The Linux XFree86 HOWTO

Helmut Geyer, Helmut.Geyer@iwr.uni-heidelberg.de v2.1 , 1. June 1994

This document describes how to obtain, install and configure XFree86™ for Linux systems. XFree86 version 2 is an enhanced version of the X Window System version 11 release 5 with support for many versions of UN*X running on Intel™ 386/486/Pentium™ platforms, including Linux. It supports a great amount of video hardware for these systems and includes many bugfixes beyond the MIT standard release of X. It is intended to help Linux users install and configure XFree86. It covers all versions of XFree86 starting from version 2.0. It is furthermore intended to answer some basic questions about X and programs using X. Please read this file and the referenced documentation files **BEFORE** installing and starting XFree86. **IF YOU ARE NOT CAREFUL, YOU MIGHT DAMAGE YOUR HARDWARE**. *If you have access to a printer capable of printing DVI or PostScript™ documents, you should consider to get those versions of this HOWTO as these provide far more readability by better formatting. A WWW version of this file and many other HOWTOs can be found in the* Linux Documentation Project (http://sunsite.unc.edu/mdw/linux.html) *homepage.*

Contents

1 Introduction

This file is meant to help you setting up your XFree86 system. It is, however, far from complete. You have to look to several other files or manpages to get complete information. It is noted at the appropriate places, which files have to be read. For a complete description on features and the installation please look at the documentation files and manuals that come with XFree86 (residing in `/usr/X386/lib/X11/etc` and its subdirectories as well as in `/usr/X386/man/man?`)[1]. The most important files will be pointed out to you in this HOWTO. The documentation files are packages separately in a file called `*-doc*` [cf. 4.1] from the XFree86 binary distribution.

Any bundeled distribution of XFree86 should include the documentation files. If it doesn't, please contact the distributor on how to obtain these.

Other manual pages from the stock X11R5 distribution can be found in the `*-man*` file of the XFree86 distribution. These manpages come as sources for the nroff program.

To read them you will need a working man program as well as the groff package for formatting them. Note that it is often required to read manpages, so you should install the groff package anyway. (I think it is no good idea that in many distributions this package is considered optional).

This file comes **WITHOUT ANY WARRANTY OF FITNESS**. If you damage anything following these informations, you are on your own.

1.1 Copyright notice

1.2 Requirements on reading this HOWTO

You should be aquainted to some very basic facts of a Unix, e.g. you should be able to read a text file or read a manpage. A simple introduction into Unix can be found in the *Installation Guide & Getting Started* document by Matt Welsh. Everyone who installes and runs Linux for the first time, should read this document. This document is written in LaTeX and can be obtained in several forms (PostScript, DVI, TeX or ASCII format) from `sunsite.unc.edu` in the directory `/pub/Linux/docs/LDP` or from `tsx-11.mit.edu` in the directory `/pub/linux/docs/LDP` by anonymous ftp.

[1]To read a manpage just run `man` *program_name*, the other documentation files are plain ascii and can be read with the editor of your choice or using `more` or `less`.

Many Linux distributions provide this document in at least one form. You should find it in the /usr/doc directory. In this directory you can find a lot of interesting documentation files, so you should check this directory and read the appropriate files.

1.3 Feedback

Comments, criticism or suggestions on this document are very welcome. If you think that something is missing, something is obsolete or wrong, or something could be done better, feel free to contact Helmut Geyer via email, Helmut.Geyer@IWR.uni-heidelberg.de.

1.4 New versions

The ASCII version of this document will be posted regularly to comp.os.linux.announce, comp.windows.x.i386unix, news.answers, and comp.answers. In addition to those sites that archive the answers newsgroups, the latest version can be found on sunsite.unc.edu in the directory /pub/Linux/docs/HOWTO. This site carries not only the ASCII version but also other formats (DVI, PostScript™ and HTML), too. These other versions are easier to read than the ASCII version, so if you have the resources to read or print these other versions, do it.

2 What are X11 and XFree86?

X11 is a windowing system for UN*X like operating systems. The X Windowing System was issued by the MIT Consortium (look at the X manpage for more information) and is put under a very liberal copyright, that allows any use of source code provided that the original copyright notices are included. The release of X11 that XFree86 2.* is based upon is X11 Release 5. The work of the MIT Consortium is continued by the X Consortium that just released a new release of X, X11R6. The next official XFree86 release will be based on X11R6 [cf Appendix B]. As X is the standard windowing system for UN*X operating systems, there are a lot of applications using it (both free and commercial).

XFree86 2.* is a port of X11R5 that supports several versions of Intel-based Unix and Unix-like operating systems. The XFree86 servers are partly derived from X386 1.2, which was the X server distributed with X11R5, but many of them are newly developed. The accelerated servers that are the most important change since XFree86 1.3 are completely new. This release consists of many new features and performance improvements as well as many bug fixes. The release is available as source patches against the MIT X11R5 code, as well as binary distributions for many architectures.

Note that while the source and installation trees retain the name 'X386' name (for simplicity of maintenance of the source tree), there is no connection between XFree86 and the commercial X386 product sold by SGCS. The XFree86 Project maintains technical contacts in an effort to keep user-affecting changes to the workings of the products from diverging too radically. There is no direct involvement of either group in the workings of the other.

2.1 The XFree86 Project, Inc.

The XFree86 team had several problems in the past due to the fact that it was no legal organisation. So it was not possible to become a member of the X Consortium, Inc. This was sometimes annoying as the new releases are only provided for members before the official release date. This and several other thoughts led to the foundation of The XFree86 Project, Inc.

XFree86 is now a trademark of The XFree86 Project, Inc. For more information read the section on The XFree86 Project in the file `/usr/X386/lib/X11/etc/README` from the documentation tarfile.

3 Supported Hardware

As this is the most often asked question, this will be the first section. Currently XFree86 (Version 2.0 - 2.1.1) supports the following chipsets in the XF86_SVGA server (note that it depends on the chipset whether a card is working and not on the vendor, with Diamond notably excluded, cf. the remark at the end of this paragraph):

- nonaccelerated chipsets:
  - Tseng ET3000, ET4000AX, ET4000/W32
  - Western Digital/Paradise PVGA1
  - Western Digital WD90C00, WD90C10, WD90C11, WD90C24, WD90C30
  - Genoa GVGA
  - Trident TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA9000, TVGA9000i, TVGA9100B, TVGA9200CX, TVGA9320, TVGA9400CX, TVGA9420
  - ATI 28800-4, 28800-5, 28800-6, 28800-a
  - NCR 77C22, 77C22E, 77C22E+
  - Cirrus Logic CLGD6205, CLGD6215, CLGD6225, CLGD6235
  - Compaq AVGA
  - OAK OTI067, OTI077
- accelerated chipsets:
  - Cirrus GLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428
  - Western Digital WD90C31

Note that the WD90C33 is currently not supported by any of the non-generic servers. There are additional problems with the generic servers of XFree86 2.0, too. Please look at the section on bugs and fixes.

These chipsets are all supported in 256 color (XF86_SVGA) and in monochrome mode (XF86_Mono) with the exception of the Cirrus chipsets, which are only supported in 256 color mode[2]. Note that the ET4000/W32 is only supported as a ET4000 clone and none of the accelerated functions are used.

[2] Note that if you have a grey scale display you need to run the colour server as grey scale behaves just like colour, not like monochrome mode

The monochrome server also supports generic VGA cards (using 64k of the video memory as a single bank), the Hercules card and the Hyundai HGC-1280. Note that these drivers are not part of the XF86_Mono server by default. If you want to use these, you have to reconfigure your XF86_Mono server using the LinkKit (or, of course, the source distribution - but the LinkKit will need less disk space and less compile time). See the Appendix for further information on using a Hercules mono card.

There is an (experimental) additional server that works on generic VGA hardware: XF86_VGA16, a 16 color server.

XFree86 supports the following accelerated chipsets with separate servers:

- S3 86C911, 86C924, 86C801, 86C805, 86C928 supported by the XF86_S3 server
- ATI mach8 supported by the XF86_Mach8 server
- ATI mach32 supported by the XF86_Mach32 server
- IBM 8514/a and true clones supported by the XF86_8514 server

Currently the S3 Vision864 and Vision964 are *not* yet working.

No other chipsets are supported; not Weitek P9000, not TIGA, not IIT AGX, not Microfield, not the new MGA chipsets (as in the Matrox cards), etc. Support for some of these will perhaps be provided in future XFree86 releases. TIGA will never be supported, as it requires licensing materials from TI (and may disallow source distributions); Microfield boards will never be supported, as they use proprietary and undocumented custom microcode interfaces. **Note further:** The following is a statement of the XFree86 Project concerning graphic cards by Diamond:

```
All Diamond cards are NOT supported by XFree86 even if they
have a supported chipset (with  the exception of the Cirrus
chipsets that have an internal clock generator). The reason
for this is that Diamond has changed  the mechanism used to
select  pixel   clock  frequencies, and  will only  release
programming information under  non-disclosure.   We are not
willing to do this (as it would mean  that source cannot be
provided).  We have had discussions with Diamond over this,
and they do not  intend to  change  this policy.  Hence  we
will do nothing  to support Diamond products  going forward
(i.e.  don't  send us a program  to run  set their clocks).
XFree86 DOES NOT SUPPORT  DIAMOND HARDWARE.  It is possible
to make some  of it work, but  we will not assist  in doing
this.
```

The author of this HOWTO does support this statement completely. If you have read this and bought nevertheless a card from Diamond, you will damage the idea of free software as Diamond does not support the efforts of a free X11 server. There is enough good and affordable graphics hardware available from companies that support XFree86.

However, if you are stuck with a card from Diamond, there is a compilation of the needed things to get XFree86 running on some of these cards. It is called the Diamond.FAQ and can be obtained from sunsite in the X11 directory.

Do not ask the XFree86 team about Diamond hardware. You will simply annoy them, and get no information.

For some of these chipsets there are specific `README` files (to be found in `/usr/X386/lib/X11/etc`). If there is one for the chipset you use, read it! In these `README`s the specific options that can be used to configure the server are explained. Currently there are special `README`s for ATI, Trident, Tsengm Western Digital, and Cirrus chipsets.

More information on the servers can be found in their manpages.

Later more on configuring the servers [cf. 5.1].

4 Installing XFree86

4.1 Where do I get XFree86 and what do I need to run it?

This section is dedicated to the Linux binary distribution so all file names and site names are for Linux only. If you read this file and are not using Linux, you should get your hands on the announcement of the according version of XFree86 for you OS. In it you will find all necessary information on obtaining it.

You can get the binary distributions of XFree86 for Linux via anonymous FTP from

```
tsx-11.mit.edu:/pub/linux/packages/X11/XFree86-ver
        or
sunsite.unc.edu:/pub/Linux/X11/XFree86-ver
```

where `ver` is the version number of the XFree86 release you want to obtain (e.g. `2.1` for XFree86 2.1). The files have names of the form `XF86-ver-name.tar.gz` (versions 2.1 and 2.1.1) or `xf86-name-2.0.tar.gz` where `ver` is the XFree86 version and `name` is the name of the package. If you already know, which server you need to run, you should not get the `*-svr*` file but the file `XF86_servername.tar.gz` where `servername` is the name of the server to run.

To get Version 2.1.1, which is only a bugfix release, you have to get the complete 2.1 distribution but the server files. In addition you have to get the 2.1.1 files (you will probably need only one server file, but you should get all the rest). Install the 2.1 files first and install the 2.1.1 files only after that.

Of course you can find it on all mirrors of those two, too. So please look for a ftp site near to you mirroring `sunsite` or `tsx-11` to save bandwidth.

You can find an incomplete list of sites mirroring these two in the `bbs-list` to be found in the docs directories of both sites. If you have access to archie, try using it to find a site carrying Linux in your neighbourhood.

The distribution consists of several gzipped tarfiles. As some of them are too large to fit on a single floppy, so you might have to split those). For a concise description on what the tarfiles contain, look at the `README` file in this directory.

You will need the following setup to run XFree86 (split by versions). You have to have at least these versions, later versions will work, too.

```
XFree86 version:         2.0         2.1          2.1.1

kernel version:      0.99pl13     0.99pl15  0.99pl15h
C library version       4.4.1       4.5.21     4.5.21
ld.so version:            1.3        1.4.3      1.4.3
```

You can find the most recent versions on most linux sites [cf. 4.1]. You can find the C library (`libc`) and the dynamic loader (`ld.so`) in the `GCC` directory on these sites.

You will need a main memory of at least 8 MB and a virtual memory of at least 16 MB (i.e. main memory + swap). It is possible to run X on a 4 MB machine if you take some precautions on memory usage [cf. 6].Note that you should nevertheless have 16 MB of virtual memory (so if you have 4 megs of physical RAM, you should have 12 megs of swap). It is recommended that you have *at least* 8 megs of physical RAM, as swaping is very slow. With only 4 megs of physical RAM, performance will suffer greatly. To run X11 comfortably, 8 MB are needed.

If you want to run memory-hog programs from within X (as e.g. `gcc`) you should at least have 16 MB of main memory and another 16MB of swap.

You will need about 17 MB of disk space for the complete installation without LinkKit, 21 MB with LinkKit. By deleting those servers that you don't need and removing the LinkKit after linking your own server, you can save several MB of diskspace.

Before installing XFree86, you should make a backup of all files that you changed. They may not be usable, but they still hold a lot of information you might want to preserve. (Your old XConfig file will not be deleted, but it's allways better to have a backup of this file)

To install the binary distribution you have to do the following as root with all needed tarfiles from above. For installing XFree86 2.0 (which is not recommended) you have to do it from `/`, for 2.1 or 2.1.1 you have to run first `mkdir /usr/X386 ; cd /usr/X386` [cf 10]. To get all permissions right you should run `umask 022` before installing it.

```
gzip -dc tarfilename | tar xvvof -
```

WARNING: This will overwrite all files from an older XFree86 version, of course not the `Xconfig` file (if you did not link it to `Xconfig.sample`, which you should **not** do). In other words, only if `Xconfig.sample` is linked to your `Xconfig`, it will be overwritten. Many other files will be overwritten, though (e.g. the `xdm` configuration files, the system wide `twm` configuration file, ...). So you should backup every file you changed before installing XFree86.

After installing it, you have to set it up correctly to match your system [cf. 5].

4.2 How do I compile it myself and where can I get the sources?

You do *not* want to do this. Only if you really want to make changes to the sources because something is not working you should consider this. You will need a lot of disk space and CPU time to do a complete build of the XFree86 system. Anything you need to know for compiling XFree86 you can find in the following files (to be found in `/usr/X386/lib/X11/etc`): `INSTALL, README, and README.Linux`.

Note that you should **not** compile XFree86 to get rid of some hardcoded restrictions (e.g. on the maximal pixel clock) as without these restrictions your hardware will probably break down.

To build a server that only includes those drivers you need, you should use the LinkKit instead of compiling the complete system. This is much easier. Read `/usr/X386/lib/Server/README` for a description how to do this. This file is *not* included in the standard XFree86 tarfiles but is part of the file that includes the LinkKit. This file is called `*-kit*` or `*-lkit*`.

For adding drivers to the SVGA servers you will only need the LinkKit. The documentation how to do this can be found in the `/usr/X386/lib/Server/VGADriverDoc` directory after installing the LinkKit package.

5 Setting up your XFree86 system

This section gives a short introduction on how to set up a XFree86 system on your computer. The most important (and most difficult) part is setting up a `Xconfig` file. If you have XFree86 2.1.1 and your graphic card is in the data base, you should use `ConfigXF86` to do this. It is comfortable and safe that way. You can use `ConfigXF86` for many things even if your card and monitor are not known to the program. Please use it if you have XFree86 2.1.1.

5.1 The Xconfig file

To be able to set up a `Xconfig` file, you need to read the following files that reside in `/usr/X386/lib/X11/etc`: `README`, `README.Config`, `VideoModes.doc`, and `README.Linux`.

Furthermore you need to read the following manpages: `Xconfig`, `XFree86`, `XFree86kbd`, and that of the server you are going to use. This section will try to give you an approach to these files as well as making some of the statements in those more understandable for the unexperienced user.

The `Xconfig` file can be located in several places:

- `/usr/X386/lib/X11`. This is the standard location for the sample `Xconfig` file, but is in some cases not possible to use (e.g. a read-only `/usr` partition).
- `/etc`.
- In the user's home directory.
- As `Xconfig.hostname` in `/usr/X386/lib/X11`.

The Linux file system standard places the Xconfig file into `/etc/X11`. This has several causes, but you should never edit that file. The XFree86 servers will not expect a `Xconfig` file at this location, so there has to be a link from one of the above places to this location. Find this link first and access the file using the link. This way you can be sure your changes will take effect.

To give you some hints, here is a list of what you need to set up the `Xconfig` file correctly:

- The server that is suitable for you system. To get a hint which is the correct one, run the `SuperProbe` program that comes with XFree86. It will identify your chipset and you can look at the table above [cf. 3], which server supports this chipset. Note that `SuperProbe` can detect *far* more hardware than XFree86 supports.
- The specifications of your monitor. Most important are the maximal horizontal and vertical scan frequency ranges and the bandwidth of the monitor. This information should be obtained from your monitor's datasheet. See the video mode section 5.1.4 for further information.
- The name of the chipset for your card. See 5.1.3 for more information.
- The available dot clocks for your card or (if supported) the name of the programmable dot clock generator. How to obtain these is explained concisely in `/usr/X386/lib/X11/etc/README.Config`. This will be done automagically by `ConfigXF86`. Note that if you have a programmable clock generator, you have to look in the `Xconfig` manpage, whether it is supported. A programmable clock cannot be detected properly.
- Mouse type (this refers to the protocol the mouse is using, not to the manufacturer, e.g. a serial microsoft mouse connected to the ps/2 port uses the `ps/2` protocol, not the `Microsoft` protocol). See 5.1.2 for more information.
- The device name of the device your mouse is connected to. See 5.1.2 for more information.
- If you want to use a national keyboard map for the kernel (i.e. you are not running the generic US keytable), you should read 5.1.5 to set up XFree86 correctly.

You should read `README.Config` that is located in `/usr/X386/lib/X11/etc` and follow the instructions there to the point. Especially the caveats that can be found int this file **must** be obeyed, otherwise you might damage your hardware.

WARNING: Do *not ever* share `Xconfig` files with people who do not have **exactly** the same Configuration (i.e. graphic card *and* monitor). By doing this you could toast your monitor. It isn't so hard to figure out modes (that is, for multisync monitors) and you surely shouldn't ever use a mode that you didn't check by yourself to be within your monitor's specs. Even if you have exactly the same setup you should check all modes yourself before trying them. There are many people who run their hardware out of specs which may not damage their hardware but could yours.

5.1.1 Running ConfigXF86

From XFree86 2.1.1 on there is a great tool available for XFree86, called `ConfigXF86`. It was written by Stephen Zwaska, comments and new database entries should be sent to him (`stz@netcom.com`). This utility will give you an easy interface to the database of tested graphic cards and monitors and it gives you some tools to configure your `Xconfig` file correctly. The documentation for `ConfigXF86` is located in `/usr/X386/lib/X11/ConfigXF86`. Please read this first, it will help you to run this program. This documentation is provided in several formats. A text version (`ConfigXF86.txt`) can be read online, a PostScript version (`ConfigXF86.PS`) can be printed if a PostScript printer is available. `ConfigXF86` can be obtained from `sunsite` in the directory `/pub/linux/packages/X11/ConfigXF86`.

After having read the documentation, start `ConfigXF86` as `root`. The program cannot be run while a X server is running on the system, so run `ConfigXF86` (and if needed `MakeCard`) from the console. You should be able to answer all questions. The program will probe your system and will (at least partly) recognize the setup. If both your graphic card and monitor are supported, you should not have any problems. The only thing you should do is test the video modes and (perhaps) try to make minor changes to the video modes to suit your monitor better [cf. 5.1.4].

This utility is very new, so there might be bugs (very probably there are). If you encounter any problems, please report them.

If your card is not supported, try to make a card database file using the `MakeCard` program that can be found in `/usr/X386/lib/X11/ConfigXF86`. Please read the next two sections before running it as it may help you with some questions. Having done that you can run `ConfigXF86` and choose the datafile you just generated. Don't forget to send the card data file and the monitor data files to `stz@netcom.com` so that other people may benefit from your achievements.

5.1.2 The easy sections

Remark: The XFree86 servers parse the `Xconfig` file in case-insensitive mode, so you need not woory about Capitalization.

Paths: There is no reason to fiddle with the standard paths as provided in the sample `Xconfig` file (any distribution that provides a different path structure should have edited this section of the `Xconfig.sample` or the template Xconfig file for `ConfigXF86`). The only thing that could be interesting is the use of a fontserver. Look at the `Xconfig` manpage if you want use one, anything you need to know is shown there. `ConfigXF86` will prompt you whether you want to use a fontserver.

The keyboard section: This is fairly easy. You should set the `ServerNumlock` option, but otherwise only those keyboard modifications that are needed for international keyboard support should be needed to set [cf. 5.1.5]. So this section should look like:

```
keyboard
    servernumlock
    RightAlt ModeShift
```

The mouse section: The keyword for the mouse section is the name for the protocol the mouse uses. The available protocol names are listed in the `Xconfig` manpage. Some remarks on the mouse keywords:

The Logitech^{TM} serial mice use several keywords: The MouseMan^{TM} uses the `MouseMan` keyword and more recent Logitech serial mice use the `Microsoft` keyword. Older Logitech serial mice use the `Logitech` keyword.

Any mouse connected to the ps/2 port uses the `ps/2` keyword (even if it is in fact a serial mouse).

As there is some confusion on the Linux mouse devicenames, here is a list. The standard names confer to the names as of `MAKEDEV` 1.2. If you are not sure, which kind of busmouse you have, look at the startup messages of the kernel. It will identify the busmouse type. You should make sure that the kernel busmouse driver is using the same IRQ as the busmouse. If this is not the case, you have to change this (either on the mouse board or in the kernel). The IRQ for busmouse devices is given in `/usr/src/linux/include/linux/busmouse.h`. The macro `MOUSE_IRQ` contains this IRQ. It is set to 5 by default.

- `/dev/atibm` for the ATI XL busmouse (*Note*: the ATI GU busmouse is a Logitech or Microsoft busmouse depending on the version you got).
- `/dev/logibm` for the Logitech busmouse (*Note*: this uses the `Busmouse` protocol, *not* the `Logitech` protocol).
- `/dev/inportbm` for the Microsoft™ busmouse (*Note*: this uses the `Busmouse` protocol, *not* the `Microsoft` protocol).
- `/dev/psaux` for a ps/2 or quickport mouse (uses the `ps/2` protocol).
- The other supported mice are serial mice, therefore the device names are that of the serial devices (`/dev/ttyS?` or `/dev/ttyS??` for Linux).

There are still distributions around that use old or different device names than those of `MAKEDEV`. A list of the major and minor numbers for the mouse devices can be found in the`/dev/MAKEDEV` script. If it is missing you can find it on `sunsite` in the `/pub/Linux/system/Admin` directory [cf 4.1].

If you have a two-button mouse, you might want to emulate the third button (emulated by pressing both buttons simultaneously) by setting `Emulate3Buttons` in the mouse section. There are quite a lot of other settings available that are usually not needed. Look at the `Xconfig` manpage for a list of available settings.

5.1.3 The server section

You should be able to set up this section by using the `ConfigXF86` program. If your graphic card is not part of the database, run `MakeCard` to build a new data file for your card. Many important things are probed by `MakeCard`, but you may need or want to set additional options. These options are described in the manpage of the server you want to use (`MakeCard` will suggest the right one) and in `README.Config`. After running `MakeCard`, run `ConfigXF86` to set up the `Xconfig` file. If you do not have the `ConfigXF86` package, you should follow the instructions in `README.Config` to get everything right.

If you want to identify the chipset your graphic card uses, run `SuperProbe`, a program that comes with XFree86 and is capable to identify a lot of graphic hardware. Note that `SuperProbe` can probe far more hardware than XFree86 supports.

5.1.4 Setting up video modes

This is the hardest part. Please read `VideoModes.doc` before doing anything here. If you are using `ConfigXF86` (which is *strongly* recommended), and your monitor is not in the database, you should choose the generic modes and start making own modes from there. If you do not have `ConfigXF86`, a tutorial on building modes can be found in the `README.Config` and `VideoModes.doc` files.

`ConfigXF86` includes a very neat utility to tune video modes. As there is no check on the fitness of a mode, you have to check the mode data against your monitor's specifications before testing the mode. The first line of the screen for tuning modes gives you information on the specifications of the mode. You have to check *always* that these values are within you monitor's capabilities before testing that mode.

5.1.5 International keyboard layout for XFree86

From version 2.1 on the XFree86 servers are able to read the keytable from the Linux kernel. So you need to set up only one keyboard layout file (that for the kernel). There are some restrictions, though (the kernel can support more keyboard functions than X11). X11 can only use 4 keytables (i.e. only one modifier is allowed, called `ModeShift`). This can be configured using the `ModeShift` line in the `Xconfig` file. Configurable keys for the ModeShift modifier are `LeftAlt`, `RightAlt` (=`AltGr`), `RightCtl`, and `ScrollLock`. By default the `ModeShift` modifier is not used.

Usually the `AltGr` key is used for national keyboard modifications. To enable the XFree86 server to read the `AltGr` keytable from the kernel, you should put

```
RightAlt "ModeShift"
```

into the keyboard section of the `Xconfig` file.

Besides supporting only one additional keymap X11 cannot use 'dead' keys. A key is called 'dead', if it will not print a character when it is typed but only after a second character is typed. Depending on the following character, the second character is modified by the 'dead' character or two characters will be printed. A typical example are accent keys. Such keys are not supported by X11. So you need to replace all dead keysymbols non-dead equivalents. Here is a list, what you have to change:

dead	non-dead
dead_tilde	asciitilde
dead_grave	grave
dead_circumflex	asciicircum
dead_acute	apostrophe
dead_diaeresis	diaeresis

Instead of supporting dead keys XFree86 supports a `Compose` key. This feature is described in the `XFree86kbd` manpage and can be modified by assigning the `Compose` function to one of the keys mentioned above. By default the `ScrollLock` key has the `Compose` function.

If you still want to have the 'dead' keys on the console, you will have to use a xmodmap file to map the keys to the right symbols under X. This is also the method that must be used with earlier versions of XFree86. On `sunsite` in the directory `/pub/Linux/X11/misc` example xmodmap files for several languages can be found. Note that you have to set the `ModeShift` modifier nevertheless to get the right keytable working.

Please read the `kbd.FAQ` that comes with the kbd package for Linux. You will find many hints on modifying your keyboard layout on the console as well as for X.

5.2 The .xinitrc file

To use X, you need a startup file that calls the local modifications, the window manager and some application you want to have right after X has started. If you are using startx (or runx) to start X, this startup file is called '`xinitrc`'. There is a standard xinitrc file, `/usr/lib/X11/xinit/xinitrc`[3] . If you are not content with what this file does (e.g. you want to use a different window-manager), you should copy this file to the file `.xinitrc` in your home directory. After copying the file, you can edit it. Look at the manpages for `startx` and `xinit` for more information.

Note that both the `.xinitrc` *and the* `.Xresources` *file must be readable and executable, so run* `chmod u+rx .xinitrc` *and* `chmod u+rx .Xresources` *after editing them.*

5.3 The personal X resource file

Sometimes you won't be content with default settings for several applications which often don't have a configruration file of their own. You can change some of these defaults by setting X resources in the `.Xresources` file in your home directory. Note that you should know what effects the setting of the resources has to the programs you use. *Read the manpages for the program and for* `xrdb` *before fiddling with the* `Xresources` *file. This section is not needed to set up working X environment, but to customize a working one.*

A resource file looks like an application default file (with the difference that in the resoucre file resources for several applications are set, you should use the full names (Progname.Resourcename) instead of abbreviating the program name by an asterisk. Examples of application default files can be found in the `/usr/X386/lib/X11/app-defaults` directory. Which resources are available for a single application is usually shown in the manpage of the application.

If you are running a colour server, you might want to put

```
#ifdef COLOR
*customization: -color
#endif
```

into your `.Xresources` file if some programs start up b&w and not colourful. If the upper change is made (it definetly should be in the system Xresources file, so you might hint that to the distribution maintainer), a program `foo` will read both the `Foo` and the `Foo-color` application default file from `/usr/X386/lib/X11/app-defaults`, not just the first which is the usual behaviour.
Note that the b&w outfit of a program may caused by the program not supporting colours.

5.4 Using xdm

If you want to run X on your system all the time, you could run `xdm` from the system startup. `xdm` is preconfigured on most systems, so you should not have to edit any of the `xdm` configuration files. Usually

[3]This is the traditional location for this file. The Linux filesystem standard places this file to `/etc/X11/xinit/xinitrc` in order to allow a read-only mounted /usr partition. So you should look at that location first.

a runlevel is attached to a X-only system (look into `/etc/inittab`), all you have to do to get it working is changing the default runlevel. On systems that use an `init` without runlevels (run `man init` to see, whether or not), you should look into the `/etc/rc` and `/etc/rc.local` files, you usually only have to remove comment signs at the beginning of a line that calls `xdm`. If no such line is present, you probably have a system that has no preconfigured `xdm`. Look at the manpages to see what has to be done (and complain to the distribution maintainer, if the distribution contains XFree86). If you install XFree86 on top of a distribution that is not providing X support, you should get enough information from the manpages. Help is also available from the X Faq, look at section 9 where to get this.

`xdm` runs by default the .xinitrc [cf 5.2]and the .Xresources [cf 5.3] files, so look at the sections above.

5.5 Configuration of the window manager

Window managers are a very local issue. There are quite a lot window managers available [cf. 7]. Configuration of one windowmanager is quite different from that of another. You should look at your `.xinitrc` file or the system `xinitrc` [cf 5.2] file, which window manager is used (hint: most window managers have a `wm` ending) and look at its manpage. Configuration is usually explained there. The most common used window managers for Linux are:
`twm` (part of the XFree86 distribution), `fvwm` (this seems to be the most popular), `olwm` or `olvwm` (from the Xview™ package), or mwm (part of Motif™).

5.6 Running X

Once you have set up a `Xconfig` file, you probably want to run X. This is **not** done by running `X`, but by running `startx`. When running it for the first time run `startx 2>1 > Xstartup.log` [4] to get a log file of the X startup. If problems are encountered, this file will help you by providing information for you and for anyone you ask about your problems.

If you encounter any problems, please look at the Appendix C and at the XFree86 documentation files in `/usr/X386/lib/X11/etc`. These files provide a lot of information on what to do if problems occur. If you still cannot solve the problems, procede as in Section 9. The FAQs that are mentioned there provide help on many problems you have with X, so you should get these and read them, too.

You should make sure that the `PATH` and `MANPATH` (or `path` and `manpath` for `csh` derivates) variables are set correctly.

6 tinyX - a XFree86 based package for low memory systems

Note that while you can run X on 4 MB RAM systems using tinyX, it is not useable for more complex applications or several applications at the same time.

If you have not enough memory to run the stock XFree86 distribution, you might try to get this package. It is based on XFree86, so the difficult parts of the configuration (setting up the `Xconfig` file) are identical. To

[4] For `csh` users this should read `startx >& Xstartup.log` .

run it you need at least 4 MB RAM and another 12 MB swap space. There are now several tinyX versions, one for each of the different servers supplied by XFree86. Please read the above description of the servers to get the right version of tinyX. These packages are named like

`tinyX-YYY-ver.tar.gz`, where YYY denotes the server name (without leading XF86_)

You can find tinyX on `sunsite` in the `/pub/Linux/X11` hierarchy. Please look in your neighbourhood for a Mirror of it [cf. 4.1]. You untar it as root from `/`. After this you have to setup a `Xconfig` file [cf. 5.1]. The documentation and READMEs for tinyX as well as most of the XFree86 documentation files can be found in `/readmes`. Read the XFree86 documentation files *before* setting up a `Xconfig` file as they are required for this.

The tinyX documentation includes a lot of useful information on memory saving techniques. You should follow at least some of these suggestions, as running X while swapping constantly is no fun at all :-(. If you are running out of memory your system will lock up completely (in many cases the only way out is to turn off power or to make a hard reset - both is very dangerous for your filesystem). So read these files before starting X so that this does not happen. You should read the sections on configuring your X11 system, too [cf. 5.2, 5.4, 5.5, and 5.6].

As tinyX does not include all documentation available with XFree86, you might notice that you are pointed to a file that is not included. In this case you must get the rest of the documentation files (and perhaps the manpages, too) from any XFree86 site [cf. 4.1]. If you have enough disk space for that, this is highly recommended anyway.

Note that XFree86 specific manpages (as the Server manpages,...) are not in the `*-man*` but in the `*-doc*` files.

7 X related packages

There are a lot of packages that are related to X. The first two are mentioned by name, as they have a great influence on the look-and-feel of X.

Xview

The `xview3L5.1.tar.gz` package is a port of SUN™'s xview3.2 package by Kenneth Osterberg to Linux. It includes the xview toolkit, a set of extensions to X. It will give you the look-and-feel of the Open Look™ extension to X (many will know that from Open Windows™ from SUN). You can get it in the `/pub/Linux/libs/X` directory on `sunsite` or in the `/pub/linux/binaries/usr.bin.X11` directory on `tsx-11` [cf 4.1].

To install it, untar the tarfile as root in some directory (e.g. `/usr/src`), read the documentation that comes with it (most important the README) and run the INSTALL script. Note that you need about 25MB free disk space to install it completely (with examples). This is partly due to the fact that for some time all files are twice on the disk. After running the INSTALL script and removing the xview3L5.1 directory you will need about 11 MB (including all examples). If you do not want to install the examples , you will need about 9 MB. If you do not want to compile any program that uses xview, you can spare another 2.5 MB by removing the static (`*.a`) and stub (`*.sa`) libraries. Do *not* remove the shared (`*.so`) libraries, though!

For running Xview you need at least 8 MB of RAM, better 16 MB.

Motif

This is commercial software, so you will have to pay for it. You can find an advertisement from Metrolink (a company that provides a port to Linux) on `tsx-11:/pub/linux/advertisements`. There are more companies that sell Motif. Every now and then advertisements will appear in `comp.os.linux.announce`.

Windowmanagers [cf. 5.5]

Both aforementioned packages provide window managers different than the standard twm that comes with XFree86. Many people do not like this window manager. There are several further windowmanagers available that run on Linux. A selection of these can be found on `sunsite` in the directory `/pub/Linux/X11/Window-managers`.

Additional libraries

Sometimes you will need additional libraries to run binary distributions of some application or to compile it yourself [cf. 8]. If the first is the case, you should find a pointer to that library in the Readme file of that package. If you do not have any pointer, you should look at the `/pub/Linux/libs` hierarchy at `sunsite`. Most libraries that work on Linux can be found there. If you cannot find it there and you have Internet access, you should follow the instructions on getting information below [cf. 9].

X11 Applications

There are many programs and applications available for X, far too many to give even examples here. Look through the `/pub/Linux/X11` hierarchy on `sunsite` for ports of many packages to Linux. Most programs need no changes at all to run on Linux, so you should look at the generic X11 program archives as well as at the Linux specific sites. One of the largest X11 sites is `ftp.x.org`. In the `R5contrib` directory you can find many programs for X11R5.

8 Compiling programs that use X

Please read the `GCC-FAQ`, that can be found in the `docs` directories of `sunsite` and `tsx-11` before compiling any programs. Many questions (in fact, most) on compiling programs with Linux are answered there. Many Linux distributions include most relevant FAQs in the directory `/usr/doc`, so you might look there first.

If you have the source code for a program that uses X11, usually it will be shipped with an Imakefile instead of a Makefile. What to do now?

Run `xmkmf` in the directory that containes the Imakefile. If there is a hierarchy of directories with Imakefiles, you usually only have to run `xmkmf` in the root directory of that hierachy. You can configure the rest by typing make Makefiles. This will build the Makefiles in all directories in the hierachy. Then you should run `make depend`. Don't be afraid if include files like `stddef.h`, `varargs.h`, ... are not found, they are `gcc` proprietary header files and therefore not in the standard include directories (XFree86 versions prior to 2.1 use a `makedepend` not based on `gcc -M`). After that you can make the program by running make and you can install it (usually in `/usr/X386/bin`) by running `make install`, installing of the manpages is done by running `make install.man`. Of course, before even running `xmkmf`, you should have read the documentation that comes usually with such packages.

If libraries are missing, please read section 7 above.

9 Finding Information on X and XFree86 on the net

For general X questions you should read the *FAQ* (`ftp://ftp.x.org/R5contrib/FAQ`) and the *FAQ-Xt* (`ftp://ftp.x.org/R5contrib/Xt-FAQ`) from the `comp.windows.x` newsgroup (you can achieve them from `ftp.x.org` in the directory `R5contrib` or from `rtfm.mit.edu` in `/pub/usenet_by_hierarchy/comp/windows/x` by anonymous ftp). Many Linux distributions include most relevant FAQs in the directory `/usr/doc`, so you might want to look there first.

Please note that `comp.windows.x` is the newsgroup for general X questions.

If you have questions on the XFree86 package and cannot find an answer in the documentation files (`XFree86, Xconfig, XF86_*` manpages or the `README` files in `/usr/X386/lib/X11/etc`) you may want to post a question to a newsgroup. The appropriate newsgroup for that purpose is `comp.windows.x.i386unix`, as this newsgroup is dedicated to the XFree86 system. If you have a problem with the server and want to post a question to this newsgroup, you should provide enough information for those that want to help you. It is no good idea telling: XFree86 is not working, what do I have to do? Here some things that should be included in all postings that are concerned with server problems:

- which Operating System and which release of it are you running
- which hardware do you have (at least bus type (ISA/EISA/VLB), graphic card (including chipset, video RAM type (D/VRAM), size and speed))
- a concise decription of the problem
- a printout of the server startup (you can generate that by running `startx 2>1 > Xerror.log` or `startx >& Xerror.log` depending on whether you use a bourne shell derivate (e.g. `bash`) or a C shell derivate (e.g. `tcsh`) as login shell)
- the *used* parts of the `Xconfig` (It is **not** a good idea to include all lines commented out by a leading #. This would be the largest part of your post and most people will already know the copyright statements :-)

Furthermore you should say what steps of trouble-shooting you already did (there are some mentioned in the XFree86 documentation files and in Appendix C).

Bug reports or questions on XFree86 can be sent directly to `XFree86@physics.su.oz.au`. This is the contact address of The XFree86 Project.

There are some other newsgroups that cover X related topics:

- `comp.windows.x.apps` - for X applications
- `comp.windows.x.intrinsics` - for X intrinsic

- `comp.windows.x.motif` - for the Motif extension to X
- `comp.windows.open-look` - for OpenLook and the Xview extension to X

Note that most of these groups have FAQ lists that are posted regularly to these groups. As always with regular postings these can be obtained from `rtfm.mit.edu` via anonymous ftp.

Please do not post X related questions to *any* newsgroup of the `comp.os.linux.*` hierarchy. You will get better answers to your questions in those newsgroups dedicated to those questions, as nearly all those questions are not Linux-specific.

10 Why should I upgrade my system and how do I do that?

Upgrading will always refer to the latest XFree86 distribution (currently 2.1.1). First of all, take a look at the bugs section [cf. A]. If you are caught by one of those bugs, you should upgrade. You should upgrade, if you want to use one of the newer features. You should upgrade in any case if you are not running at least XFree86 2.0 as the shared libraries of XFree86 version 2 and those of the earlier versions are partly incompatible. Some bugs have been fixed since then and most binary packages that use X11 are linked with the XFree86 2.? libraries.

Every subsection has a list of new features (if there are any) angainst the next newer version.

Furthermore you should upgrade if you have problems with your `Xconfig` file and want to use the new `ConfigXF86` utility.

10.1 Upgrading from XFree86 versions before 2.0

You should get the complete distribution but the fonts files (`*-fnt*`) which have not changed.

10.2 Upgrading from XFree86 2.0

You should get the complete libraries and programs file, as there are some bugfixes (most important several security holes in xterm). New features in 2.1 are:

- kernel keytable support.
- 8bit colour lookup table support for some RAMDACs (S3 &Mach32 servers).
- Support for SPEA Mercury, TI RAMDACs, additional line lengths, and (preliminary) Bt485 (S3 server).
- Better DRAM card support (S3 server).
- Pixmux code for some RAMDACs (Mach32 server).

10.3 Upgrading from XFree86 2.1

You only need to upgrade if you have problems due to one of the bugs. If you start using XFree86 you should get 2.1.1, as many bugs have been fixed and there is a new configuration utility that will make installation much easier. New features in 2.1.1:

- `ConfigXF86`
- full Bt485 support (S3 server).

A Bugs in XFree86

A.1 XFree86 2.0

The know bugs in XFree86 2.0 are:

- No Bt485 pixmux support (S3 server) - fixed in 2.1.1.
- Font restauration and text mode restauration are not working correctly on some S3 cards (S3 server) - fixed in 2.1.1.
- No line length of 1152 supported (S3 server) - fixed in 2.1.
- Linear Frame buffer code buggy (S3 server) - fixed in 2.1.
- LinkKit bug (S3 & VGA16 server) - fixed in 2.1.
- The Compaq AVGA driver is not working correctly - fixed in 2.1.
- The clock probing code is not working for WD90C3? cards - fixed in 2.1.
- The ATI SVGA server is not working correctly - fixed in 2.1.1.
- The Mach32 server has problems to detect some more recent revisions of the mach32 chip.
- There are some bugs in the accelerated Cirrus server.
- There is a problem with `xset fp rehash` due to the memory management - supposedly fixed in 2.1.1.

A.2 XFree86 2.1

The know bugs in XFree86 2.1 are:

- The Bt485 pixmux support (S3 server) is only working for some cards (SPEA Mercury). It is not working on most other cards (#9, STB Pegasus,...) - fixed in 2.1.1 for most cards.
- Font restauration and text mode restauration still not working correctly on some S3 cards - fixed in 2.1.1.

- The ATI SVGA server does not work correctly - fixed in 2.1.1.
- The Mach32 server has problems to detect some more recent revisions of the mach32 chip.
- Less bugs in the Cirrus drivers, but there still are some.
- Still problem with the Compaq driver - fixed in 2.1.1.
- still `xset fp rehash` problems - fixed in 2.1.1.

A.3 XFree86 2.1.1

- Some not yet resolved problems with the Mach servers.

B The Future of XFree86

At the moment the XFree86 team is working on X11 release 6. The next version, XFree86 3.0 is based on X11R6 and is incorporated in the official X11 distribution. This release is not yet useable on Linux (at least parts of it). Please wait until XFree86 3.1 is released which will provide all features of both X11R6 and the current XFree86 version (2.1.1).

To get version 3.1 working is first priority for the XFree86 team. After this is done, new features and new hardware will be worked upon.

C Some Questions and Answers

Q: I noticed that X11 Release 6 has been announced. Can I use it on Linux?
A: Please look at appendix B above. You should not use it at the moment.

Q: I started X by running '`X`'. I get the stippled background, but no window at all. What am I doing wrong?
A: Running X starts only the server, no window-manager and no applications. The usual way to start X is running '`startx`'.

Q: Is there support for 16-bit or 24-bit colour modes?
A: No, there is only support for 1-bit, 4-bit and 8-bit colour in the moment. The 4-bit server (`XF86_VGA16`) is in an experimental stage, so there are bugs and problems using it. Most of the known problems are mentioned in the manpage.

Q: I have a Logitech serial mouse but it does not work with the `Logitech` keyword.
A: The `Logitech` keyword is for older Logitech serial mice, more recent Logitech serial mice use the `MouseMan` or `Microsoft` keyword. Logitech serial mice plugged into a ps/2 port use the `ps/2` keyword.

Q: My server is unable to find some of the fonts (e.g. font `fixed` cannot be loaded at startup).
A: First check that the directories in the font path in the `Xconfig` file are named correctly and contain fonts. If that is the case, running mkfontdir in all of those directories may help you.

Q: I have the problem that the fonts are not restored correctly after leaving X. My screen looks totally garbled.
A: This is not yet completely resolved. There are utilities called runx , restoretext,... coming with svgalib (`sunsite:/pub/Linux/libs`) that help in most cases. In the S3 server from XFree86 2.1.1 this bug is fixed.

Q: I have a s3 card and it does not work properly. The screen gets distorted when dragging windows and it frequently locks up.
A: First try to use options `nomemaccess` or `nolinear`. If both does not help and you have a s3 928, try specifying chipset s3_generic in your `Xconfig`. If nothing helps, it might be a incompatibility problem between your graphic card and your motherboard. Try to see if the card works in a different system, if you can get your hands on one.

Q: I had a normal SVGA card up to now with 1 MB of video RAM. With it I could do a virtual resolution of 1152x900 or 1024x1024. Why does it not work on my new s3 card?
A: All but the most recent revisions of the s3 chips cannot use a line length of 1152, so a length of 1280 is used for this resolution in XFree86 2.0. The later versions can use the line length of 1152 for the more recent revisions. If you have a new revision, grade up to 2.1.1 and it should work. Furthermore the s3 server needs 1k for the cursor, so a virtual resolution using 1MB will not work. *Note* that you will need some memory on the graphics card for font and pixmap caching if you want to get maximum performance from your graphic card. So the memory is *not* wasted, but will get you a great performance boost. Try running some font or pixmap specific mode of x11perf with maximal allowed virtual resolution (e.g. 1024x1023) and with 1024x768 and compare the results! Recommended resolutions are 1024x768 and 1152x800 for 1MB cards. With the older chip revisions the latter resolution leaves not too much memory to the font cache, so you will have a performance loss using large fonts. For normal use with fonts up to 18 points it should be OK, though.

Q: I have a s3 Localbus card and the server cannot find a high memory area. Then it disables direct memory access. What should i do?
A: There are some strange problems with Localbus cards. It is being worked upon. If you are using XFree86 2.0, put `option "nolinear"` *and* `option "memaccess"` into your `Xconfig` file. This will enable the server to use the 64k aperture that is supported even for ISA cards. This will furthermore prevent the following problem (this bug is only in XFree86 2.0): Due to a bug in the linear frame buffer code, the s3 processor is not reset correctly, once a large (e.g. 1024k) linear frame buffer is found. Thus the server will not work after that if it fails to find the high memory area until a hard reset is sent to the chip (i.e. even a normal reboot will not reset the system).

Q: I have a WD90C3x based graphic card. XFree86 2.0 seems not to probe the clocks correctly. What can I do?
A: Yup, there is a bug in the clock probing code in XFree86-2.0. It is fixed in 2.1. Grade up!

Q: I have a Diamond Viper card. Why does it not work?
A: In addition to being a Diamond product, this card is based on the P9000 chip by Weitek. This chip is not yet supported by XFree86, so you have only a chance to get it working with one of the generic servers (i.e. Mono or VGA16). You will have nevertheless the clock setting problem that prevents the other diamond cards from working.

Q: I have a card with the Compaq AVGA chip. It does not work.
A: The AVGA support seems to be completely broken in 2.0. This is fixed in 2.1.

Q: I have a busmouse and am running selection. The server has problems with recognizing the mouse.
A: Selection has some problems with busmice. XFree86 cannot open the mouse device if selection is running on it. So kill selection before starting XFree86.

Q: The server dies with the message "Cannot find a free VT".
A: XFree86 needs a free VT to run on. So if you have put a getty process on every virtual console in you `/etc/inittab`, XFree86 will not be able to start. I suggest to let `/dev/tty8` (for kernel messages) and `/dev/tty7` (for XFree86) free of a getty process.

Q: Why is the `XF86_Mono` server not working with my hercules card, I thought this card was supported?
A: The hercules driver (as well as the BDM2 driver) is not part of `XF86_Mono` by default. Get the LinkKit and you can include it. You will need about 4 MB free disk space and about 5 minutes for building the server (on a 486/30).

Q: I have problems with the backspace and delete keys.
A: You can solve this using either loadkeys or xmodmap. In your shell initialization file (`.bashrc`, `.zshrc`, ...) you have to set the right behaviour of the shell using stty. Please read the manpages how to do that.

There are a lot strange phenomena reported that cannot be handeled here. If your card has a switch for bus waitstates, it helps in many cases to put this to the 1WS setting. You should also look at the remarks above on setting some BIOS values. As many reports (especially on BIOS setting) are contradicting, you should try several settings.

D Acknowledgements

Dirk Hohndel (`hohndel@aib.com`) and David E. Wexelblat (`dwex@aib.com`) helped writing this HOWTO.

E Trademarks