# Information for SVR4 Users

The XFree86 Project, Inc 14 December 1994

**NOTE:** If you intend to use any of the accelerated servers, read section 10 and follow the instructions. Otherwise the X server will crash when exiting, restarting, or switching VTs.

## Contents



## 1 SVR4 versions on which XFree86 has been tested

XFree86 has been tested on the following versions of **SVR4.0**:

- Microport: 2.2, 3.1, 4.1, 4.2
- Esix: 4.0.3A, 4.0.4, 4.0.4.1
- Dell: 2.1, 2.2, 2.2.1
- UHC: 2.0, 3.6
- Consensys: 1.2

- MST: 4.0.3
- AT&T: 2.1, 4.0
- ISC: 4.0.3
- NCR: MP-RAS

and the following versions of **SVR4.2**:

- Consensys
- Novell UnixWare

Basically, we believe that XFree86 binaries will run unmodified on any ISA, EISA, or MCA platform version version of SVR4.0 (Solaris 2.x is an exception), or SVR4.2. If you run XFree86 on another version of SVR4 that's not in this list, please let us know about it.

# 2 How to cope with VT-switching hotkeys

Some versions of SVR4 (Esix and Microport) have mechanisms for enabling two-key sequences for VT switching (`Alt-Fn`). The standard SVR4 mechanism is `Alt-SysReq-Fn`, which all versions we know use. Running under X, the `Alt-Fn` sequences are stolen by the driver before the server can see them, so you can't use them for X applications. So you want to switch back to the standard 3-key sequences while you are running X. Here's how to do it:

**Microport**

Microport makes this very simple. The 2-key mode is called "Microport Mode", and the 3-key mode is called "Compatible Mode". You enter Microport Mode by pressing `Alt-SysReq-m`. You enter Compatible Mode by pressing `Alt-SysReq-c`. So all you need to do is press `Alt-SysReq-c` after to allow X clients access to the `Alt-Fn` sequences.

**Esix**

Esix has no keyboard-driven way to switch modes. There are two levels at which this can be handled:

1. There is a kernel tunable that determines which mode is the default. The tunable is the initialisation of kd_2keysw in `/etc/conf/pack.d/kd/space.c`. When set to 1 (the default), 2-key mode is enabled. When set to 0 it is disabled.
2. The mode can be changed for individual VTs programatically by an ioctl(). To make life easier for XFree86 users, a program called '2key' is provided (in `xc/programs/Xserver/hw/xfree86/etc/` in the source tree, and in `/usr/X11R6/lib/X11/etc/` in the binary kit). You can compile and install this program. Then to make use of it, add the line '`VTInit "2key off"`' to the Keyboard section of your `XF86Config` file to cause the program to be run automatically when the server starts up. Doing this means that 2-key switching will be turned off while in the server's VT, but will still be on for the other VTs.

For further details, refer to the keyboard(7) man page included with the release notes (the on-line man page doesn't have this information).

# 3 Running SVR3 binaries on SVR4.0.4 and SVR4.2

SVR4.0.4 added the 'Advanced Compatibility Package', which provides iBCS-2 compliance for running SVR3 binaries. These facilities are also present in SVR4.2. XFree86 makes use of this to accept local connections from SVR3 clients. The XFree86 binary distribution is built to use these capabilities. You need to install the 'Advanced Compatibility Package', if you have not done so already.

We have found that SVR4.0.4 is not able to run all SCO, and perhaps not many ISC SVR3 binaries. This is not a failing of XFree86, but of SVR4 itself. One particular example is that many SVR3 programs are ignorant of the UFS filesystem, and attempt to read directories as files, rather than using the system call that is defined for the purpose. This will fail for obvious reasons. The SVR4.0.4 release notes from USL (which you should have gotten from your vendor) have lots of suggestions for how to improve compatibility.

That said, we have had luck with several SCO binaries right out of the box. No changes are needed - just go to an xterm window and run the program.

ISC users will need a binary editor before they can attempt to run their binaries. ISC, for whatever reason, put the pipe for local connections in `/tmp/.X11-unix/Xn`. This unfortunately is the same place as the X Consortium X server puts the Unix-domain socket normally used for local connections. The XFree86 server was modified to use `/dev/X/ISCCONN/Xn` for local connections to ISC clients. So what you must do is use a binary editor to edit your client program. Search for `/tmp/.X11-unix`, and change it to `/dev/X/ISCCONN`. Now you just have to worry about base-OS compatibility.

# 4 Notes for building XFree86 on SVR4

1. If you are using gcc with SVR4, we highly recommend that you use gcc-2.4.5 (or a later stable release). Version 2.6.0 has some problems on i386 platforms and is not recommended.

2. It is recommended that you increase the `UFSNINODE` (for a UFS filesystem) and/or the `S5NINODE` (for an S5 filesystem) kernel parameter to about 650 before attempting to build the distribution. See the "Notes for running XFree86 on SVR4" section for some other parameters you may want to change.

3. The `BOOTSTRAPCFLAGS` required are:

   For Unixware: "`-DUSL`" For NCR: "`-DNCR`"
   For other SVR4: "`-DSVR4 -Di386`"

# 5 Notes for running XFree86 on SVR4

**NOTE:** If you intend to use any of the accelerated servers, read section 10 and follow the instructions. Otherwise the X server will crash when exiting, restarting, or switching VTs.

1. For SVR4, you may also need to add `/usr/X11R6/lib` to your `LD_LIBRARY_PATH`, but this is not required for running properly built clients.

2. You may want to increase some kernel parameters (either by running `idtune`, or editing `/etc/conf/cf.d/stune`, and rebuilding the kernel with `idbuild`):

   `[HS]FNOLIM` hard/soft limit for number of open files`MAXUP` max number of processes per user

   `ARG_MAX` max length of an arg list

3. Choose which mouse driver you will use. For SVR4 the best choice depends on which version you are using. If you have a bus mouse then Xqueue is probably the only option. For a serial mouse the options are as follows:

   **Esix 4.0.3**

   Xqueue works. It is also possible to use the standard asy driver directly, but the mouse operation is "jerky".

   **Microport SVR4** 34.1

   Xqueue works fine, and the asy driver can also be used directly giving smooth mouse operation.

   To use Xqueue, set the `Protocol` to `Xqueue` in both the `Keyboard` and `Pointer` sections of your `XF86Config` file, and You must have the mouse driver package installed, and must run mouseadmin to set it up for your mouse. If mouseadmin won't work try doing '`touch /dev/gmse`' before running it. (Note that mouseadmin will need to be rerun after rebuilding a kernel unless you add an appropriate entry to `/etc/conf/node.d/gmse`.)

   If you have problems with both Xqueue and your standard asy driver with SVR4, then you should install SAS. When using SAS, set up `XF86Config` as you would for the standard driver.

   SAS is available from ftp.physics.su.oz.au. When using SAS for a serial mouse, you will get smoother operation if you change `EVENT_TIME` from 80 to 30 in `sas.h`. A couple of details which aren't spelled out in the SAS README are:

   - An example of the line you should add to `/etc/ap/chan.ap` is:

   ```
   MAJOR    0        255     ldterm ttcompat
   ```

   where `MAJOR` is replaced by the major number used for SAS devices. To determine what that is, check `/etc/conf/cf.d/mdevice` after rebuilding the kernel. The major number is the sixth field in the line starting with 'sas'. This file must be updated before rebooting with the new kernel.

   - The installation instructions omit the following:

   3a) Disable the asy driver by either running '`kconfig`' or editing `/etc/conf/sdevice.d/asy`.3b) Rebuild the kernel by running `/etc/conf/bin/idbuild`

4. If you want to use xdm with SVR4, extract the files from the shar file in `/usr/X11R6/lib/X11/etc/XdmConf.svr4` into a temporary directory.

The `README` file tells where the individual files should be installed. Be sure to read through each file and make any site-specific changes that you need.

**NOTE:** Some SVR4 versions (one example is Esix 4.0.3) have a default inittab which runs 'vtgetty' on the console. This does not work well when starting xdm at boot time. The problem is that when you logout from a vtgetty session it wants to close all the VTs – including the one xdm is using for the server. It is recommended that you use 'getty'. If you change `/etc/inittab`, remember to also change `/etc/conf/cf.d/init.base` or you will lose the changes when you next rebuild the kernel.

5. If you want to change the number of VTs available on SVR4, just edit the file `/etc/default/workstations` and change the number there. The device nodes will be created/deleted next time you reboot.

6. The default local connection types have changed in X11R6. Unix domain sockets are no longer treated as a "local" connection type. This means that a client connecting to :0 will use not use a Unix socket for the connection. To use the Unix socket connection, the client must connect to unix:0.The local connection types available are "`NAMED`" (named streams pipe), "`PTS`" (old-stype USL streams pipe), "`SCO`" (SCO Xsight streams pipe), and "`ISC`" (ISC streams pipe). The `XLOCAL` environment variable can be used to set which types of local connection should be used in order of preference. The default setting is `PTS:NAMED:ISC:SCO`. It is recommended that `NAMED` be used in most cases because it is faster than the default `PTS`, and because using PTS can cause you to run out of `/dev/pts/` devices (each client using PTS requires a `/dev/pts` device). To set up the default local connection type, make sure that `XLOCAL` is set and exported in your `.xinitrc` file (when using xinit or startx) or your `/usr/X11R6/lib/xdm/Xsession` script (when using xdm).

   The X server doesn't currently remove the links it makes to its PTS listener when it exits. This can lead to problems when starting a new server if something else is now using that PTS device. This is usually noticed by delays in clients making connections to the server. The only solution is to make sure that an old link in `/dev/X/server.N` is removed before starting a server for display N.

# 6 Notes for running XFree86 on SVR4.2

In addition to the notes for SVR4.0, you need to be aware of a few problems with SVR4.2. Basically, the base SVR4.2 code has broken Unix-domain sockets in such a way that making local connections via `UNIXCONN` does not work properly (this bug is known to exist on Consensys SVR4.2 and Novell UnixWare). The manifestation of this bug is that windows remain on the screen after the client program exits, until you move the mouse into the window, or otherwise cause the server to try to write to the client.

If you run XFree86 and see the manifestation of the Unix-domain socket bug described above, you can work around this problem quickly and effectively by changing the default local connection mode to `NAMED` rather than `UNIX`. The mechanisms for doing this are described above. This is not a problem for clients using the X11R6 X libraries because `UNIX` is no longer considered a local connection type.

# 7 Building non-core clients with SVR4

1. A lot of clients (even some which have explicit SVR4 support) require `-DSYSV` when building under SVR4. This will not be set when using the default configuration. A quick fix is to add something like the following to the client's Imakefile:

   ```
   #if SystemV4
        DEFINES = -DSYSV OTHER_CLIENT_DEPENDENT_DEFINES
   #endif
   ```

   The best solution is to modify the code so it compiles correctly without `-DSYSV`.

2. The default compiler options include '`-ansi`' for gcc, and '`-Xc`' for cc. A consequence of this is `__STDC__` gets `#defined` to '1'. There are a number of functions which will not have prototypes declared unless either `__STDC__` is not defined, or

   ```
   __STDC__ == 0 || defined(_POSIX_SOURCE) || defined(_XOPEN_SOURCE)
   ```

   Possible solutions are to change the definition of `CCOPTIONS` by adding a line to the Imakefile, or to add the required prototypes to the source.

# 8 Using DOS/Merge 2.2 with XFree86

It is possible to use the Locus DOS/Merge 2.2 X clients with XFree86. You need to do a couple of things for this to work, though. One change is a generic problem with the X client and X11R5/6; the others are to work with some things that are specific to the XFree86 servers. Here are the things you need to do:

1. Set and export `$XMERGE` in your `.xinitrc` and/or `.xsession` files. In general, you should set `XMERGE=vga`.

2. You MUST use the "xqueue" driver instead of the server's native keyboard and mouse driver, if you intend to use the "zoom" feature of the 'dos' client. Otherwise the mouse will cease to function after the first time you "zoom" (because the 'dos' client uses the native driver, and the server will not be able to access the mouse after the zoom ends). The only other alternative is to use separate mice on separate devices.

3. You need to install the 'dos' client fonts in the XFree86 font directories. Locate the BDF files (search for files with names matching the pattern '*pc???.bdf'). These will likely be `/usr/lib/X11/fonts/misc`. Go to the directory where these files are located, and execute the following (using 'sh' or 'ksh'):

   ```
   for i in *pc???.bdf
   do
       /usr/X11R6/bin/bdftopcf $i > \
                   /usr/X11R6/lib/X11/fonts/misc/`basename $i .bdf`.pcf
   done
   ```

```
cd /usr/X11R6/lib/X11/fonts/misc
/usr/X11R6/bin/mkfontdir
# Do this only if the server is already running.
/usr/X11R6/bin/xset fp rehash
```

4. The 'dos' client program uses a translation table to map from an internal key representation to the X keymap. It is likely that the table supplied with Merge 2.2 use the mapping for SCO's server. A correct mapping table is available in `/usr/X11R6/lib/X11/etc/xcode.xfree86`. This file should be installed in `/usr/lib/merge/xc`. In addition, you must add the following resource to the 'dos' client's application-defaults file (usually in `/usr/lib/X11/app-defaults/DOS`):

   ```
   dos*xcodetable: /usr/lib/merge/xc/xcode.xfree86
   ```

   It will be obvious if this new code table is needed, as the arrow keys on the keypad will fail to function in the 'dos' client if the wrong table is installed.

5. For the "zoom" feature to work correctly, you must run 'dos' with $DISPLAY set to "unix:N" or "host_name:N". If you use just ":0", the client will not function properly. 'dos' does not accept a '-display' parameter. Hence it is probably a good idea to replace the 'dos' program with something like this:

   ```
   #!/usr/bin/ksh
   if [ "X${DISPLAY}" != "X" ]
   then
           case ${DISPLAY} in
           :*)
                   DISPLAY=unix${DISPLAY}
                   ;;
           esac
   fi
   /usr/bin/dos.real "$@"
   ```

# 9 Keyboard mapping problems with some Esix systems

One of the console driver patches for Esix 4.0.3A causes the XFree86 server's default keymap to be corrupted. If you are being affected by this problem it will be obvious because few (if any) of the keys will be mapped correctly. There are two solutions to this. One is to remove the console driver patch which introduced the problem. The second is to use xmodmap(1) to reset the default mapping after server startup. The default mapping is provided in the file `/usr/X11R6/lib/X11/etc/xmodmap.std`, and can be installed automatically by adding the line:

```
xmodmap /usr/X11R6/lib/X11/etc/xmodmap.std
```

to your `.xinitrc` file (or your `Xsetup` file if using xdm).

# 10 A kernel patch that is required for accelerated servers

SVR4.0 has a bug handling programs that access extended I/O registers (above 0x3FF). Boards like S3 and 8514/A use these extended I/O registers. XFree86 now supports boards that tickle this bug. In preparation for using these servers, we have produced a kernel patch that works around the problem, and provided scripts for you that will both install and back out the patch. You must install this if you intend to use the S3, 8514, Mach8, Mach32, P9000, AGX or W32 servers.

Dell 2.2 is known to not need the patch, because Thomas Roell found and fixed the bug while he was working for Dell. Microport has fixed this in their 4.0 v4.2 release. Also, SVR4.2 does not need this patch, as the problem has been fixed by USL.

The patch scripts are located in `xc/programs/Xserver/hw/xfree86/etc` in the source tree, and `/usr/X11R6/lib/X11/etc` in the binary distribution. The files are 'svr4_patch' to install the patch, and 'svr4_patch_rem' to back it out. The file that is being patched is `/etc/conf/pack.d/kernel/os.o`. The patch script verifies the presence of the bug before patching, and will tell you whether or not it succeeded in patching. You need to run the 'svr4_patch' script as root, obviously. The original `os.o` file, as well as the patching program, and a copy of the removal script are stored in the directory `/etc/conf/pack.d/kernel/.xfree86`

Thanks to John M. Sully of Microport for helping us find a simple workaround for this problem, and giving us permission to release the information.

```
$XConsortium: SVR4.sgml,v 1.2 95/01/16 13:17:49 kaleb Exp $
$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/SVR4.sgml,v 3.3 1995/01/28 16:02:30 dawes Exp $
```