NOTE

If you are not currently logged in as a superuser, you will need to click the Administrator Mode... Button. You will then be asked for a superuser password. Entering a correct password will allow you to modify the settings of this module.

### 4.1.1 Appearance

From this page you can change the visual appearance of kdm, KDE's graphical login manager.

The Greeting: is the title of the login screen. Setting this is especially useful if you have many servers users may log in to. You may use various placeholders, which are described along with the corresponding key [GreetString](#) in `kdmrc`.

You can then choose to show either the current system time, a logo or nothing special in the login box. Make your choice in the radio buttons labeled Logo area:. This corresponds to [LogoArea](#) in `kdmrc`

If you chose Show logo you can now choose a logo:

- Drop an image file on the image button.
- Click on the image button and select a new image from the image chooser dialog.

## The kdm Handbook

If you do not specify a logo, the default `$KDEDIR/share/apps/kdm/pics/kdelogo.xpm` will be displayed.

Normally the login box is centered on the screen. Use the `Position:` options if you want it to appear elsewhere on the screen. You can specify the relative position (percentage of the screen size) for the center of the login window, relative to the top left of the display, in the fields labeled `X:` and `Y:` respectively. These correspond to the key `GreeterPos` in `kdmrc`.

While KDE's style depends on the settings of the user logged in, the style used by kdm can be configured using the `GUI Style:` and `Color Scheme:` options. These correspond to the keys `GUIStyle` and `ColorScheme` in `kdmrc` respectively.

Below that, you have a drop down box to choose the language for your login box, corresponding to setting `Language` in `kdmrc`.

### 4.1.2 Font

From this section of the module you can change the fonts used in the login window. Only fonts available to all users are available here, not fonts you have installed on a per user basis.

You can select three different font styles from the drop down box (`General:`, `Failures:`, `Greeting:`). When you click on the `Choose...` button a dialog appears from which you can select the new characteristics for the font style.

- The `General:` font is used in all other places in the login window.
- The `Failures:` font is used when a login fails.
- The `Greeting:` font is the font used for the title (`Greeting String`).

You can also check the box labeled `Use anti-aliasing for fonts` if you want smoothed fonts in the login dialog.

### 4.1.3 Background

Here you can change the desktop background which will be displayed before a user logs in. The options are identical to those described in the [background settings](#), save some that are not applicable to the kdm context.

### 4.1.4 Shutdown

Allow Shutdown

Use this drop down box to choose who is allowed to shut down:

- `Nobody`: No one can shutdown the computer using kdm. You must be logged in, and execute a command.

## The kdm Handbook

- **Everybody:** Everyone can shutdown the computer using kdm.
- **Only Root:** kdm requires that the `root` password be entered before shutting down the computer.

You can independently configure who is allowed to issue a shutdown command for the Local: and Remote: users.

### *Commands*

Use these text fields to define the exact shutdown command.

The **Halt:** command defaults to `/sbin/halt`. The **Restart:** command defaults to `/sbin/reboot`.

When **Show boot options** is enabled, kdm will on reboot offer you options for the lilo boot manager. For this feature to work, you will need to supply the correct paths to your **lilo** command and to lilo's map file. Note that this option is not available on all operating systems.

### 4.1.5 Users

From here you can change the way users are represented in the login window.

You may disable the user list in kdm entirely in the **Show Users** section. You can choose from:

**Show List** Only show users you have specifically enabled in the list alongside

If you do not check this box, no list will be shown. This is the most secure setting, since an attacker would then have to guess a valid login name as well as a password. It's also the preferred option if you have more than a handful of users to list, or the list itself would become unwieldy.

**Inverse selection** Allows you to instead select a list of users that should *not* be shown, and all other users will be listed.

Independently of the users you specify by name, you can use the System UIDs to specify a range of valid UIDs that are shown in the list. By default user id's under 1000, which are often system or daemon users, and user id's over 65000, are not shown.

You can also enable the **Sort users** checkbox, to have the user list sorted alphabetically. If this is disabled, users will appear in the order they are listed in the password file. kdm will also autocomplete user names if you enable the **Autocompletion** option.

If you choose to show users, then the login window will show images (which you select), of a list of users. When someone is ready to login, they may select their user name/image, enter their password, and they are granted access.

If you permit a user image, then you can configure the source for those images.

You can configure the admin picture here, for each user on the system. Depending on the order selected above, users may be able to override your selection.

If you choose not to show users, then the login window will be more traditional. Users will need to type their username and password to gain entrance. This is the preferred way if you have many users on this terminal.

## 4.1.6 Convenience

In the convenience tab you can configure some options that make life easier for lazy people, like automatic login or disabling passwords.

### IMPORTANT

Please think more than twice before using these options. Every option in the Convenience tab is well-suited to seriously compromise your system security. Practically, these options are only to be used in a completely non-critical environment, e.g. a private computer at home.

### 4.1.6.1 Automatic Login

Automatic login will give anyone access to a certain account on your system without doing any authentication. You can enable it using the option Enable Auto-login.

You can choose the account to be used for automatic login from the list labeled User:.

Automatic login can be suppressed by pressing the Shift key immediately after the X-Server switches to graphics mode and releasing it when kdm's hourglass cursor appears.

### 4.1.6.2 Password-Less Login

Using this feature, you can allow certain users to login without having to provide their password. Enable this feature using the Enable Password-less logins option.

Below this option you'll see a list of users on the system. Enable password-less login for specific users by checking the checkbox next to the login names. By default, this feature is disabled for all users.

### IMPORTANT

Again, this option should only be used in a safe environment. If you enable it on a rather public system you should take care that only users with heavy access restrictions are granted password-less login, e.g. `guest`.

You can also choose which user is 'preselected' when kdm starts. The default is None, but you can choose Previous to have kdm default to the last successfully logged in user, or you can Specify a particular user to always be selected from the list. You can also have kdm set the focus to the password field, so that when you reach the kdm login screen, you can type the password immediately.

The Automatically login after X server crash option allows you to skip the authentication procedure when your X server accidentally crashed.

## Chapter 5

# The Files kdm Uses for Configuration

This chapter documents the files that control kdm's behavior. Some of this can be also controlled from the KControl module, but not all.

### 5.1 `kdmrc` - The kdm master configuration file

The basic format of the file is 'INI-like'. Options are key/value pairs, placed in sections. Everything in the file is case sensitive. Syntactic errors and unrecognized key/section identifiers cause kdm to issue non-fatal error messages.

Lines beginning with # are comments; empty lines are ignored as well.

Sections are denoted by `[Name of Section]`.

You can configure every X-display individually.

Every display has a display name, which consists of a host name (which is empty for local displays specified in `StaticServers` or `ReserveServers`), a colon, and a display number. Additionally, a display belongs to a display class (which can be ignored in most cases).

Sections with display-specific settings have the formal syntax `[X-host [: number [_ class]] - sub-section]`

All sections with the same `sub-section` make up a section class.

You can use the wildcard \* (match any) for `host`, `number`, and `class`. You may omit trailing components; they are assumed to be \* then. The host part may be a domain specification like `.inf.tu-dresden.de` or the wildcard + (match non-empty).

From which section a setting is actually taken is determined by these rules:

## The kdm Handbook

- An exact match takes precedence over a partial match (for the host part), which in turn takes precedence over a wildcard (+ taking precedence over \*).
- Precedence decreases from left to right for equally exact matches.
- Example: display name 'myhost.foo:0', class 'dpy'
  - [X-myhost.foo:0\_dpy] precedes
  - [X-myhost.foo:0\_\*] (same as [X-myhost.foo:0]) precedes
  - [X-myhost.foo:\*\_dpy] precedes
  - [X-myhost.foo:\*\_\*] (same as [X-myhost.foo]) precedes
  - [X-.foo:\*\_\*] (same as [X-.foo]) precedes
  - [X-+:0\_dpy] precedes
  - [X-\*:0\_dpy] precedes
  - [X-\*:0\_\*] (same as [X-\*:0]) precedes
  - [X-:\*\_\*] (same as [X-\*]).
  - These sections do *not* match this display:  
[X-hishost], [X-myhost.foo:0\_dec], [X-\*:1], [X-:\*]

Common sections are [X-\*] (all displays), [X-:\*] (all local displays) and [X-:0] (the first local display).

The format for all keys is *key = value*. Keys are only valid in the section class they are defined for. Some keys do not apply to particular displays, in which case they are ignored.

If a setting is not found in any matching section, the default is used.

Special characters need to be backslash-escaped (leading and trailing spaces (\s), tab (\t), linefeed (\n), carriage return (\r) and the backslash itself (\\)).

In lists, fields are separated with commas without whitespace in between.

Some command strings are subject to simplified sh-style word splitting: single quotes (') and double quotes (") have the usual meaning; the backslash quotes everything (not only special characters). Note that the backslashes need to be doubled because of the two levels of quoting.

### NOTE

A pristine `kdmrc` is very thoroughly commented. All comments will be lost if you change this file with the `kcontrol` frontend.

### 5.1.1 The [General] section of `kdmrc`

This section contains global options that do not fit into any specific section.

**ConfigVersion** This option exists solely for the purpose of clean automatic upgrades. *Do not* change it, you may interfere with future upgrades and this could result in kdm failing to run.

**StaticServers** List of displays (X-Servers) permanently managed by kdm. Displays with a hostname are foreign displays which are expected to be already running, the others are local displays for which kdm starts an own X-Server; see `ServerCmd`. Each display may belong to a display class; append it to the display name separated by an underscore. See Section 5.2 for the details.  
The default is `'0'`.

**ReserveServers** List of on-demand displays. See `StaticServers` for syntax.  
Empty by default.

**ServerVTs** List of Virtual Terminals to allocate to X-Servers. For negative numbers the absolute value is used, and the VT will be allocated only if the kernel says it is free. If kdm exhausts this list, it will allocate free VTs greater than the absolute value of the last entry in this list. Currently Linux only.  
Empty by default.

**ConsoleTTYs** This option is for operating systems (OSs) with support for virtual terminals (VTs), by both kdm and the OSs itself. Currently this applies only to Linux.  
When kdm switches to console mode, it starts monitoring all TTY lines listed here (without the leading `/dev/`). If none of them is active for some time, kdm switches back to the X login.  
Empty by default.

**PidFile** The filename specified will be created to contain an ASCII representation of the process ID of the main kdm process; the PID will not be stored if the filename is empty.  
Empty by default.

**LockPidFile** This option controls whether kdm uses file locking to keep multiple display managers from running onto each other.  
The default is `'true'`.

**AuthDir** This names a directory under which kdm stores X-Server authorization files while initializing the session. kdm expects the system to clean up this directory from stale files on reboot.  
The authorization file to be used for a particular display can be specified with the `AuthFile` option in `[X-*Core]`.  
The default is `'/var/run/xauth'`.

## The kdm Handbook

**AutoRescan** This boolean controls whether kdm automatically re-reads its configuration files if it finds them to have changed.

The default is 'true'.

**ExportList** Additional environment variables kdm should pass on to all programs it runs. `LD_LIBRARY_PATH` and `XCURSOR_THEME` are good candidates; otherwise, it should not be necessary very often.

Empty by default.

**RandomFile** If the system has no native entropy source like `/dev/urandom` (see `RandomDevice`) and no entropy daemon like EGD (see `PrngdSocket` and `PrngdPort`) is running, kdm will fall back to its own pseudo-random number generator that will, among other things, successively checksum parts of this file (which, obviously, should change frequently).

This option does not exist on Linux and various BSDs.

The default is `'/dev/mem'`.

**PrngdSocket** If the system has no native entropy source like `/dev/urandom` (see `RandomDevice`), read random data from a Pseudo-Random Number Generator Daemon, like EGD (<http://egd.sourceforge.net>) via this UNIX domain socket.

This option does not exist on Linux and various BSDs.

Empty by default.

**PrngdPort** Same as `PrngdSocket`, only use a TCP socket on localhost.

**RandomDevice** The path to a character device which kdm should read random data from. Empty means to use the system's preferred entropy device if there is one.

This option does not exist on OpenBSD, as it uses the `arc4_random` function instead.

Empty by default.

**FifoDir** The directory in which the command sockets should be created; make it empty to disable them.

The default is `'/var/run/xdmctl'`.

**FifoGroup** The group to which the global command socket should belong; can be either a name or a numerical ID.

**DataDir** The directory in which kdm should store persistent working data; such data is, for example, the previous user that logged in on a particular display.

The default is `'/var/lib/kdm'`.

**DmrcDir** The directory in which kdm should store users' `.dmrc` files. This is only needed if the home directories are not readable before actually logging in (like with AFS).

Empty by default.

### 5.1.2 The [Xdmcp] section of `kdmrc`

This section contains options that control kdm's handling of XDMCP requests.

**Enable** Whether kdm should listen to incoming XDMCP requests.

The default is 'true'.

**Port** This indicates the UDP port number which kdm uses to listen for incoming XDMCP requests. Unless you need to debug the system, leave this with its default value.

The default is '177'.

**KeyFile** XDM-AUTHENTICATION-1 style XDMCP authentication requires a private key to be shared between kdm and the terminal. This option specifies the file containing those values. Each entry in the file consists of a display name and the shared key.

Empty by default.

**Xaccess** To prevent unauthorized XDMCP service and to allow forwarding of XDMCP IndirectQuery requests, this file contains a database of host-names which are either allowed direct access to this machine, or have a list of hosts to which queries should be forwarded to. The format of this file is described in Section 5.3.

The default is ``${kde_confdir}/kdm/Xaccess``.

**ChoiceTimeout** Number of seconds to wait for the display to respond after the user has selected a host from the chooser. If the display sends an XDMCP IndirectQuery within this time, the request is forwarded to the chosen host; otherwise, it is assumed to be from a new session and the chooser is offered again.

The default is '15'.

**RemoveDomainname** When computing the display name for XDMCP clients, the name resolver will typically create a fully qualified host name for the terminal. As this is sometimes confusing, kdm will remove the domain name portion of the host name if it is the same as the domain name of the local host when this option is enabled.

The default is 'true'.

**SourceAddress** Use the numeric IP address of the incoming connection on multihomed hosts instead of the host name. This is to avoid trying to connect on the wrong interface which might be down at this time.

The default is 'false'.

**Willing** This specifies a program which is run (as `root`) when an XDMCP DirectQuery or BroadcastQuery is received and this host is configured to offer XDMCP display management. The output of this program may be displayed in a chooser window. If no program is specified, the string 'Willing to manage' is sent.

Empty by default.

### 5.1.3 The [Shutdown] section of `kdmrc`

This section contains global options concerning system shutdown.

**HaltCmd** The command (subject to word splitting) to run to halt/poweroff the system.

The default is something reasonable for the system on which kdm was built, like `/sbin/shutdown -h now`.

**RebootCmd** The command (subject to word splitting) to run to reboot the system.

The default is something reasonable for the system kdm on which was built, like `/sbin/shutdown -r now`.

**AllowFifo** Whether it is allowed to shut down the system via the global command socket.

The default is 'false'.

**AllowFifoNow** Whether it is allowed to abort active sessions when shutting down the system via the global command socket.

This will have no effect unless `AllowFifo` is enabled.

The default is 'true'.

**BootManager** The boot manager kdm should use for offering boot options in the shutdown dialog.

**None** no boot manager

**Grub** Grub boot manager

**Lilo** Lilo boot manager (Linux on i386 & x86-64 only)

The default is 'None'.

### 5.1.4 The [X\*-Core] section class of `kdmrc`

This section class contains options concerning the configuration of the kdm backend (core).

**OpenDelay** See `OpenRepeat`.

The default is '15'.

**OpenTimeout** See `OpenRepeat`.

The default is '120'.

**OpenRepeat** These options control the behavior of kdm when attempting to open a connection to an X-Server. `OpenDelay` is the length of the pause (in seconds) between successive attempts, `OpenRepeat` is the number of attempts to make and `OpenTimeout` is the amount of time to spend on a connection attempt. After `OpenRepeat` attempts have been made, or if `OpenTimeout` seconds elapse in any particular connection attempt, the start attempt is considered failed.

The default is '5'.

## The kdm Handbook

**StartAttempts** How many times kdm should attempt to start a foreign display listed in `StaticServers` before giving up and disabling it. Local displays are attempted only once, and XDMCP displays are retried indefinitely by the client (unless the option `-once` was given to the X-Server).

The default is '4'.

**ServerAttempts** How many times kdm should attempt to start up a local X-Server. Starting up includes executing it and waiting for it to come up.

The default is '1'.

**ServerTimeout** How many seconds kdm should wait for a local X-Server to come up.

The default is '15'.

**ServerCmd** The command line to start the X-Server, without display number and VT spec. This string is subject to word splitting.

The default is something reasonable for the system on which kdm was built, like `/usr/X11R6/bin/X`.

**ServerArgsLocal** Additional arguments for the X-Servers for local sessions. This string is subject to word splitting.

Empty by default.

**ServerArgsRemote** Additional arguments for the X-Servers for remote sessions. This string is subject to word splitting.

Empty by default.

**ServerVT** The VT the X-Server should run on. `ServerVTs` should be used instead of this option. Leave it zero to let kdm assign a VT automatically. Set it to -1 to avoid assigning a VT altogether - this is required for setups with multiple physical consoles. Currently Linux only.

**ServerTTY** This option is for OSs without support for VTs, either by kdm or the OS itself. Currently this applies to all OSs but Linux.

When kdm switches to console mode, it starts monitoring this TTY line (specified without the leading `/dev/`) for activity. If the line is not used for some time, kdm switches back to the X login.

Empty by default.

**PingInterval** See `PingTimeout`.

The default is '5'.

**PingTimeout** To discover when *remote* displays disappear, kdm regularly pings them. `PingInterval` specifies the time (in minutes) between the pings and `PingTimeout` specifies the maximum amount of time (in minutes) to wait for the terminal to respond to the request. If the terminal does not respond, the session is declared dead and terminated.

If you frequently use X terminals which can become isolated from the managing host, you may wish to increase the timeout. The only worry is

## The kdm Handbook

that sessions will continue to exist after the terminal has been accidentally disabled.

The default is '5'.

**TerminateServer** Whether kdm should restart the local X-Server after session exit instead of resetting it. Use this if the X-Server leaks memory or crashes the system on reset attempts.

The default is 'false'.

**ResetSignal** The signal number to use to reset the local X-Server.

The default is '1 (SIGHUP)'.

**TermSignal** The signal number to use to terminate the local X-Server.

The default is '15 (SIGTERM)'.

**Authorize** Controls whether kdm generates and uses authorization for *local* X-Server connections. For XDMCP displays the authorization requested by the display is used; foreign non-XDMCP displays do not support authorization at all.

The default is 'true'.

**AuthNames** If *Authorize* is true, use the authorization mechanisms listed herein. The MIT-MAGIC-COOKIE-1 authorization is always available; XDM-AUTHORIZATION-1, SUN-DES-1 and MIT-KERBEROS-5 might be available as well, depending on the build configuration.

The default is 'DEF\_AUTH\_NAME'.

**ResetForAuth** Some *old* X-Servers re-read the authorization file at X-Server reset time, instead of when checking the initial connection. As kdm generates the authorization information just before connecting to the display, an old X-Server would not get up-to-date authorization information. This option causes kdm to send SIGHUP to the X-Server after setting up the file, causing an additional X-Server reset to occur, during which time the new authorization information will be read.

The default is 'false'.

**AuthFile** This file is used to communicate the authorization data from kdm to the X-Server, using the `-auth` X-Server command line option. It should be kept in a directory which is not world-writable as it could easily be removed, disabling the authorization mechanism in the X-Server. If not specified, a random name is generated from `AuthDir` and the name of the display.

Empty by default.

**Resources** This option specifies the name of the file to be loaded by `xrdb` as the resource database onto the root window of screen 0 of the display. KDE programs generally do not use X-resources, so this option is only needed if the `Setup` program needs some X-resources.

Empty by default.

## The kdm Handbook

**Xrdb** The `xrdb` program to use to read the X-resources file specified in `Recoures`. The command is subject to word splitting.

The default is `'${x_bindir}/xrdb'`.

**Setup** This string is subject to word splitting. It specifies a program which is run (as `root`) before offering the greeter window. This may be used to change the appearance of the screen around the greeter window or to put up other windows (e.g., you may want to run `xconsole` here). The conventional name for a file used here is `Xsetup`. See Section 5.4.1.

Empty by default.

**Startup** This string is subject to word splitting. It specifies a program which is run (as `root`) after the user authentication process succeeds. The conventional name for a file used here is `Xstartup`. See Section 5.4.2.

Empty by default.

**Reset** This string is subject to word splitting. It specifies a program which is run (as `root`) after the session terminates. The conventional name for a file used here is `Xreset`. See Section 5.4.4.

Empty by default.

**Session** This string is subject to word splitting. It specifies the session program to be executed (as the user owning the session). The conventional name for a file used here is `Xsession`. See Section 5.4.3.

The default is `'${x_bindir}xterm -ls -T'`.

**FailsafeClient** If the `Session` program fails to execute, `kdm` will fall back to this program. This program is executed with no arguments, but executes using the same environment variables as the session would have had (see Section 5.4.3).

The default is `'${x_bindir}xterm'`.

**UserPath** The `PATH` environment variable for non-root Sessions.

The default depends on the system `kdm` was built on.

**SystemPath** The `PATH` environment variable for all programs but non-root Sessions. Note that it is good practice not to include `.` (the current directory) into this entry.

The default depends on the system `kdm` was built on.

**SystemShell** The `SHELL` environment variable for all programs but the Session.

The default is `'/bin/sh'`.

**UserAuthDir** When `kdm` is unable to write to the usual user authorization file (`$HOME/.Xauthority`), it creates a unique file name in this directory and points the environment variable `XAUTHORITY` at the created file.

The default is `'/tmp'`.

**ForceUserAuthDir** If `true`, `UserAuthDir` will be used unconditionally.

The default is `'false'`.

## The kdm Handbook

**AutoReLogin** If enabled, kdm will automatically restart a session after an X-Server crash (or if it is killed by Alt-Ctrl-BackSpace). Note that enabling this feature opens a security hole: a secured display lock can be circumvented (unless KDE's built-in screen locker is used).

The default is 'false'.

**AllowRootLogin** If disabled, do not allow `root` (and any other user with `UID = 0`) to log in directly.

The default is 'true'.

**AllowNullPasswd** If disabled, only users that have passwords assigned can log in.

The default is 'true'.

**AllowShutdown** Who is allowed to shut down the system. This applies both to the greeter and to the command sockets.

**None** no Shutdown... menu entry is shown at all

**Root** the `root` password must be entered to shut down

**All** everybody can shut down the machine

The default is 'All'.

**AllowSdForceNow** Who is allowed to abort active sessions when shutting down.

**None** no forced shutdown is allowed at all

**Root** the `root` password must be entered to shut down forcibly

**All** everybody can shut down the machine forcibly

The default is 'All'.

**DefaultSdMode** The default choice for the shutdown condition/timing.

**Schedule** shut down after all active sessions exit (possibly at once)

**TryNow** shut down, if no active sessions are open; otherwise, do nothing

**ForceNow** shut down unconditionally

The default is 'Schedule'.

**ScheduledSd** How to offer shutdown scheduling options:

**Never** not at all

**Optional** as a button in the simple shutdown dialogs

**Always** instead of the simple shutdown dialogs

The default is 'Never'.

**NoPassEnable** Enable password-less logins on this display. *Use with extreme care!*

The default is 'false'.

## The kdm Handbook

**NoPassUsers** The users that do not need to provide a password to log in. Items which are prefixed with @ represent all users in the user group named by that item. \* means all users but root (and any other user with UID = 0). *Never* list root.

Empty by default.

**AutoLoginEnable** Enable automatic login. *Use with extreme care!*

The default is 'false'.

**AutoLoginAgain** If true, auto-login after logout. If false, auto-login is performed only when a display session starts up.

The default is 'false'.

**AutoLoginDelay** The delay in seconds before automatic login kicks in. This is also known as 'Timed Login'.

**AutoLoginUser** The user to log in automatically. *Never* specify root!

Empty by default.

**AutoLoginPass** The password for the user to log in automatically. This is *not* required unless the user is logged into a NIS or Kerberos domain. If you use this option, you should **chmod 600 kdmrc** for obvious reasons.

Empty by default.

**AutoLoginLocked** Immediately lock the automatically started session. This works only with KDE sessions.

The default is 'false'.

**SessionsDirs** A list of directories containing session type definitions.

The default is '{kde\_datadir}/kdm/sessions'.

**ClientLogFile** The file (relative to the user's home directory) to redirect the session output to.

The following character pairs are replaced by their value:

%d name of the current display

%u login name of the current user

%r empty at first. See below.

%% a single %

When the constructed filename cannot be used safely and the specification contains %stuffr, other names will be tried - this time expanding %stuffr to stuff followed by a random number.

The default is '.xsession-errors'.

**ClientLogFallback** Fallback when ClientLogFile cannot be used. The same expansions are supported. *Do not* use relative paths here.

The default is '/tmp/xerr-%u-%d%-r'.

**UseSessReg** Specify whether kdm's built-in utmp/wtmp/lastlog registration should be used. If it is not, the tool `sessreg` should be used in the `Startup` and `Reset` scripts, or, alternatively, the `pam_lastlog` module should be used on PAM-enabled systems.

The default is 'true'.

### 5.1.5 The [X\*-Greeter] section class of `kdmrc`

This section class contains options concerning the configuration of the kdm frontend (`greeter`).

**GUIStyle** Specify the widget style for the greeter. Empty means to use the built-in default which currently is `Plastik`.

Empty by default.

**ColorScheme** Specify the widget color scheme for the greeter. Empty means to use the built-in default which currently is yellowish grey with some light blue and yellow elements.

Empty by default.

**LogoArea** What should be shown in the greeter righthand of the input lines (if `UserList` is disabled) or above them (if `UserList` is enabled):

**None** nothing

**Logo** the image specified by `LogoPixmap`

**Clock** a neat analog clock

The default is 'Clock'.

**LogoPixmap** The image to show in the greeter if `LogoArea` is `Logo`.

Empty by default.

**GreeterPos** The relative coordinates (percentages of the screen size; X,Y) at which the center of the greeter is put. kdm aligns the greeter to the edges of the screen it would cross otherwise.

The default is '50,50'.

**GreeterScreen** The screen the greeter should be displayed on in multi-headed and Xinerama setups. The numbering starts with 0. For Xinerama, it corresponds to the listing order in the active `ServerLayout` section of `XF86Config`; -1 means to use the upper-left screen, -2 means to use the upper-right screen.

**GreetString** The headline in the greeter. An empty greeting means none at all.

The following character pairs are replaced by their value:

`%d` name of the current display

`%h` local host name, possibly with the domain name

## The kdm Handbook

**%n** local node name, most probably the host name without the domain name

**%s** operating system

**%r** operating system version

**%m** machine (hardware) type

**%%** a single %

The default is 'Welcome to %s at %n'.

**AntiAliasing** Whether the fonts used in the greeter should be antialiased.

The default is 'false'.

**GreetFont** The font for the greeter headline.

The default is 'Serif,20,bold'.

**StdFont** The normal font used in the greeter.

The default is 'Sans Serif,10'.

**FailFont** The font used for the 'Login Failed' message.

The default is 'Sans Serif,10,bold'.

**NumLock** What to do with the Num Lock modifier for the time the greeter is running:

**Off** turn off

**On** turn on

**Keep** do not change the state

The default is 'Keep'.

**Language** Language and locale to use in the greeter, encoded like \$LC\_LANG.

The default is 'en\_US'.

**UserCompletion** Enable autocompletion in the username line edit.

The default is 'false'.

**UserList** Show a user list with unix login names, real names, and images in the greeter.

The default is 'true'.

**ShowUsers** This option controls which users will be shown in the user view (**UserList**) and/or offered for autocompletion (**UserCompletion**). If it is **Selected**, **SelectedUsers** contains the final list of users. If it is **NotHidden**, the initial user list are all users found on the system. Users contained in **HiddenUsers** are removed from the list, just like all users with a **UID** greater than specified in **MaxShowUID** and users with a non-zero **UID** less than specified in **MinShowUID**. Items in **SelectedUsers** and **HiddenUsers** which are prefixed with @ represent all users in the user group named by that item. Finally, the user list will be sorted alphabetically, if **SortUsers** is enabled.

The default is 'NotHidden'.

## The kdm Handbook

**SelectedUsers** See ShowUsers.

Empty by default.

**HiddenUsers** See ShowUsers.

Empty by default.

**MinShowUID** See ShowUsers.

**MaxShowUID** See ShowUsers.

The default is '65535'.

**SortUsers** See ShowUsers.

The default is 'true'.

**FaceSource** If `UserList` is enabled, this specifies where kdm gets the images from:

**AdminOnly** from `<FaceDir>/$USER.face[.icon]`

**PreferAdmin** prefer `<FaceDir>`, fallback on `$HOME`

**PreferUser** ... and the other way round

**UserOnly** from the user's `$HOME/.face[.icon]`

The images can be in any format Qt recognizes, but the filename must match kdm's expectations: `.face.icon` should be a 48x48 icon, while `.-face` should be a 300x300 image. Currently the big image is used only as a fallback and is scaled down, but in the future it might be displayed full-size in the logo area or a tooltip.

The default is 'AdminOnly'.

**FaceDir** See FaceSource.

The default is `'${kde_datadir}/kdm/faces'`.

**PreselectUser** Specify, if/which user should be preselected for log in:

**None** do not preselect any user

**Previous** the user which successfully logged in last time

**Default** the user specified in the `DefaultUser` option

If `FocusPasswd` is enabled and a user was preselected, the cursor is placed in the password input field automatically.

### NOTE

Enabling user preselection can be considered a security hole, as it presents a valid login name to a potential attacker, so he 'only' needs to guess the password. On the other hand, one could set `DefaultUser` to a fake login name.

The default is 'None'.

**DefaultUser** See PreselectUser.

Empty by default.

## The kdm Handbook

**FocusPasswd** See `PreselectUser`.

The default is 'false'.

**EchoPasswd** If this is true, the entered password is echoed as bullets. Otherwise, no feedback is given at all.

The default is 'true'.

**UseBackground** If enabled, kdm will automatically start the **krootimage** program to set up the background; otherwise, the `Setup` program is responsible for the background.

The default is 'true'.

**BackgroundCfg** The configuration file to be used by **krootimage**. It contains a section named `[Desktop0]` like `kdesktoprc` does. Its options are not described herein; guess their meanings or use the control center.

The default is `'${kde_confdir}/kdm/backgroundrc'`.

**GrabServer** To improve security, the greeter grabs the X-Server and then the keyboard when it starts up. This option specifies if the X-Server grab should be held for the duration of the name/password reading. When disabled, the X-Server is ungrabbed after the keyboard grab succeeds; otherwise, the X-Server is grabbed until just before the session begins.

### NOTE

Enabling this option disables `UseBackground` and `Setup`.

The default is 'false'.

**GrabTimeout** This option specifies the maximum time kdm will wait for the grabs to succeed. A grab may fail if some other X-client has the X-Server or the keyboard grabbed, or possibly if the network latencies are very high. You should be cautious when raising the timeout, as a user can be spoofed by a look-alike window on the display. If a grab fails, kdm kills and restarts the X-Server (if possible) and the session.

The default is '3'.

**AuthComplain** Warn, if a display has no X-authorization. This will be the case if

- the authorization file for a local X-Server could not be created,
- a remote display from XDMCP did not request any authorization or
- the display is a 'foreign' display specified in `StaticServers`.

The default is 'true'.

**LoginMode** Specify whether the greeter of local displays should start up in host chooser (remote) or login (local) mode and whether it is allowed to switch to the other mode.

**LocalOnly** only local login possible

## The kdm Handbook

**DefaultLocal** start up in local mode, but allow switching to remote mode

**DefaultRemote** ... and the other way round

**RemoteOnly** only choice of remote host possible

The default is 'LocalOnly'.

**ChooserHosts** A list of hosts to be automatically added to the remote login menu. The special name \* means broadcast. Has no effect if LoginMode is LocalOnly.

The default is '\*'.

**ForgingSeed** Use this number as a random seed when forging saved session types, etc. of unknown users. This is used to avoid telling an attacker about existing users by reverse conclusion. This value should be random but constant across the login domain.

**ShowLog** Enable kdm's built-in **xconsole**. Note that this can be enabled for only one display at a time. This option is available only if kdm was **configured** with `--enable-kdm-xconsole`.

The default is 'false'.

**LogSource** The data source for kdm's built-in **xconsole**. If empty, a console log redirection is requested from `/dev/console`. Has no effect if ShowLog is disabled.

Empty by default.

**PluginsLogin** Specify conversation plugins for the login dialog; the first in the list is selected initially. Each plugin can be specified as a base name (which expands to `$kde_modulesdir/kgreet_base`) or as a full pathname.

Conversation plugins are modules for the greeter which obtain authentication data from the user. Currently only the `classic` plugin is shipped with KDE; it presents the well-known username and password form.

The default is 'classic'.

**PluginsShutdown** Same as PluginsLogin, but for the shutdown dialog.

The default is 'classic'.

**PluginOptions** A list of options of the form `Key=Value`. The conversation plugins can query these settings; it is up to them what possible keys are.

Empty by default.

**AllowConsole** Show the Console Login action in the greeter (if `ServerTTY/ConsoleTTYs` is configured).

The default is 'true'.

**AllowClose** Show the Restart X Server/Close Connection action in the greeter.

The default is 'true'.

**Preloader** A program to run while the greeter is visible. It is supposed to preload as much as possible of the session that is going to be started (most probably).

Empty by default.

**UseTheme** Whether the greeter should be themed.

The default is 'false'.

**Theme** The theme to use for the greeter. Can point to either a directory or an XML file.

Empty by default.

## 5.2 Specifying permanent X-Servers

Each entry in the `StaticServers` list indicates a display which should constantly be managed and which is not using XDMCP. This method is typically used only for local X-Servers that are started by kdm, but kdm can manage externally started ('foreign') X-Servers as well, may they run on the local machine or rather remotely.

The formal syntax of a specification is

```
display name [_display class]
```

for all X-Servers. 'Foreign' displays differ in having a host name in the display name, may it be `localhost`.

The *display name* must be something that can be passed in the `-display` option to an X program. This string is used to generate the display-specific section names, so be careful to match the names. The display name of XDMCP displays is derived from the display's address by reverse host name resolution. For configuration purposes, the `localhost` prefix from locally running XDMCP displays is *not* stripped to make them distinguishable from local X-Servers started by kdm.

The *display class* portion is also used in the display-specific sections. This is useful if you have a large collection of similar displays (such as a corral of X terminals) and would like to set options for groups of them. When using XDMCP, the display is required to specify the display class, so the manual for your particular X terminal should document the display class string for your device. If it does not, you can run kdm in debug mode and `grep` the log for 'class'.

The displays specified in `ReserveServers` will not be started when kdm starts up, but when it is explicitly requested via the command socket. If reserve displays are specified, the KDE menu will have a Start New Session item near the bottom; use that to activate a reserve display with a new login session. The monitor will switch to the new display, and you will have a minute to login. If there are no more reserve displays available, the menu item will be disabled.

When kdm starts a session, it sets up authorization data for the X-Server. For local servers, kdm passes `-auth filename` on the X-Server's command line to point it at its authorization data. For XDMCP displays, kdm passes the authorization data to the X-Server via the 'Accept' XDMCP message.

### 5.3 XDMCP access control

The file specified by the `AccessFile` option provides information which kdm uses to control access from displays requesting service via XDMCP. The file contains four types of entries: entries which control the response to 'Direct' and 'Broadcast' queries, entries which control the response to 'Indirect' queries, macro definitions, and entries which control on which network interfaces kdm listens for XDMCP queries. Blank lines are ignored, # is treated as a comment delimiter causing the rest of that line to be ignored, and \ causes an immediately following newline to be ignored, allowing host lists to span multiple lines.

The format of the 'Direct' entries is simple, either a host name or a pattern, which is compared against the host name of the display device. Alternatively, a macro may be used to make the entry match everything the macro expands to. Patterns are distinguished from host names by the inclusion of one or more meta characters; \* matches any sequence of 0 or more characters, and ? matches any single character. If the entry is a host name, all comparisons are done using network addresses, so any name which converts to the correct network address may be used. Note that only the first network address returned for a host name is used. For patterns, only canonical host names are used in the comparison, so ensure that you do not attempt to match aliases. Host names from XDMCP queries always contain the local domain name even if the reverse lookup returns a short name, so you can use patterns for the local domain. Preceding the entry with a ! character causes hosts which match that entry to be excluded. Preceding it with an = has no effect but is required when specifying a macro to distinguish the entry from a macro definition. To only respond to 'Direct' queries for a host or pattern, it can be followed by the optional `NOBROADCAST` keyword. This can be used to prevent a kdm server from appearing on menus based on 'Broadcast' queries.

An 'Indirect' entry also contains a host name, pattern or macro, but follows it with a list of host names or macros to which the queries should be forwarded. 'Indirect' entries can be excluding as well, in which case a (valid) dummy host name must be supplied to make the entry distinguishable from a 'Direct' entry. If compiled with IPv6 support, multicast address groups may also be included in the list of addresses the queries are forwarded to. If the indirect host list contains the keyword `CHOOSEER`, 'Indirect' queries are not forwarded, but instead a host chooser dialog is displayed by kdm. The chooser will send a 'Direct' query to each of the remaining host names in the list and offer a menu of all the hosts that respond. The host list may contain the keyword `BROADCAST`, to make the chooser send a 'Broadcast' query as well; note that on some operating systems, UDP packets cannot be broadcast, so this feature will not work.

When checking access for a particular display host, each entry is scanned in turn and the first matching entry determines the response. 'Direct' and 'Broadcast' entries are ignored when scanning for an 'Indirect' entry and vice-versa.

A macro definition contains a macro name and a list of host names and other macros that the macro expands to. To distinguish macros from hostnames, macro names start with a % character.

The last entry type is the `LISTEN` directive. The formal syntax is

```
LISTEN [interface [multicast list]]
```

If one or more `LISTEN` lines are specified, kdm listens for XDMCP requests only on the specified interfaces. *interface* may be a hostname or IP address representing a network interface on this machine, or the wildcard `*` to represent all available network interfaces. If multicast group addresses are listed on a `LISTEN` line, kdm joins the multicast groups on the given interface. For IPv6 multicasts, the IANA has assigned `ff0x:0:0:0:0:0:0:12b` as the permanently assigned range of multicast addresses for XDMCP. The *x* in the prefix may be replaced by any valid scope identifier, such as 1 for Node-Local, 2 for Link-Local, 5 for Site-Local, and so on (see IETF RFC 2373 or its replacement for further details and scope definitions). kdm defaults to listening on the Link-Local scope address `ff02:0:0:0:0:0:0:12b` to most closely match the IPv4 subnet broadcast behavior. If no `LISTEN` lines are given, kdm listens on all interfaces and joins the default XDMCP IPv6 multicast group (when compiled with IPv6 support). To disable listening for XDMCP requests altogether, a `LISTEN` line with no addresses may be specified, but using the `[Xdmcp] Enable` option is preferred.

## 5.4 Supplementary programs

The following programs are run by kdm at various stages of a session. They typically are shell scripts.

The Setup, Startup and Reset programs are run as `root`, so they should be careful about security. Their first argument is `auto` if the session results from an automatic login; otherwise, no arguments are passed to them.

### 5.4.1 Setup program

The `Xsetup` program is run after the X-Server is started or reset, but before the greeter is offered. This is the place to change the root background (if `UseBackground` is disabled) or bring up other windows that should appear on the screen along with the greeter.

In addition to any specified by `ExportList`, the following environment variables are passed:

**DISPLAY** the associated display name

**PATH** the value of `SystemPath`

**SHELL** the value of `SystemShell`

**XAUTHORITY** may be set to an authority file

**DM\_CONTROL** the value of `FifoDir`

Note that since kdm grabs the keyboard, any other windows will not be able to receive keyboard input. They will be able to interact with the mouse, however; beware of potential security holes here. If `GrabServer` is set, `Xsetup` will not be able to connect to the display at all. Resources for this program can be put into the file named by `Resources`.

### 5.4.2 Startup program

The `Xstartup` program is run as `root` when the user logs in. This is the place to put commands which add entries to `utmp` (the `sessreg` program may be useful here), mount users' home directories from file servers, or abort the session if some requirements are not met (but note that on modern systems, many of these tasks are already taken care of by PAM modules).

In addition to any specified by `ExportList`, the following environment variables are passed:

**DISPLAY** the associated display name

**HOME** the initial working directory of the user

**LOGNAME** the username

**USER** the username

**PATH** the value of `SystemPath`

**SHELL** the value of `SystemShell`

**XAUTHORITY** may be set to an authority file

**DM\_CONTROL** the value of `FifoDir`

kdm waits until this program exits before starting the user session. If the exit value of this program is non-zero, kdm discontinues the session and starts another authentication cycle.

### 5.4.3 Session program

The `Xsession` program is the command which is run as the user's session. It is run with the permissions of the authorized user. One of the keywords `failsafe`, `default` or `custom`, or a string to `eval` by a Bourne-compatible shell is passed as the first argument.

In addition to any specified by `ExportList`, the following environment variables are passed:

- DISPLAY** the associated display name
- HOME** the initial working directory of the user
- LOGNAME** the username
- USER** the username
- PATH** the value of `UserPath` (or `SystemPath` for root user sessions)
- SHELL** the user's default shell
- XAUTHORITY** may be set to a non-standard authority file
- KRBTKFILE** may be set to a Kerberos4 credentials cache name
- KRB5CCNAME** may be set to a Kerberos5 credentials cache name
- DM\_CONTROL** the value of `FifoDir`
- XDM\_MANAGED** will contain a comma-separated list of parameters the session might find interesting, like which conversation plugin was used for the login
- DESKTOP\_SESSION** the name of the session the user has chosen to run

#### 5.4.4 Reset program

Symmetrical with `Xstartup`, the `Xreset` program is run after the user session has terminated. Run as `root`, it should contain commands that undo the effects of commands in `Xstartup`, removing entries from `utmp` or unmounting directories from file servers.

The environment variables that were passed to `Xstartup` are also passed to `Xreset`.

## Chapter 6

# Creating themes for the kdm greeter

This section describes the creation of themes for the themed greeter. For examples including screenshots, see the installed standard themes and the themes from [the theme website](#).

### 6.1 Theme Overview

kdm themes can be created by creating an XML file that follows the specification in `$KDEDIR/share/apps/doc/kdm/greeter.dtd`. Theme files are stored in the directory `$KDEDIR/share/apps/kdm/themes/theme_name`. The theme directory should contain a file called `KdmGreeterTheme.desktop` which has similar format to other `.desktop` files and looks like:

```
[KdmGreeterTheme]
Greeter=circles.xml
Name=Circles
Description=Theme with blue circles
Author=Bond, James Bond
Copyright=(c) 2002 Bond, James Bond
Screenshot=screenshot.png
```

The `Name`, `Description`, `Author` and `Copyright` fields can be translated just like in other `.desktop` files. All the files that are mentioned should be in the theme directory itself. The `Screenshot` field points to a file which should be a 200x150 screenshot of the theme in action (it is OK not to have one, but it makes it nicer for the user). The `Greeter` entry points to an XML file that contains the description of the theme.

Once a theme has been fully tested, make a tarball that contains the directory as it would be installed to the `$(KDEDIR)/share/apps/kdm/themes` directory. This is the standard format for distributing kdm themes.

## 6.2 Detailed Description of Theme XML format

### 6.2.1 Toplevel Node

kdm themes are XML files with the `<greeter>` tag at their root. The toplevel node is an [item node](#) of type `rect` with an implicit [fixed layout](#).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE greeter SYSTEM "greeter.dtd">
<greeter>
[... ]
</greeter>
```

Contained within the greeter tag can be the nodes described in the next sections of this document. Some of these nodes are containers (layout nodes, item nodes) which can contain other nodes again.

### 6.2.2 Item Nodes

A kdm theme is created by specifying a hierarchy of item and layout nodes. Item nodes can have the following value for the `type` attribute:

**button** A button field. This field uses a Qt button.

It is also possible to make any other item act like a button by setting its `button` attribute to `true`. However, it is better to use Qt buttons in kdm themes since these are accessible to users with disabilities.

**entry** An input widget like a line edit or combo box. Note that this is merely a placeholder for Qt widgets.

**label** A text label. Must contain either a [text node](#) or a [stock node](#) to specify the text.

**list** A face browser widget.

**pixmap** A raster image in a format that Qt supports, e.g. PNG, JPEG, Tiff, etc.

**rect** A plain rectangle.

**svg** A vector image in SVG format.

For example:

```
<item type="label">
```

## The kdm Handbook

An item that acts as a button:

```
<item type="rect" id="disconnect_button" button="true">.
```

By default, the kdm login screen will disappear after authentication. This can result in flicker between the login screen and the session. The `background` attribute allows users to specify what elements of the theme are the background image. When used, this will cause kdm to remove all non-background items from the display and render the remaining background items to the root window. This can be used to create a smooth transition between the login screen and the session:

```
<item type="rect" background="true">
  <normal file="background.svg"/>
  <pos x="0" y="0" width="100%" height="-75"/>
</item>
```

To use a different background for login transition than the one used for login, the theme should specify two item nodes (which could contain pixmaps or svg images, for example). The item which corresponds to the greeter background should not have the `background` property while the item which corresponds to the transition background should have the `background` property. For instance :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE greeter SYSTEM "greeter.dtd">
<greeter>
  <item type="rect" background="true">
    <normal file="background_for_login.svg" element=" ←
      background"/>
    <pos x="0" y="0" width="100%" height="100%"/>
  </item>
  <item type="rect">
    <normal file="background_for_greeter.svg"/>
    <pos x="0" y="0" width="100%" height="100%"/>
  </item>
  [...]
</greeter>
```

In multi-screen setups, themes may also specify the look of other screens than the one the greeter is on - but typically only background items will appear there. To specify which screen(s) an item should appear on, the `screen` attribute can be used with the value being one of `greeter`, `other` or `all`, meaning the screen the greeter is on, all screens the greeter is not on and all screens, resp.

**button items and items with the `button="true"` attribute.** Buttons typically trigger certain actions. Additionally, kdm will hide buttons whose actions are not available for some reason.

*Id:* `chooser_button`, *Action:* Runs the XDMCP chooser.

*Id:* `disconnect_button`, *Action:* Disconnect from remote session.

*Id:* `session_button`, *Action:* Open the session type selection menu.

*Id:* `system_button`, *Action:* Open a catch-all menu with various options and actions, depending on the configuration.

**Label items** kdm will show/hide these labels and set their text depending on the state of the login dialog.

*Id:* `pam-error`, *Function:* This displays the Login failed. message.

**Widget embedding items** kdm will embed particular Qt widgets into these items.

*Id:* `user-entry`, *Function:* Entry field for username entry.

*Id:* `pw-entry`, *Function:* Entry field for password entry.

*Id:* `domain-entry`, *Function:* Some ‘conversation’ plugins use this field to query a domain name. If this field is present, the related enclosing items should have [show nodes](#) with the type `plugin-domain-entry`.

*Id:* `talker`, *Function:* This item should be of type `rect`. It represents the ‘hot’ area of the greeter: it contains the `label` and `entry` items which concern the authentication process. If a given ‘conversation’ plugin cannot match the existing items with its needs, it tries to embed a complex widget with an own layout into this item, thus completely overriding the theme’s ‘talker’. Strictly speaking, kdm themes do not need to provide own ‘talker’ designs at all, as all kdm authentication plugins are able make use of the `talker` item.

*Id:* `userlist`, *Function:* This item must be of type `list`. If the user list feature is enabled, kdm will embed the user list widget here. Otherwise, this item is hidden.

*Id:* `xconsole`, *Function:* This item should be of type `rect`. If the built-in **xconsole** feature is compiled in and enabled, kdm will embed the console log widget here. Otherwise, this item is hidden.

**Other items** kdm will show/hide these items depending on the configuration and the current state of the greeter. kdm does not impose type requirements on them, but they usually lend themselves to a particular type.

*Id:* `timed-label`, *Shown only when ...:* ... timed login is in progress.

*Id:* `caps-lock-warning`, *Shown only when ...:* ... Caps Lock is active.

*Id:* `xauth-warning`, *Shown only when ...:* ... the X-Server requires no X authorization to connect.

*Id:* `userlist-rect`, *Shown only when ...:* ... the user list is enabled. By nesting the `userlist` item into this one, it is possible to create something like a frame around the list and have it shown only when the user list itself is shown.

*Id:* `xconsole-rect`, *Shown only when ...:* ... the built-in **xconsole** is enabled. Analogous to `userlist-rect`.

### 6.2.3 Layout Container Nodes

Layout nodes appear inside item nodes and contain item nodes again. The type of the layout node determines the arrangement of its child nodes. An item node can contain one layout node of each type.

### 6.2.3.1 Box Nodes

Box nodes automatically arrange their children in a row. They are specified as follows:

```
<box orientation="alignment" min-width="num" min-height="num"
      xpadding="num" ypadding="num" spacing="num"
      homogeneous="bool">
```

Where *num* means number of pixels and *bool* means either true or false. The *alignment* value can be either horizontal or vertical. If you leave any attribute off, it will default to zero for numbers, false for bools and vertical for the orientation.

The spacing is the distance between neighboring child items. The padding is the box' outer margin. If the box is homogeneous, the same amount of space is allocated to each child item.

### 6.2.3.2 Fixed Nodes

Fixed is a container that has its children laid out at precise coordinates. The size of this container is the smallest rectangle that contains all the children. Fixed has no extra attributes and so you just use:

```
<fixed>
```

Then you put other items with proper position nodes inside it.

## 6.2.4 Position Nodes

Each item can specify its position and size via the `pos` node. For example:

```
<pos x="0" y="4" width="100%" height="100%"/>
```

If the size is not specified, it will be the item's 'natural' size, called the size hint. Note that not all items have a useful size hint.

Both position and size can be given in percent and will be calculated relative to the size of the enclosing container in this case. For toplevel items it is the percentage of the whole screen. By appending circumflexes (^) to the size specification it is possible to modify it to be relative to the size of the enclosing item's enclosing item and so on.

If the item contains a box, `width` and `height` can be specified to be `box` to mean that they are supposed to be the width and height of the box, that is the items in the box plus the padding.

One of `width` and `height` can be specified to be `scale` to mean that it should be scaled according to the other dimension's scale relative to its size hint. Use this to preserve the aspect ratio of scaled images automatically.

## The kdm Handbook

If the `expand` attribute is specified and `true`, this item will be expanded in the enclosing box as much as possible (that is it will be given more space if available).

If `width` or `height` is a plain number, a negative value represents an offset from the enclosing container's size. Note that it is possible to specify a positive offset by writing two minus signs.

In either case it is possible to constrain the final size with the `min-width`, `min-height`, `max-width` and `max-height` attributes which can be specified in the same ways as `width` and `height`.

If `x` or `y` is a plain number, a negative value represents an offset from the right resp. bottom edge, unlike the default which is the left resp. top edge.

It is also possible to specify which point within the item the position refers to. This is called the anchor and can be either `c` for center or one of `n`, `ne`, `e`, `se`, `s`, `sw`, `w` and `nw` which stand for the different edges/corners corresponding the directions on a topographical map. The default is `nw`, which is the upper left corner. For example:

```
<pos x="10%" y="50%" anchor="w" width="80%" height="95"/>
```

### 6.2.5 Show Nodes

You can specify the `type` attribute to indicate that certain items should only be displayed if the type is set. Prefixing the type with an exclamation mark (!) reverses the condition. Valid values include the following:

*Type:* `chooser`, *Display if ...:* switching to remote login is permitted.

*Type:* `halt` and `reboot`, *Display if ...:* system shutdown is permitted.

*Type:* `system`, *Display if ...:* no condition (always set in kdm).

*Type:* `plugin-entry-name`, *Display if ...:* the 'conversation' plugin provides a corresponding input field.

For example:

```
<show type="chooser"/>
```

Alternatively, you can specify a `min-screen-width` or `min-screen-height` value to indicate that certain items should be displayed only if the screen resolution is the at least the specified size. For example:

```
<show min-screen-height="768"/>
```

### 6.2.6 Normal/Active/Prelight Nodes

The look of most item types can be parametrized via the following tags:

**normal** Normal state.

**prelight** When the mouse is hovering over the item.

**active** When a mouse button is clicked on the item.

```
rect  
<normal color="#000000" alpha="0.0"/>
```

Either of the attributes may be omitted, in which case the default is used (the example represents the defaults). `alpha` is a floating point number between zero (transparent) and one (opaque). `color` is a hashmark followed by a six-digit hex number; the format is `'#rrggbb'`. Alternatively, `color` may be specified as an eight-digit hex number, in which case the first two digits are the alpha value.

```
label  
<normal color="#ffffff" alpha="1.0" font="Sans 14"/>
```

`alpha` and `color` are specified like in 'rect' items.

`font` follows the format `'family-list style-options size'`. Each part is optional.

`family-list` is a comma-separated list of font families like 'helvetica', 'monospace', etc.

`style-options` is a space-delimited list of keywords from the categories `style`, `weight` and `stretch`; from each category at most one. The style can be normal, italic or oblique. Weight can be ultra-light, light, medium, semi-bold, bold, ultra-bold or heavy. Allowable stretches comprise ultra-condensed, extra-condensed, condensed, semi-condensed, normal, semi-expanded, expanded, extra-expanded and ultra-expanded.

`size` is either a floating point number representing the size in points (1/72 inch) or an integer followed by `px` representing the size in pixels. Point sizes are preferable, as they are independent from the display resolution.

If either attribute is left out, the values from the [style node](#) are used. If this yields no window-text color specification, white is used. The default font is the one configured in `kdmrc`.

```
pixmap, svg  
<normal file="picture.png" tint="#dddddd" alpha="1.0"/>
```

`file` specifies the file containing the image. Relative pathnames are relative to the theme's directory.

`element` specifies the element id of a SVG file. If empty, the whole SVG image will be rendered.

For `pixmap` nodes, `kdm` first tries to open a file that matches `basename-widthxheight.extension` based on the specified file name, so the best-quality image for a given resolution can be used.

`scalemode` specifies how to adjust the size of images which do not match the element's size. `free` (the default) means to simply scale the image to

the right size, possibly distorting its aspect ratio. The other two modes maintain the image's aspect ratio: `fit` means to zoom the image to the maximal size which fits into the element's geometry. The image will be centered. The remaining area will not be painted by this element, so it should be placed on top of a solid-filled `rect`. `crop` means to zoom the image to the minimal size which completely fills the element's geometry. The image will be clipped symmetrically.

`tint` and `alpha` form a color specification like in `rect` items. Each pixel of the image is multiplied with this color component-wise.

### 6.2.7 Widget Style Nodes

This tag makes it possible to change the appearance of labels and Qt widgets embedded into the theme, e.g., line edits, buttons or the user list. The style settings are inherited by child items, but can be overridden individually. The defaults are taken from `kdmrc`.

The `font` attribute defines the font for all widgets. For widgets with an input function it can be overridden with the `edit-font` attribute. Fonts are specified the same way as in [normal/active/prelight nodes](#).

Usually, the theming engine tries hard to remove any frames from Qt widgets, so they melt into the theme seamlessly. In cases where this is not desired, the `frame` attribute can be set to `true`.

It is possible to specify almost the entire palette for the widgets as documented at [Trolltech's site](#). Attribute names are composed from a scope, a color role and a suffix. Possible scopes are - in order of increasing precedence - `all-` for all color groups, no scope for the active and inactive color groups and `active-`, `inactive-` and `disabled-` for the respective color group. Supported color roles are `window`, `window-text`, `base`, `alternate-base`, `text`, `bright-text`, `highlight`, `highlighted-text`, `button` and `button-text`. The suffix can be `-color` or `-alpha` with the respective meaning as in [normal/active/prelight nodes](#).

Example:

```
<style edit-font="Comic 16" text-color="#dddddd" frame="true ↵  
"/>
```

### 6.2.8 Face Browser Color Nodes

Color nodes permit overriding the background color of the items in the face browser. `labelcolor` and `altlabelcolor` are essentially equivalent to `all-base-color` resp. `all-alternate-base-color` in [style nodes](#). If only `labelcolor` is specified, alternating item backgrounds are disabled.

```
<color labelcolor="#80ffffff" altlabelcolor="#80f0f0"/>
```

## 6.2.9 Text Nodes

Text tags are used by labels. They can be used to display localized text as follows (if the `xml:lang` attribute is omitted, the POSIX locale is assumed):

```
<text xml:lang="fr">Option</text>
```

Text nodes can contain the following special character sequences which will be translated as follows:

*Sequence: %%, Expansion: A literal % character*  
*Sequence: %c, Expansion: Wall clock time and date*  
*Sequence: %d, Expansion: Display name (DISPLAY environment variable)*  
*Sequence: %h, Expansion: Hostname (gethostname output)*  
*Sequence: %m, Expansion: Machine name (machine part of uname output)*  
*Sequence: %n, Expansion: Node name (nodename part of uname output)*  
*Sequence: %o, Expansion: Domain name (getdomainname output)*  
*Sequence: %r, Expansion: Release name (release part of uname output)*  
*Sequence: %s, Expansion: System name (sysname part of uname output)*  
*Sequence: %t, Expansion: Remaining number of seconds until timed login is performed, plus the appropriate i18n plural form of 'second'*  
*Sequence: %u, Expansion: Username for timed login*  
*Sequence: \_, Expansion: Causes the following character to be an accelerator*

`%t` and `%u` are intended to be used only internally to display the `timed-label` message, which is automatically updated every second.

## 6.2.10 Stock Nodes

Certain common localized labels can be specified via the stock tags. The `text` tag is ignored if the `stock` tag is used. You really should use the stock labels rather than just putting all the translations into the themes. This yields faster load times and likely better translations. The following values are valid:

*Type: caps-lock-warning, Expansion: 'Caps Lock is enabled'*  
*Type: chooser, Expansion: 'XDMCP Choose\_r'*  
*Type: quit, Expansion: '\_Quit'*  
*Type: disconnect, Expansion: 'Disconn\_ect'*  
*Type: halt, Expansion: 'Power o\_ff'*  
*Type: language, Expansion: 'Lan\_guage'*  
*Type: login, Expansion: '\_Login'*  
*Type: session, Expansion: 'Session \_Type'*  
*Type: reboot, Expansion: 'Re\_boot'*  
*Type: system, Expansion: '\_Menu'*  
*Type: timed-label, Expansion: 'User %u will login in %t'*  
*Type: domain-label, Expansion: '\_Domain:'*  
*Type: username-label, Expansion: '\_Username:'*  
*Type: password-label, Expansion: '\_Password:'*  
*Type: welcome-label, Expansion: 'Welcome to %h'*

For example:

```
<stock type="welcome-label"/>
```

### 6.2.11 Buddy Nodes

Items which do not directly cause an action can be assigned a buddy. The buddy is given input focus when the item is activated (clicked or a label's accelerator pressed).

The buddy is referenced by id with the `idref` attribute. Obviously, it must be given an id. Example:

```
<item type="label">
  <stock type="username-label"/>
  <buddy idref="user-entry"/>
  [...]
</item>
[...]
<item type="entry" id="user-entry">
  [...]
</item>
```

## Chapter 7

# Configuring your system to use kdm

This chapter assumes that your system is already configured to run the X Window System®, and that you only need to reconfigure it to allow graphical login.

### 7.1 Setting up kdm

The fundamental thing that controls whether your computer boots to a terminal prompt (console mode) or a graphical login prompt is the default runlevel. The runlevel is set by the program `/sbin/init` under the control of the configuration file `/etc/inittab`. The default runlevels used by different UNIX® systems (and different Linux® distributions) vary, but if you look at `/etc/inittab` the start of it should be something like this:

```
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)

id:3:initdefault:
```

All but the last line of this extract are comments. The comments show that runlevel 5 is used for X11 and that runlevel 3 is used for multi-user mode without X11 (console mode). The final line specifies that the default runlevel of the system is 3 (console mode). If your system currently uses graphical login (for

## The kdm Handbook

example, using `xdm`) its default runlevel will match the runlevel specified for X11.

The runlevel with graphical login (`xdm`) for some common Linux® distributions is:

- 5 for Red Hat® 3.x and later, and for Mandrake™
- 4 for Slackware
- 3 for SuSE®. 4.x and 5.x

The first step in configuring your system is to ensure that you can start `kdm` from the command line. Once this is working, you can change your system configuration so that `kdm` starts automatically each time you reboot your system.

To test `kdm`, you must first bring your system to a runlevel that does not run `xdm`. To do so, issue a command like this:

```
/sbin/init 3
```

Instead of the number 3 you should specify the appropriate runlevel for console mode on your system.

If your system uses Pluggable Authentication Modules (PAM), which is normal with recent Linux® and Solaris™ systems, you should check that your PAM configuration permits login through the service named `kde`. If you previously used `xdm` successfully, you should not need to make any changes to your PAM configuration in order to use `kdm`. `/etc/pam.conf` or `/etc/pam.d/kde`. Information on configuring PAM is beyond the scope of this handbook, but PAM comes with comprehensive documentation (try looking in `/usr/share/doc/*pam*/html/`).

Now it's time for you to test `kdm` by issuing the following command:

```
kdm -nodaemon
```

If you get a `kdm` login dialog and you are able to log in, things are going well. The main thing that can go wrong here is that the run-time linker might not find the shared Qt™ or KDE libraries. If you have a binary distribution of the KDE libraries, make sure `kdm` is installed where the libraries believe KDE is installed and try setting some environment variables to point to your KDE and Qt™ libraries.

For example:

```
export
KDEDIR=/opt/kde
export
QTDIR=/usr/lib/qt4
export
PATH=$KDEDIR/bin:$QTDIR/bin:$PATH
export
LD_LIBRARY_PATH=$KDEDIR/lib:$QTDIR/lib
```

## The kdm Handbook

If you are still unsuccessful, try starting `xdm` instead, to make sure that you are not suffering from a more serious X configuration problem.

When you are able to start `kdm` successfully, you can start to replace `xdm` by `kdm`. Again, this is distribution-dependent.

- For Red Hat®, edit `/etc/inittab`, look for this line:

```
x:5:respawn:/usr/X11/bin/xdm -nodaemon
```

and replace with:

```
x:5:respawn:/opt/kde/bin/kdm
```

This tells `init(8)` to respawn `kdm` when the system is in run level 5. Note that `kdm` does not need the `-nodaemon` option.

- For Mandrake™, the X11 runlevel in `/etc/inittab` invokes the shell script `/etc/X11/prefdm`, which is set up to select from amongst several display managers, including `kdm`. Make sure that all the paths are correct for your installation.
- For SuSE®, edit `/sbin/init.d/xdm` to add a first line:

```
. /etc/rc.config  
DISPLAYMANAGER=kdm  
export DISPLAYMANAGER
```

- For FreeBSD, edit `/etc/ttys` and find the line like this:

```
ttv8 "/usr/X11R6/bin/xdm -nodaemon" xterm off secure
```

and edit it to this:

```
ttv8 "/opt/kde/bin/kdm" xterm on secure
```

- Most other distributions are a variation of one of these.

At this stage, you can test `kdm` again by bringing your system to the runlevel that should now run `kdm`. To do so, issue a command like this:

```
/sbin/init 5
```

Instead of the number 5 you should specify the appropriate runlevel for running X11 on your system.

The final step is to edit the `initdefault` entry in `/etc/inittab` to specify the appropriate runlevel for X11.

## The kdm Handbook

### WARNING

Before you make this change, ensure that you have a way to reboot your system if a problem occurs. This might be a 'rescue' floppy-disk provided by your operating system distribution or a specially-designed 'rescue' floppy-disk, such as `tomsrtbt`. Ignore this advice at your peril.

This usually involves changing the line:

```
id:3:initdefault:
```

to

```
id:5:initdefault:
```

When you reboot your system, you should end up with the graphical kdm login dialog.

If this step is unsuccessful the most likely problem is that the environment used at boot time differs from the environment that you used for testing at the command line. If you are trying to get two versions of KDE to co-exist, be particularly careful that the settings you use for your `PATH` and `LD_LIBRARY_PATH` environment variables are consistent, and that the startup scripts are not over-riding them in some way.

## Chapter 8

# Supporting multiple window managers

kdm detects most available window manager and desktop environments when it is run. Installing a new one should make it automatically available in the kdm main dialog's Session Type submenu.

If you have a very new window manager, or something that kdm does not support, the first thing you should check is that the application to be run is in the `PATH` and has not been renamed during the install into something unexpected.

If the case is that the application is too new and not yet supported by kdm, you can quite simply add a new session.

The sessions are defined in `.desktop` files in `$KDEDIR/share/apps/kdm/sessions`. You can simply add an appropriately named `.desktop` file in this directory. The fields are:

```
[Desktop Entry]
Encoding=UTF-8 This is fixed to UTF-8 and
may be omitted
Type=XSession This is fixed to XSession and
may be omitted
Exec=executable name Passed to
eval exec in a Bourne shell
TryExec=executable name Supported
but not required
Name=name to show in the kdm session list
```

There are also three 'magic' types:

**default** The default session for kdm is normally KDE but can be configured by the system administrator.

## The kdm Handbook

**custom** The Custom session will run the user's `/.xsession` if it exists.

**failsafe** Failsafe will run a very plain session, and is useful only for debugging purposes.

To override a session type, copy the `.desktop` file from the data dir to the config dir and edit it at will. Removing the shipped session types can be accomplished by 'shadowing' them with `.desktop` files containing `Hidden=true`. For the magic session types no `.desktop` files exist by default, but kdm pretends they would, so you can override them like any other type. I guess you already know how to add a new session type by now. ;-)

## Chapter 9

# Using kdm for Remote Logins (XDMCP)

XDMCP is the Open Group standard, the 'X Display Manager Control Protocol'. This is used to set up connections between remote systems over the network.

XDMCP is useful in multiuser situations where there are users with workstations and a more powerful server that can provide the resources to run multiple X sessions. For example, XDMCP is a good way to reuse old computers - a Pentium or even 486 computer with 16 Mb RAM is sufficient to run X itself, and using XDMCP such a computer can run a full modern KDE session from a server. For the server part, once a single KDE (or other environment) session is running, running another one requires very few extra resources.

However, allowing another method of login to your machine obviously has security implications. You should run this service only if you need to allow remote X Servers to start login sessions on your system. Users with a single UNIX® computer should not need to run this.

## Chapter 10

# Advanced Topics

### 10.1 Command Sockets

This is a feature you can use to remote-control kdm. It's mostly intended for use by ksmserver and kdesktop from a running session, but other applications are possible as well.

The sockets are UNIX® domain sockets which live in subdirectories of the directory specified by `FifoDir=`. The subdir is the key to addressing and security; the sockets all have the file name `socket` and file permissions `rw-rw-rw-` (0666). This is because some systems don't care for the file permission of the socket files.

There are two types of sockets: the global one (`dmctl`) and the per-display ones (`dmctl-<display>`).

The global one's subdir is owned by root, the subdirs of the per-display ones' are owned by the user currently owning the session (root or the logged in user). Group ownership of the subdirs can be set via `FifoGroup=`, otherwise it's root. The file permissions of the subdirs are `rxr-x---` (0750).

The fields of a command are separated by tabs (`\t`), the fields of a list are separated by spaces, literal spaces in list fields are denoted by `\s`.

The command is terminated by a newline (`\n`).

The same applies to replies. The reply on success is `ok`, possibly followed by the requested information. The reply on error is an `errno`-style word (e.g. `perm`, `noent`, etc.) followed by a longer explanation.

GLOBAL COMMANDS:

**login display** (*now* | *schedule*) *user password* [*session\_arguments*] login user at specified display. if *now* is specified, a possibly running session is killed, otherwise the login is done after the session exits. *session\_arguments* are printf-like escaped contents for `.dmrc`. Unlisted keys will default to previously saved values.

## The kdm Handbook

**resume** Force return from console mode, even if TTY logins are still active.

**manage *display* [*display\_class* [*auth\_name* *auth\_data*]]** Start managing the named foreign display.

*display\_class*, if specified and non-empty, will be used for configuration matching; see chapter 5.

*auth\_name* and *auth\_data* need to be passed if the display requires X authorization. The format is the same as the 2nd and 3rd column of the **xauth list** output.

**unmanage *display*** Stop managing the named foreign display.

### PER-DISPLAY COMMANDS:

**lock** The display is marked as locked. If the X-Server crashes in this state, no auto-relogin will be performed even if the option is on.

**unlock** Reverse the effect of **lock**, and re-enable auto-relogin.

**suicide** The currently running session is forcibly terminated. No auto-relogin is attempted, but a scheduled "login" command will be executed.

### COMMANDS FOR ALL SOCKETS

**caps** Returns a list of this socket's capabilities:

**kdm** identifies kdm, in case some other DM implements this protocol, too

**list, lock, suicide, login, resume, manage** The respective command is supported

**bootoptions** The **listbootoptions** command and the = to **shutdown** are supported

**shutdown <list>** **shutdown** is supported and allowed for the listed users (a comma separated list.) \* means all authenticated users.

**nuke <list>** Forced shutdown may be performed by the listed users.

**nuke** Forced shutdown may be performed by everybody

**reserve <number>** Reserve displays are configured, and number are available at this time

**list [*all* | *alllocal*]** Return a list of running sessions. By default all active sessions are listed. if *all* is specified, passive sessions are listed as well. If *alllocal* is specified, passive sessions are listed as well, but all incoming remote sessions are skipped.

Each session entry is a comma separated tuple of:

- Display or TTY name
- VT name for local sessions
- Logged in user's name, empty for passive sessions and outgoing remote sessions (local chooser mode)

## The kdm Handbook

- Session type or '<remote>' for outgoing remote sessions, empty for passive sessions.
- A Flag field:
  - \* for the display belonging to the requesting socket.
  - ! for sessions that cannot be killed by the requesting socket.

New fields may be added in the future.

**reserve** [*timeout in seconds*] Start a reserve login screen. If nobody logs in within the specified amount of time (one minute by default), the display is removed again. When the session on the display exits, the display is removed, too.

Permitted only on sockets of local displays and the global socket.

**activate** (*vt | display*) Switch to a particular VT (virtual terminal). The VT may be specified either directly (e.g. *vt3*) or by a display using it (eg; *:2*).

Permitted only on sockets of local displays and the global socket.

**listbootoptions** List available boot options.

The return value contains these tokens:

- A list of boot options (as shown in kdm itself).
- The default boot option.
- The current boot option.

The default and current option are zero-based indices into the list of boot options. If either one is unset or not determinable, it is -1.

**shutdown** (*reboot | halt*) [=bootchoice] (*ask | trynow | forcenow | schedule | start* [-1/end (for

Request a system shutdown, either a reboot or a halt/poweroff.

An OS choice for the next boot may be specified from the list returned by **listbootoptions**

Shutdowns requested from per-display sockets are executed when the current session on that display exits. Such a request may pop up a dialog asking for confirmation and/or authentication

*start* is the time for which the shutdown is scheduled. If it starts with a plus-sign, the current time is added. Zero means immediately.

*end* is the latest time at which the shutdown should be performed if active sessions are still running. If it starts with a plus-sign, the start time is added. -1 means wait infinitely. If end is through and active sessions are still running, kdm can do one of the following:

- *cancel* - give up the shutdown
- *force* - shut down nonetheless
- *forcemy* - shut down nonetheless if all active sessions belong to the requesting user. Only for per-display sockets.

*start* and *end* are specified in seconds since the UNIX® epoch.

*trynow* is a synonym for *0 0 cancel*, *forcenow* for *0 0 force* and *schedule* for *0 -1*.

*ask* attempts an immediate shutdown and interacts with the user if active sessions are still running. Only for per-display sockets.

## The kdm Handbook

**shutdown cancel** [*local* | *global*] Cancel a scheduled shutdown. The global socket always cancels the currently pending shutdown, while per-display sockets default to cancelling their queued request.

**shutdown status** Return a list with information about shutdowns.

The entries are a comma-separated tuples of:

- (global | local) - pending vs. queued shutdown. A local entry can be returned only by a per-display socket.
- (halt | reboot)
- start
- end
- ("ask" | "force" | "forcemy" | "cancel")
- Numeric user ID of the requesting user, -1 for the global socket.
- The next boot OS choice or "-" for none.

New fields might be added later

There are two ways of using the sockets:

- Connecting them directly. FifoDir is exported as `$DM_CONTROL`; the name of per-display sockets can be derived from `$DISPLAY`.
- By using the **kdmctl** command (e.g. from within a shell script). Try **kdmctl -h** to find out more.

Here is an example bash script 'reboot into FreeBSD':

```
if kdmctl | grep -q shutdown; then
  IFS=$'\t'
  set -- `kdmctl listbootoptions`
  if [ "$1" = ok ]; then
    fbsd=$(echo "$2" | tr ' ' '\n' | sed -ne 's,\\s, ,g;/ ↵
      freebsd/I{p;q}')
    if [ -n "$fbsd" ]; then
      kdmctl shutdown reboot "$fbsd" ask > /dev/null
    else
      echo "FreeBSD boot unavailable."
    fi
  else
    echo "Boot options unavailable."
  fi
else
  echo "Cannot reboot system."
fi
```

## Chapter 11

# Other sources of information

Since kdm is descended from xdm, the [xdm man page](#) may provide useful background information. For X-related problems try the man pages [X](#) and [startx](#). If you have questions about kdm that are not answered by this handbook, take advantage of the fact the kdm is provided under the terms of the GNU General Public License: look at the source code.

## Chapter 12

# Credits and License

kdm is derived from, and includes code from, xdm (c) Keith Packard, MIT X Consortium.

kdm 0.1 was written by Matthias Ettrich. Later versions till KDE 2.0.x were written by Steffen Hansen. Some new features for KDE 2.1.x and a major rewrite for KDE 2.2.x made by Oswald Buddenhagen.

Other parts of the kdm code are copyright by the authors, and licensed under the terms of the [GNU GPL](#). Anyone is allowed to change kdm and redistribute the result as long as the names of the authors are mentioned.

kdm requires the Qt™ library, which is copyright Troll Tech AS.

Documentation contributors:

- Documentation written by Steffen Hansen [steffen@klaralvdalens-datakonsult.se](mailto:steffen@klaralvdalens-datakonsult.se)
- Documentation extended by Gregor Zumstein [zumstein@ssd.ethz.ch](mailto:zumstein@ssd.ethz.ch). Last update August 9, 1998
- Documentation revised for KDE 2 by Neal Crook [nac@forth.org](mailto:nac@forth.org). Last update August 6, 2000
- Documentation extended and revised for KDE 2.2 and 4.0 by Oswald Buddenhagen [ossi@kde.org](mailto:ossi@kde.org). Last update December 7, 2007

Documentation copyright Steffen Hansen, Gregor Zumstein, Neal Crook and Oswald Buddenhagen. This document also includes large parts of the xdm man page, which is (c) Keith Packard. The theme format documentation is heavily based on the GDM manual, which is (c) Martin K. Petersen, George Lebl, Red Hat®, Inc. and Sun Microsystems, Inc.

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

## Chapter 13

# Glossary

**greeter** The greeter is the login dialog, i.e. the part of kdm which the user sees.

**entropy** The entropy of a system is the measure of its unpredictability. This is used during the generation of random numbers.