

The KTalkd Handbook

David Faure



The KTalkd Handbook

Contents

1	Introduction	1
2	Usage	3
2.1	Choosing a Forwarding Method	3
3	Questions and Answers	5
4	Copyright and Licenses	7
A	Installation	8
A.1	How to obtain KTalkd	8
A.2	Requirements	8
A.3	Compilation and Installation	8

Abstract

KTalkd is an enhanced **talk** daemon - a program to handle incoming **talk** requests, announce them and allow you to respond to it using a talk client.

Chapter 1

Introduction

KTalkd is an enhanced **talk** daemon - a program to handle incoming **talk** requests, announce them and allow you to respond to it using a **talk** client.

IMPORTANT

Note that KTalkd is designed to run on a single-user workstation, and shouldn't be run on a multi-user machine: since it reads users' configuration files, users can get the **talk** daemon to run any command, which is particularly dangerous. Do not use KTalkd if you create accounts on your machine, to people you don't fully trust.

In this document, if somebody wants to talk to you, you are designated as the 'callee'.

KTalkd has the following features :

Answering machine If the callee isn't logged on, or doesn't answer after the second announcement, an answering machine is launched, takes the message, and mails it to the callee.

Sound If desired, a sound is played with the announcement.

X Announce If compiled with KDE installed, KTalkd will use **ktalkdlg**, a KDE dialog, for announcement. If KTalk is running, it will be asked to make the announcement itself. (New since 0.8.8).

Multiple displays announcement If you are logged remotely (e.g. with an **export DISPLAY=...** command), the X announcement will be made on this display too. Answer on the one you want! If you're also logged in a text terminal, and if you're *not* using xterms (internal restriction), then you'll see a text announcement too, in case you're using the text terminal at the time of the announcement.

The KTalkd Handbook

Forwarding (*New since 0.8.0*) You can set up a forward to another user even to another host if you're away. There are 3 different forwarding methods. See section [Usage](#).

Configuration If KTalkd is compiled for KDE, it reads config from KDE config files, the sitewide (`$KDEDIR/share/config/ktalkdrc`) and the user one, in its home folder. The sitewide one has to be manually edited by the administrator, but there is now a configuration dialog for the user one. It's called **kcmktalkd** and can be found in the KControl after installing KTalkd. On non-KDE systems, KTalkd will read `/etc/talkd.conf`.

Internationalization Under KDE, the announcement will be in your language provided that you set it in the KDE menus and that someone translated **ktalkdlg** to your language. The same goes for the configuration dialog, **kcmktalkd**.

Support for otalk and ntalk (*New since 0.8.1*) KTalkd now supports both protocols, even when forwarding. KTalk supports both protocols as well.

I hope you will enjoy this talk daemon,

David Faure faure@kde.org

Chapter 2

Usage

To use KTalkd, you need a **talk** client. The text-based **talk** is available on most UNIX® systems. Try **talk your_username** to see what happens when you receive a **talk** request.

You can also try the answering machine the same way: initiate a **talk** to yourself, ignore the announcement twice, and you'll see the answering machine.

There is a **talk** client with a graphical interface for KDE, KTalk. It's not yet shipped with KDE packages, but you can find it on <ftp://ftp.kde.org>. It should be in <ftp://ftp.kde.org/pub/kde/stable/latest/apps/network>

The announcement dialog box is trivial: respond or ignore.

The configuration dialog should be rather straight forward, except for setting up a forward to another user (or even to another host).

2.1 Choosing a Forwarding Method

None is perfect, they all have pros (+) and cons (-).

FWA - Forward announcement only. Direct connection. Not recommended.

- (+) You know who the caller is, but
- (-) Caller will have to respond to an announcement from you. Annoying.
- (-) Don't use if you have an answering machine on your 'away' location. (The answering machine can't popup an announcement, it would be confusing!)

FWR - Forward all requests, changing info when necessary Direct connection.

The KTalkd Handbook

- (+) Caller won't know that you're away, but
- (-) You won't really know who the caller is - only his username, (so you might see `talk from Wintalk@my_host`)

FWT - Forward all requests and take the talk. No direct connection.

- (+) Same as above, but also works if you and caller can't be in direct contact one with the other (e.g. firewall).
- (+) You'll be told who's really talking to you when you accept the talk
- (-) But as in FWR, you won't know his machine name in the announcement

In short, use FWT if you want to use it behind a firewall (and if KTalkd can access both networks), and FWR otherwise.

Chapter 3

Questions and Answers

1. *Why doesn't root receive KDE announcements?*

Because this would be security hole, with the current user detection. You can bypass the limitation by adding two lines in **x**dm config files (which are the same as kdm ones).

NOTE

The S.u.S.E Linux® distribution includes those lines by default.

Those config files are normally in a folder such as `/etc/X11/xdm`, or `/usr/X11R6/lib/X11/xdm` on other systems. The following supposes that they are in `/etc/X11/xdm`, so you might have to translate them for another folder. Here is what you have to do:

- (a) Edit the file `Xstartup`, or create it, (in the **x**dm config folder) so that it reads:

```
#!/bin/sh
/etc/X11/xdm/GiveConsole
sessreg -a -l $DISPLAY -x /etc/X11/xdm/Xservers ↔
$USER
```

- (b) and the file `Xreset` so that it reads:

```
#!/bin/sh
/etc/X11/xdm/TakeConsole
sessreg -d -l $DISPLAY $USER
```

- (c) Make sure that `x`dm-config make reference to those two files:

```
DisplayManager._0.startup:      /etc/X11/xdm/ ↔
    Xstartup
DisplayManager._0.reset:        /etc/X11/xdm/Xreset
```

The KTalkd Handbook

This will make `kdm` (or `xdm`) log the user into `utmp`, which is the right thing to do. It's not up to `Konsole`, nor `xterm`, to log the user, but to `xdm` and `kdm`, in my opinion. However, this will not log the user as an X user when using `startx`... Any hint about that ?

2. *Why don't I, as a normal user, receive KDE announcements?*

If you're running a Linux® system (with `/proc` enabled), this behavior is a bug. Please send me a description of it so that I correct it. If you're running Linux® 2.0.35, this is a known bug in the kernel, which doesn't let `root` read `/proc`. The solution is the same as in the previous question, provided that you run `kdm` or `xdm` to log into X. Or upgrade! Otherwise, this is normal. KTalkd can't find the user, as KDE doesn't log him into `utmp` and the Linux® based (`/proc`) detection is disabled. The solution is the same as in the previous question, provided that you run `kdm` or `xdm` to log into X. Another solution is to make sure you always have an `xterm` running.

3. *How do I get debug output from KTalkd?*

As it is a daemon, there is no debug output on standard output. To get debugging output (for instance before submitting me a bug report!), update the lines in `inetd.conf` which launches KTalkd and KOTalkd to be:

```
talk    dgram    udp    wait    root    /usr/sbin/tcpd  / ↔  
        opt/kde/bin/ktalkd -d  
ntalk   dgram    udp    wait    root    /usr/sbin/tcpd  / ↔  
        opt/kde/bin/ktalkd -d
```

Notice the `-d` option. Then edit `/etc/syslog.conf` to add the following line:

```
*.*                /var/log/all_messages
```

To make it work, you then have to restart `inetd` and `syslogd`:

```
% killall -HUP inetd  
% killall -HUP syslogd
```

Finally, run a `talk` session and see the result in `/var/log/all_messages`. When submitting a bug report, never forget to include the debugging output, but also KTalkd's version number and the `./configure` output. Thanks.

Chapter 4

Copyright and Licenses

KTalkd is maintained and improved by David Faure, faure@kde.org

The original program was written by Robert Cimrman, cimrman3@students.zcu.cz

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

This program is licensed under the terms of the [GNU General Public License](#).

Appendix A

Installation

A.1 How to obtain KTalkd

KTalkd is now a core application of the KDE project <http://www.kde.org>, part of the kdenetwork package.

You can always download the latest KTalkd from the main FTP site of the KDE project, <ftp://ftp.kde.org/pub/kde> and from its mirrors. It's usually found in <ftp://ftp.kde.org/pub/kde/stable/latest/apps/network>

A.2 Requirements

In order to successfully compile KTalkd, you need the latest versions of the KDE libraries as well as the Qt™ C++ library. All required libraries as well as ktalkd itself can be found on <ftp://ftp.kde.org/pub/kde/>.

A.3 Compilation and Installation

In order to compile and install KTalkd on your system, type the following in the base folder of the KTalkd distribution:

```
% ./configure
% make
% make install
```

As KTalkd is a daemon, **make install** will require root privileges.

Don't forget to update `/etc/inetd.conf`. For example, on a Linux® system, if KDE is in `/opt/kde`, change the lines concerning **talk** and **ntalk** to:

The KTalkd Handbook

```
talk  dgram  udp  wait  root  /usr/sbin/tcpd  /opt/ ←
      kde/bin/ktalkd
ntalk dgram  udp  wait  root  /usr/sbin/tcpd  /opt/ ←
      kde/bin/ktalkd
```

A script is provided, to make the necessary change automatically. Update your `inetd.conf` file just by running

```
% ./post-install.sh
```

Anyway, you'll have to *restart `inetd`* after this. On most Linux® systems, do:

```
% killall -HUP inetd
```

On newer systems, using **xinetd**, there is no more `/etc/inetd.conf`, and you should edit or create `/etc/xinetd.d/talk` instead, with those lines:

```
service talk
{
    socket_type = dgram
    wait = yes
    user = root
    server = /usr/bin/ktalkd
}

service ntalk
{
    socket_type = dgram
    wait = yes
    user = root
    server = /usr/bin/ktalkd
}
```

then restart **xinetd**.

Please inform me of any modification you had to undertake in order to get KTalkd to compile or work on your platform.