NAME

xorg.conf, xorg.conf.d − configuration files for Xorg X server

INTRODUCTION

Xorg supports several mechanisms for supplying/obtaining configuration and run-time parameters: command line options, environment variables, the xorg.conf and xorg.conf.d configuration files, auto-detection, and fallback defaults. When the same information is supplied in more than one way, the highest precedence mechanism is used. The list of mechanisms is ordered from highest precedence to lowest. Note that not all parameters can be supplied via all methods. The available command line options and environment variables (and some defaults) are described in the Xserver(1) and Xorg(1) manual pages. Most configuration file parameters, with their defaults, are described below. Driver and module specific configuration parameters are described in the relevant driver or module manual page.

DESCRIPTION

Xorg uses a configuration file called *xorg.conf* and files ending in the suffix *.conf* from the directory *xorg.conf.d* for its initial setup. The *xorg.conf* configuration file is searched for in the following places when the server is started as a normal user:

/etc/X11/<cmdline>
/usr/etc/X11/<cmdline>
*/etc/X11/***$XORGCONFIG**
*/usr/etc/X11/***$XORGCONFIG**
/etc/X11/xorg.conf
/etc/xorg.conf
/usr/etc/X11/xorg.conf.<hostname>
/usr/etc/X11/xorg.conf
/usr/lib/X11/xorg.conf.<hostname>
/usr/lib/X11/xorg.conf

where <*cmdline*> is a relative path (with no ".." components) specified with the **−config** command line option, **$XORGCONFIG** is the relative path (with no ".." components) specified by that environment variable, and <*hostname*> is the machine's hostname as reported by **gethostname**(3).

When the Xorg server is started by the "root" user, the config file search locations are as follows:

<cmdline>
/etc/X11/<cmdline>
/usr/etc/X11/<cmdline>
$XORGCONFIG
*/etc/X11/***$XORGCONFIG**
*/usr/etc/X11/***$XORGCONFIG**
/etc/X11/xorg.conf
/etc/xorg.conf
/usr/etc/X11/xorg.conf.<hostname>
/usr/etc/X11/xorg.conf
/usr/lib/X11/xorg.conf.<hostname>
/usr/lib/X11/xorg.conf

where <*cmdline*> is the path specified with the **−config** command line option (which may be absolute or relative), **$XORGCONFIG** is the path specified by that environment variable (absolute or relative), **$HOME** is the path specified by that environment variable (usually the home directory), and <*hostname*> is the machine's hostname as reported by **gethostname**(3).

Additional configuration files are searched for in the following directories when the server is started as a normal user:

/etc/X11/<cmdline>
/etc/X11/<cmdline>
/etc/X11/xorg.conf.d
/etc/X11/xorg.conf.d

where <*cmdline*> is a relative path (with no ".." components) specified with the **−configdir** command line option.

When the Xorg server is started by the "root" user, the config directory search locations are as follows:

<cmdline>
/etc/X11/<cmdline>
/etc/X11/<cmdline>
/etc/X11/xorg.conf.d
/etc/X11/xorg.conf.d

where <*cmdline*> is the path specified with the **−configdir** command line option (which may be absolute or relative).

Finally, configuration files will also be searched for in directories reserved for system use. These are to separate configuration files from the vendor or 3rd party packages from those of local administration. These files are found in the following directories:

/usr/share/X11/xorg.conf.d
/usr/share/X11/xorg.conf.d

The *xorg.conf* and *xorg.conf.d* files are composed of a number of sections which may be present in any order, or omitted to use default configuration values. Each section has the form:

Section "*SectionName*"
  SectionEntry
  ...
EndSection

The section names are:

Files File pathnames
ServerFlags Server flags
Module Dynamic module loading
Extensions Extension enabling
InputDevice Input device description
InputClass Input class description
Device Graphics device description
VideoAdaptor Xv video adaptor description
Monitor Monitor description
Modes Video modes descriptions
Screen Screen configuration
ServerLayout Overall layout
DRI DRI−specific configuration
Vendor Vendor−specific configuration

The following obsolete section names are still recognised for compatibility purposes. In new config files, the **InputDevice** section should be used instead.

Keyboard Keyboard configuration
Pointer Pointer/mouse configuration

The old **XInput** section is no longer recognised.

The **ServerLayout** sections are at the highest level. They bind together the input and output devices that will be used in a session. The input devices are described in the **InputDevice** sections. Output devices usually consist of multiple independent components (e.g., a graphics board and a monitor). These multiple components are bound together in the **Screen** sections, and it is these that are referenced by the **ServerLayout** section. Each **Screen** section binds together a graphics board and a monitor. The graphics boards are described in the **Device** sections, and the monitors are described in the **Monitor** sections.

Config file keywords are case−insensitive, and "_" characters are ignored. Most strings (including **Option** names) are also case-insensitive, and insensitive to white space and "_" characters.

Each config file entry usually takes up a single line in the file. They consist of a keyword, which is possibly followed by one or more arguments, with the number and types of the arguments depending on the keyword. The argument types are:

Integer an integer number in decimal, hex or octal
Real a floating point number
String a string enclosed in double quote marks (")

Note: hex integer values must be prefixed with “0x”, and octal values with “0”.

A special keyword called **Option** may be used to provide free−form data to various components of the server. The **Option** keyword takes either one or two string arguments. The first is the option name, and the optional second argument is the option value. Some commonly used option value types include:

Integer an integer number in decimal, hex or octal
Real a floating point number
String a sequence of characters
Boolean a boolean value (see below)
Frequency a frequency value (see below)

Note that *all* **Option** values, not just strings, must be enclosed in quotes.

Boolean options may optionally have a value specified. When no value is specified, the option’s value is **TRUE**. The following boolean option values are recognised as **TRUE**:

1, **on**, **true**, **yes**

and the following boolean option values are recognised as **FALSE**:

0, **off**, **false**, **no**

If an option name is prefixed with "**No**", then the option value is negated.

Example: the following option entries are equivalent:

```
Option "Accel"   "Off"
Option "NoAccel"
Option "NoAccel" "On"
Option "Accel"   "false"
Option "Accel"   "no"
```

Frequency option values consist of a real number that is optionally followed by one of the following frequency units:

Hz, **k**, **kHz**, **M**, **MHz**

When the unit name is omitted, the correct units will be determined from the value and the expectations of the appropriate range of the value. It is recommended that the units always be specified when using frequency option values to avoid any errors in determining the value.

FILES SECTION

The **Files** section is used to specify some path names required by the server. Some of these paths can also be set from the command line (see **Xserver**(1) and **Xorg**(1)). The command line settings override the values specified in the config file. The **Files** section is optional, as are all of the entries that may appear in it.

The entries that can appear in this section are:

FontPath "*path*"

sets the search path for fonts. This path is a comma separated list of font path elements which the Xorg server searches for font databases. Multiple **FontPath** entries may be specified, and they will be concatenated to build up the fontpath used by the server. Font path elements can be absolute directory paths, catalogue directories or a font server identifier. The formats of the later two are explained below:

Catalogue directories:

Catalogue directories can be specified using the prefix **catalogue:** before the directory name. The directory can then be populated with symlinks pointing to the real font directories, using the following syntax in the symlink name:

<*identifier*>:*[attribute]:*pri=<*priority*>

where <*identifier*> is an alphanumeric identifier, *[attribute]* is an attribute which will be passed to the underlying FPE and <*priority*> is a number used to order the fontfile FPEs. Examples:

75dpi:unscaled:pri=20 -> /usr/share/X11/fonts/75dpi
gscript:pri=60 -> /usr/share/fonts/default/ghostscript
misc:unscaled:pri=10 −> /usr/share/X11/fonts/misc

Font server identifiers:

Font server identifiers have the form:

<*trans*>/<*hostname*>:<*port−number*>

where <*trans*> is the transport type to use to connect to the font server (e.g., **unix** for UNIX−domain sockets or **tcp** for a TCP/IP connection), <*hostname*> is the hostname of the machine running the font server, and <*port−number*> is the port number that the font server is listening on (usually 7100).

When this entry is not specified in the config file, the server falls back to the compiled−in default font path, which contains the following font path elements (which can be set inside a catalogue directory):

/usr/share/fonts/X11/misc/
/usr/share/fonts/X11/TTF/
/usr/share/fonts/X11/OTF/
/usr/share/fonts/X11/Type1/
/usr/share/fonts/X11/100dpi/
/usr/share/fonts/X11/75dpi/

Font path elements that are found to be invalid are removed from the font path when the server starts up.

ModulePath "*path***"**
sets the search path for loadable Xorg server modules. This path is a comma separated list of directories which the Xorg server searches for loadable modules loading in the order specified. Multiple **ModulePath** entries may be specified, and they will be concatenated to build the module search path used by the server. The default module path is

/usr/lib64/xorg/modules

XkbDir "*path***"**
sets the base directory for keyboard layout files. The **−xkbdir** command line option can be used to override this. The default directory is

/usr/share/X11/xkb

SERVERFLAGS SECTION

In addition to options specific to this section (described below), the **ServerFlags** section is used to specify some global Xorg server options. All of the entries in this section are **Options**, although for compatibility purposes some of the old style entries are still recognised. Those old style entries are not documented here, and using them is discouraged. The **ServerFlags** section is optional, as are the entries that may be specified in it.

Options specified in this section (with the exception of the **"DefaultServerLayout" Option**) may be overridden by **Options** specified in the active **ServerLayout** section. Options with command line equivalents are overridden when their command line equivalent is used. The options recognised by this section are:

Option "DefaultServerLayout" "*layout−id*"
This specifies the default **ServerLayout** section to use in the absence of the **−layout** command line option.

Option "NoTrapSignals" "*boolean*"
This prevents the Xorg server from trapping a range of unexpected fatal signals and exiting cleanly. Instead, the Xorg server will die and drop core where the fault occurred. The default behaviour is for the Xorg server to exit cleanly, but still drop a core file. In general you never want to use this option unless you are debugging an Xorg server problem and know how to deal with the consequences.

Option "UseSIGIO" "*boolean*"
This controls whether the Xorg server requests that events from input devices be reported via a SIGIO signal handler (also known as SIGPOLL on some platforms), or only reported via the standard select(3) loop. The default behaviour is platform specific. In general you do not want to use this option unless you are debugging the Xorg server, or working around a specific bug until it is fixed, and understand the consequences.

Option "DontVTSwitch" "*boolean*"
This disallows the use of the **Ctrl+Alt+F***n* sequence (where F*n* refers to one of the numbered function keys). That sequence is normally used to switch to another "virtual terminal" on operating systems that have this feature. When this option is enabled, that key sequence has no special meaning and is passed to clients. Default: off.

Option "DontZap" "*boolean*"
This disallows the use of the **Terminate_Server** XKB action (usually on Ctrl+Alt+Backspace, depending on XKB options). This action is normally used to terminate the Xorg server. When this option is enabled, the action has no effect. Default: off.

Option "DontZoom" "*boolean*"
This disallows the use of the **Ctrl+Alt+Keypad−Plus** and **Ctrl+Alt+Keypad−Minus** sequences. These sequences allows you to switch between video modes. When this option is enabled, those key sequences have no special meaning and are passed to clients. Default: off.

Option "DisableVidModeExtension" "*boolean*"
This disables the parts of the VidMode extension used by the xvidtune client that can be used to change the video modes. Default: the VidMode extension is enabled.

Option "AllowNonLocalXvidtune" "*boolean*"
This allows the xvidtune client (and other clients that use the VidMode extension) to connect from another host. Default: off.

Option "AllowMouseOpenFail" "*boolean*"
This tells the mousedrv(4) and vmmouse(4) drivers to not report failure if the mouse device can't be opened/initialised. It has no effect on the evdev(4) or other drivers. Default: false.

Option "BlankTime" "*time*"
sets the inactivity timeout for the **blank** phase of the screensaver. *time* is in minutes. This is equivalent to the Xorg server's **−s** flag, and the value can be changed at run−time with **xset(1).** Default: 10 minutes.

Option "StandbyTime" "*time*"
sets the inactivity timeout for the **standby** phase of DPMS mode. *time* is in minutes, and the value can be changed at run−time with **xset(1).** Default: 10 minutes. This is only suitable for VESA DPMS compatible monitors, and may not be supported by all video drivers. It is only enabled for screens that have the "**DPMS**" option set (see the MONITOR section below).

Option "SuspendTime" "*time*"
sets the inactivity timeout for the **suspend** phase of DPMS mode. *time* is in minutes, and the value can be changed at run−time with **xset(1).** Default: 10 minutes. This is only suitable for VESA DPMS compatible monitors, and may not be supported by all video drivers. It is only enabled for

screens that have the **"DPMS"** option set (see the MONITOR section below).

Option "OffTime" "*time***"**
sets the inactivity timeout for the **off** phase of DPMS mode. *time* is in minutes, and the value can be changed at run−time with **xset(1).** Default: 10 minutes. This is only suitable for VESA DPMS compatible monitors, and may not be supported by all video drivers. It is only enabled for screens that have the **"DPMS"** option set (see the MONITOR section below).

Option "Pixmap" "*bpp***"**
This sets the pixmap format to use for depth 24. Allowed values for *bpp* are 24 and 32. Default: 32 unless driver constraints don't allow this (which is rare). Note: some clients don't behave well when this value is set to 24.

Option "NoPM" "*boolean***"**
Disables something to do with power management events. Default: PM enabled on platforms that support it.

Option "Xinerama" "*boolean***"**
enable or disable XINERAMA extension. Default is disabled.

Option "AIGLX" "*boolean***"**
enable or disable AIGLX. AIGLX is enabled by default.

Option "DRI2" "*boolean***"**
enable or disable DRI2. DRI2 is disabled by default.

Option "GlxVisuals" "*string***"**
This option controls how many GLX visuals the GLX modules sets up. The default value is **typical**, which will setup up a typical subset of the GLXFBConfigs provided by the driver as GLX visuals. Other options are **minimal**, which will set up the minimal set allowed by the GLX specification and **all** which will setup GLX visuals for all GLXFBConfigs.

Option "UseDefaultFontPath" "*boolean***"**
Include the default font path even if other paths are specified in xorg.conf. If enabled, other font paths are included as well. Enabled by default.

Option "IgnoreABI" "*boolean***"**
Allow modules built for a different, potentially incompatible version of the X server to load. Disabled by default.

Option "AutoAddDevices" "*boolean***"**
If this option is disabled, then no devices will be added from the HAL or udev backends. Enabled by default.

Option "AutoEnableDevices" "*boolean***"**
If this option is disabled, then the devices will be added (and the DevicePresenceNotify event sent), but not enabled, thus leaving policy up to the client. Enabled by default.

Option "AutoAddGPU" "*boolean***"**
If this option is disabled, then no GPU devices will be added from the udev backend. Enabled by default. (May need to be disabled to setup Xinerama).

Option "Log" "*string***"**
This option controls whether the log is flushed and/or synced to disk after each message. Possible values are **flush** or **sync**. Unset by default.

MODULE SECTION

The **Module** section is used to specify which Xorg server modules should be loaded. This section is ignored when the Xorg server is built in static form. The type of modules normally loaded in this section are Xorg server extension modules. Most other module types are loaded automatically when they are needed via other mechanisms. The **Module** section is optional, as are all of the entries that may be specified in it.

Entries in this section may be in two forms. The first and most commonly used form is an entry that uses the **Load** keyword, as described here:

Load "*modulename*"
This instructs the server to load the module called *modulename*. The module name given should be the module's standard name, not the module file name. The standard name is case−sensitive, and does not include the "lib" or "cyg" prefixes, or the ".so" or ".dll" suffixes.

Example: the DRI extension module can be loaded with the following entry:

Load "dri"

Disable "*modulename*"
This instructs the server to not load the module called *modulename*. Some modules are loaded by default in the server, and this overrides that default. If a **Load** instruction is given for the same module, it overrides the **Disable** instruction and the module is loaded. The module name given should be the module's standard name, not the module file name. As with the **Load** instruction, the standard name is case-sensitive, and does not include the "lib" prefix, or the ".a", ".o", or ".so" suffixes.

The second form of entry is a **SubSection,** with the subsection name being the module name, and the contents of the **SubSection** being **Options** that are passed to the module when it is loaded.

Example: the extmod module (which contains a miscellaneous group of server extensions) can be loaded, with the XFree86−DGA extension disabled by using the following entry:

```
SubSection "extmod"
  Option  "omit XFree86−DGA"
EndSubSection
```

Modules are searched for in each directory specified in the **ModulePath** search path, and in the drivers, extensions, input, internal, and multimedia subdirectories of each of those directories. In addition to this, operating system specific subdirectories of all the above are searched first if they exist.

To see what extension modules are available, check the extensions subdirectory under:

/usr/lib64/xorg/modules

The "extmod", "dbe", "dri", "dri2", "glx", and "record" extension modules are loaded automatically, if they are present, unless disabled with "Disable" entries. It is recommended that at very least the "extmod" extension module be loaded. If it isn't, some commonly used server extensions (like the SHAPE extension) will not be available.

EXTENSIONS SECTION

The **Extensions** section is used to specify which X11 protocol extensions should be enabled or disabled. The **Extensions** section is optional, as are all of the entries that may be specified in it.

Entries in this section are listed as Option statements with the name of the extension as the first argument, and a boolean value as the second. The extension name is case−sensitive, and matches the form shown in the output of "Xorg -extension ?".

Example: the MIT-SHM extension can be disabled with the following entry:

```
Section "Extensions"
  Option "MIT-SHM" "Disable"
EndSection
```

INPUTDEVICE SECTION

The config file may have multiple **InputDevice** sections. Recent X servers employ HAL or udev backends for input device enumeration and input hotplugging. It is usually not necessary to provide **InputDevice** sections in the xorg.conf if hotplugging is in use (i.e. AutoAddDevices is enabled). If hotplugging is enabled, **InputDevice** sections using the **mouse, kbd** and **vmmouse** driver will be ignored.

If hotplugging is disabled, there will normally be at least two: one for the core (primary) keyboard and one for the core pointer. If either of these two is missing, a default configuration for the missing ones will be

used. In the absence of an explicitly specified core input device, the first **InputDevice** marked as **CorePointer** (or **CoreKeyboard**) is used. If there is no match there, the first **InputDevice** that uses the "mouse" (or "kbd") driver is used. The final fallback is to use built−in default configurations. Currently the default configuration may not work as expected on all platforms.

InputDevice sections have the following format:

```
Section "InputDevice"
  Identifier "name"
  Driver     "inputdriver"
  options
  ...
EndSection
```

The **Identifier** and **Driver** entries are required in all **InputDevice** sections. All other entries are optional.

The **Identifier** entry specifies the unique name for this input device. The **Driver** entry specifies the name of the driver to use for this input device. When using the loadable server, the input driver module "*inputdriver*" will be loaded for each active **InputDevice** section. An **InputDevice** section is considered active if it is referenced by an active **ServerLayout** section, if it is referenced by the **−keyboard** or **−pointer** command line options, or if it is selected implicitly as the core pointer or keyboard device in the absence of such explicit references. The most commonly used input drivers are **evdev**(4) on Linux systems, and **kbd**(4) and **mousedrv**(4) on other platforms.

InputDevice sections recognise some driver−independent **Options**, which are described here. See the individual input driver manual pages for a description of the device−specific options.

Option "AutoServerLayout" "*boolean***"**
Always add the device to the ServerLayout section used by this instance of the server. This affects implied layouts as well as explicit layouts specified in the configuration and/or on the command line.

Option "CorePointer"
Deprecated, see **Floating**

Option "CoreKeyboard"
Deprecated, see **Floating**

Option "AlwaysCore" "*boolean***"**
Deprecated, see **Floating**

Option "SendCoreEvents" "*boolean***"**
Deprecated, see **Floating**

Option "Floating" "*boolean***"**
When enabled, the input device is set up floating and does not report events through any master device or control a cursor. The device is only available to clients using the X Input Extension API. This option is disabled by default. The options **CorePointer, CoreKeyboard, AlwaysCore,** and **SendCoreEvents,** are the inverse of option **Floating** (i.e. **SendCoreEvents "on"** is equivalent to **Floating "off"**).

This option controls the startup behavior only, a device may be reattached or set floating at runtime.

Option "TransformationMatrix" "*a b c d e f g h i***"**
Specifies the 3x3 transformation matrix for absolute input devices. The input device will be bound to the area given in the matrix. In most configurations, "a" and "e" specify the width and height of the area the device is bound to, and "c" and "f" specify the x and y offset of the area. The value range is 0 to 1, where 1 represents the width or height of all root windows together, 0.5 represents half the area, etc. The values represent a 3x3 matrix, with the first, second and third group of three

values representing the first, second and third row of the matrix, respectively. The identity matrix is "1 0 0 0 1 0 0 0 1".

POINTER ACCELERATION

For pointing devices, the following options control how the pointer is accelerated or decelerated with respect to physical device motion. Most of these can be adjusted at runtime, see the xinput(1) man page for details. Only the most important acceleration options are discussed here.

Option "AccelerationProfile" "*integer*"

Select the profile. In layman's terms, the profile constitutes the "feeling" of the acceleration. More formally, it defines how the transfer function (actual acceleration as a function of current device velocity and acceleration controls) is constructed. This is mainly a matter of personal preference.

0 classic (mostly compatible)
-1 none (only constant deceleration is applied)
1 device-dependent
2 polynomial (polynomial function)
3 smooth linear (soft knee, then linear)
4 simple (normal when slow, otherwise accelerated)
5 power (power function)
6 linear (more speed, more acceleration)
7 limited (like linear, but maxes out at threshold)

Option "ConstantDeceleration" "*real*"

Makes the pointer go **deceleration** times slower than normal. Most useful for high-resolution devices. A value between 0 and 1 will speed up the pointer.

Option "AdaptiveDeceleration" "*real*"

Allows to actually decelerate the pointer when going slow. At most, it will be **adaptive deceleration** times slower. Enables precise pointer placement without sacrificing speed.

Option "AccelerationScheme" "*string*"

Selects the scheme, which is the underlying algorithm.

predictable default algorithm (behaving more predictable)
lightweight old acceleration code (as specified in the X protocol spec)
none no acceleration or deceleration

Option "AccelerationNumerator" "*integer*"

Option "AccelerationDenominator" "*integer*"

Set numerator and denominator of the acceleration factor. The acceleration factor is a rational which, together with threshold, can be used to tweak profiles to suit the users needs. The **simple** and **limited** profiles use it directly (i.e. they accelerate by the factor), for other profiles it should hold that a higher acceleration factor leads to a faster pointer. Typically, 1 is unaccelerated and values up to 5 are sensible.

Option "AccelerationThreshold" "*integer*"

Set the threshold, which is roughly the velocity (usually device units per 10 ms) required for acceleration to become effective. The precise effect varies with the profile however.

INPUTCLASS SECTION

The config file may have multiple **InputClass** sections. These sections are optional and are used to provide configuration for a class of input devices as they are automatically added. An input device can match more than one **InputClass** section. Each class can override settings from a previous class, so it is best to arrange the sections with the most generic matches first.

InputClass sections have the following format:

```
Section "InputClass"
  Identifier "name"
```

entries
...
options
...
EndSection

The **Identifier** entry is required in all **InputClass** sections. All other entries are optional.

The **Identifier** entry specifies the unique name for this input class. The **Driver** entry specifies the name of the driver to use for this input device. After all classes have been examined, the "*inputdriver*" module from the first **Driver** entry will be enabled when using the loadable server.

When an input device is automatically added, its characteristics are checked against all **InputClass** sections. Each section can contain optional entries to narrow the match of the class. If none of the optional entries appear, the **InputClass** section is generic and will match any input device. If more than one of these entries appear, they all must match for the configuration to apply.

There are two types of match entries used in **InputClass** sections. The first allows various tokens to be matched against attributes of the device. An entry can be constructed to match attributes from different devices by separating arguments with a '|' character. Multiple entries of the same type may be supplied to add multiple matching conditions on the same attribute. For example:

Section "InputClass"
Identifier "My Class"
 # product string must contain example and
 # either gizmo or gadget
MatchProduct "example"
MatchProduct "gizmo|gadget"
...
EndSection

MatchProduct "*matchproduct***"**
This entry can be used to check if the substring "*matchproduct*" occurs in the device's product name.

MatchVendor "*matchvendor***"**
This entry can be used to check if the substring "*matchvendor*" occurs in the device's vendor name.

MatchDevicePath "*matchdevice***"**
This entry can be used to check if the device file matches the "*matchdevice*" pathname pattern.

MatchOS "*matchos***"**
This entry can be used to check if the operating system matches the case-insensitive "*matchos*" string. This entry is only supported on platforms providing the **uname**(2) system call.

MatchPnPID "*matchpnp***"**
The device's Plug and Play (PnP) ID can be checked against the "*matchpnp*" shell wildcard pattern.

MatchUSBID "*matchusb***"**
The device's USB ID can be checked against the "*matchusb*" shell wildcard pattern. The ID is constructed as lowercase hexadecimal numbers separated by a ':'. This is the same format as the **lsusb**(8) program.

MatchDriver "*matchdriver***"**
Check the case-sensitive string "*matchdriver*" against the currently configured driver of the device. Ordering of sections using this entry is important since it will not match unless the driver has been set by the config backend or a previous **InputClass** section.

MatchTag "*matchtag*"
This entry can be used to check if tags assigned by the config backend matches the "*matchtag*" pattern. A match is found if at least one of the tags given in "*matchtag*" matches at least one of the tags assigned by the backend.

MatchLayout "*matchlayout*"
Check the case-sensitive string "*matchlayout*" against the currently active **ServerLayout** section. The empty string "" matches an implicit layout which appears if no named **ServerLayout** sections have been found.

The second type of entry is used to match device types. These entries take a boolean argument similar to **Option** entries.

MatchIsKeyboard "*bool*"

MatchIsPointer "*bool*"

MatchIsJoystick "*bool*"

MatchIsTablet "*bool*"

MatchIsTouchpad "*bool*"

MatchIsTouchscreen "*bool*"

When an input device has been matched to the **InputClass** section, any **Option** entries are applied to the device. One **InputClass** specific **Option** is recognized. See the **InputDevice** section above for a description of the remaining **Option** entries.

Option "Ignore" "*boolean***"**
This optional entry specifies that the device should be ignored entirely, and not added to the server. This can be useful when the device is handled by another program and no X events should be generated.

DEVICE SECTION

The config file may have multiple **Device** sections. There must be at least one, for the video card being used.

Device sections have the following format:

```
Section "Device"
  Identifier "name"
  Driver   "driver"
  entries
  ...
EndSection
```

The **Identifier** and **Driver** entries are required in all **Device** sections. All other entries are optional.

The **Identifier** entry specifies the unique name for this graphics device. The **Driver** entry specifies the name of the driver to use for this graphics device. When using the loadable server, the driver module "*driver*" will be loaded for each active **Device** section. A **Device** section is considered active if it is referenced by an active **Screen** section.

Device sections recognise some driver−independent entries and **Options**, which are described here. Not all drivers make use of these driver−independent entries, and many of those that do don't require them to be specified because the information is auto−detected. See the individual graphics driver manual pages for further information about this, and for a description of the device−specific options. Note that most of the **Options** listed here (but not the other entries) may be specified in the **Screen** section instead of here in the **Device** section.

BusID "*bus−id*"
This specifies the bus location of the graphics card. For PCI/AGP cards, the *bus−id* string has the form **PCI:***bus***:***device***:** *function* (e.g., "PCI:1:0:0" might be appropriate for an AGP card). This field is usually optional in single-head configurations when using the primary graphics card. In

multi-head configurations, or when using a secondary graphics card in a single-head configuration, this entry is mandatory. Its main purpose is to make an unambiguous connection between the device section and the hardware it is representing. This information can usually be found by running the pciaccess tool scanpci.

Screen *number*
This option is mandatory for cards where a single PCI entity can drive more than one display (i.e., multiple CRTCs sharing a single graphics accelerator and video memory). One **Device** section is required for each head, and this parameter determines which head each of the **Device** sections applies to. The legal values of *number* range from 0 to one less than the total number of heads per entity. Most drivers require that the primary screen (0) be present.

Chipset "*chipset*"
This usually optional entry specifies the chipset used on the graphics board. In most cases this entry is not required because the drivers will probe the hardware to determine the chipset type. Don't specify it unless the driver-specific documentation recommends that you do.

Ramdac "*ramdac−type*"
This optional entry specifies the type of RAMDAC used on the graphics board. This is only used by a few of the drivers, and in most cases it is not required because the drivers will probe the hardware to determine the RAMDAC type where possible. Don't specify it unless the driver-specific documentation recommends that you do.

DacSpeed *speed*

DacSpeed *speed−8 speed−16 speed−24 speed−32*
This optional entry specifies the RAMDAC speed rating (which is usually printed on the RAMDAC chip). The speed is in MHz. When one value is given, it applies to all framebuffer pixel sizes. When multiple values are given, they apply to the framebuffer pixel sizes 8, 16, 24 and 32 respectively. This is not used by many drivers, and only needs to be specified when the speed rating of the RAMDAC is different from the defaults built in to driver, or when the driver can't autodetect the correct defaults. Don't specify it unless the driver-specific documentation recommends that you do.

Clocks *clock ...*
specifies the pixel that are on your graphics board. The clocks are in MHz, and may be specified as a floating point number. The value is stored internally to the nearest kHz. The ordering of the clocks is important. It must match the order in which they are selected on the graphics board. Multiple **Clocks** lines may be specified, and each is concatenated to form the list. Most drivers do not use this entry, and it is only required for some older boards with non-programmable clocks. Don't specify this entry unless the driver-specific documentation explicitly recommends that you do.

ClockChip "*clockchip−type*"
This optional entry is used to specify the clock chip type on graphics boards which have a programmable clock generator. Only a few Xorg drivers support programmable clock chips. For details, see the appropriate driver manual page.

VideoRam *mem*
This optional entry specifies the amount of video ram that is installed on the graphics board. This is measured in kBytes. In most cases this is not required because the Xorg server probes the graphics board to determine this quantity. The driver-specific documentation should indicate when it might be needed.

BiosBase *baseaddress*
This optional entry specifies the base address of the video BIOS for the VGA board. This address is normally auto-detected, and should only be specified if the driver-specific documentation recommends it.

MemBase *baseaddress*
This optional entry specifies the memory base address of a graphics board's linear frame buffer. This entry is not used by many drivers, and it should only be specified if the driver-specific documentation recommends it.

IOBase *baseaddress*
This optional entry specifies the IO base address. This entry is not used by many drivers, and it should only be specified if the driver-specific documentation recommends it.

ChipID *id*
This optional entry specifies a numerical ID representing the chip type. For PCI cards, it is usually the device ID. This can be used to override the auto-detection, but that should only be done when the driver-specific documentation recommends it.

ChipRev *rev*
This optional entry specifies the chip revision number. This can be used to override the auto-detection, but that should only be done when the driver-specific documentation recommends it.

TextClockFreq *freq*
This optional entry specifies the pixel clock frequency that is used for the regular text mode. The frequency is specified in MHz. This is rarely used.

Option "ModeDebug" "*boolean***"**
Enable printing of additional debugging information about modesetting to the server log.

Options
Option flags may be specified in the **Device** sections. These include driver−specific options and driver−independent options. The former are described in the driver−specific documentation. Some of the latter are described below in the section about the **Screen** section, and they may also be included here.

VIDEOADAPTOR SECTION

Nobody wants to say how this works. Maybe nobody knows ...

MONITOR SECTION

The config file may have multiple **Monitor** sections. There should normally be at least one, for the monitor being used, but a default configuration will be created when one isn't specified.

Monitor sections have the following format:

Section "Monitor"
  Identifier "*name***"**
  entries
  ...
EndSection

The only mandatory entry in a **Monitor** section is the **Identifier** entry.

The **Identifier** entry specifies the unique name for this monitor. The **Monitor** section may be used to provide information about the specifications of the monitor, monitor-specific **Options**, and information about the video modes to use with the monitor.

With RandR 1.2-enabled drivers, monitor sections may be tied to specific outputs of the video card. Using the name of the output defined by the video driver plus the identifier of a monitor section, one associates a monitor section with an output by adding an option to the Device section in the following format:

Option "Monitor-*outputname***" "***monitorsection***"**

(for example, **Option "Monitor-VGA" "VGA monitor"** for a VGA output)

In the absence of specific association of monitor sections to outputs, if a monitor section is present the

server will associate it with an output to preserve compatibility for previous single-head configurations.

Specifying video modes is optional because the server will use the DDC or other information provided by the monitor to automatically configure the list of modes available. When modes are specified explicitly in the **Monitor** section (with the **Mode**, **ModeLine**, or **UseModes** keywords), built-in modes with the same names are not included. Built-in modes with different names are, however, still implicitly included, when they meet the requirements of the monitor.

The entries that may be used in **Monitor** sections are described below.

VendorName **"***vendor***"**
This optional entry specifies the monitor's manufacturer.

ModelName **"***model***"**
This optional entry specifies the monitor's model.

HorizSync *horizsync−range*
gives the range(s) of horizontal sync frequencies supported by the monitor. *horizsync−range* may be a comma separated list of either discrete values or ranges of values. A range of values is two values separated by a dash. By default the values are in units of kHz. They may be specified in MHz or Hz if **MHz** or **Hz** is added to the end of the line. The data given here is used by the Xorg server to determine if video modes are within the specifications of the monitor. This information should be available in the monitor's handbook. If this entry is omitted, a default range of 28−33kHz is used.

VertRefresh *vertrefresh−range*
gives the range(s) of vertical refresh frequencies supported by the monitor. *vertrefresh−range* may be a comma separated list of either discrete values or ranges of values. A range of values is two values separated by a dash. By default the values are in units of Hz. They may be specified in MHz or kHz if **MHz** or **kHz** is added to the end of the line. The data given here is used by the Xorg server to determine if video modes are within the specifications of the monitor. This information should be available in the monitor's handbook. If this entry is omitted, a default range of 43−72Hz is used.

DisplaySize *width height*
This optional entry gives the width and height, in millimetres, of the picture area of the monitor. If given this is used to calculate the horizontal and vertical pitch (DPI) of the screen.

Gamma *gamma−value*

Gamma *red−gamma green−gamma blue−gamma*
This is an optional entry that can be used to specify the gamma correction for the monitor. It may be specified as either a single value or as three separate RGB values. The values should be in the range 0.1 to 10.0, and the default is 1.0. Not all drivers are capable of using this information.

UseModes **"***modesection−id***"**
Include the set of modes listed in the **Modes** section called *modesection−id*. This makes all of the modes defined in that section available for use by this monitor.

Mode **"***name***"**
This is an optional multi-line entry that can be used to provide definitions for video modes for the monitor. In most cases this isn't necessary because the built-in set of VESA standard modes will be sufficient. The **Mode** keyword indicates the start of a multi-line video mode description. The mode description is terminated with the **EndMode** keyword. The mode description consists of the following entries:

DotClock *clock*
is the dot (pixel) clock rate to be used for the mode.

HTimings *hdisp hsyncstart hsyncend htotal*
specifies the horizontal timings for the mode.

VTimings *vdisp vsyncstart vsyncend vtotal*
specifies the vertical timings for the mode.

Flags "*flag*" ...
specifies an optional set of mode flags, each of which is a separate string in double quotes. "**Interlace**" indicates that the mode is interlaced. "**DoubleScan**" indicates a mode where each scanline is doubled. "**+HSync**" and "**−HSync**" can be used to select the polarity of the HSync signal. "**+VSync**" and "**−VSync**" can be used to select the polarity of the VSync signal. "**Composite**" can be used to specify composite sync on hardware where this is supported. Additionally, on some hardware, "**+CSync**" and "**−CSync**" may be used to select the composite sync polarity.

HSkew *hskew*
specifies the number of pixels (towards the right edge of the screen) by which the display enable signal is to be skewed. Not all drivers use this information. This option might become necessary to override the default value supplied by the server (if any). "Roving" horizontal lines indicate this value needs to be increased. If the last few pixels on a scan line appear on the left of the screen, this value should be decreased.

VScan *vscan*
specifies the number of times each scanline is painted on the screen. Not all drivers use this information. Values less than 1 are treated as 1, which is the default. Generally, the **"DoubleScan" Flag** mentioned above doubles this value.

ModeLine "*name*" *mode−description*
This entry is a more compact version of the **Mode** entry, and it also can be used to specify video modes for the monitor. This is a single line format for specifying video modes. In most cases this isn't necessary because the built−in set of VESA standard modes will be sufficient.

The *mode−description* is in four sections, the first three of which are mandatory. The first is the dot (pixel) clock. This is a single number specifying the pixel clock rate for the mode in MHz. The second section is a list of four numbers specifying the horizontal timings. These numbers are the *hdisp*, *hsyncstart*, *hsyncend*, and *htotal* values. The third section is a list of four numbers specifying the vertical timings. These numbers are the *vdisp*, *vsyncstart*, *vsyncend*, and *vtotal* values. The final section is a list of flags specifying other characteristics of the mode. **Interlace** indicates that the mode is interlaced. **DoubleScan** indicates a mode where each scanline is doubled. **+HSync** and **−HSync** can be used to select the polarity of the HSync signal. **+VSync** and **−VSync** can be used to select the polarity of the VSync signal. **Composite** can be used to specify composite sync on hardware where this is supported. Additionally, on some hardware, **+CSync** and **−CSync** may be used to select the composite sync polarity. The **HSkew** and **VScan** options mentioned above in the **Mode** entry description can also be used here.

Option "DPMS" "*bool*"
This option controls whether the server should enable the DPMS extension for power management for this screen. The default is to enable the extension.

Option "SyncOnGreen" "*bool*"
This option controls whether the video card should drive the sync signal on the green color pin. Not all cards support this option, and most monitors do not require it. The default is off.

Option "Primary" "*bool*"
This optional entry specifies that the monitor should be treated as the primary monitor. (RandR 1.2-supporting drivers only)

Option "PreferredMode" "*name*"
This optional entry specifies a mode to be marked as the preferred initial mode of the monitor. (RandR 1.2-supporting drivers only)

Option "ZoomModes" "*name name ...***"**
This optional entry specifies modes to be marked as zoom modes. It is possible to switch to the next and previous mode via **Ctrl+Alt+Keypad−Plus** and **Ctrl+Alt+Keypad−Minus**. All these keypad available modes are selected from the screen mode list. This list is a copy of the compatibility output monitor mode list. Since this output is the output connected to the lowest dot-area monitor, as determined from its largest size mode, that monitor defines the available zoom modes. (RandR 1.2-supporting drivers only)

Option "Position" "*x y***"**
This optional entry specifies the position of the monitor within the X screen. (RandR 1.2-supporting drivers only)

Option "LeftOf" "*output***"**
This optional entry specifies that the monitor should be positioned to the left of the output (not monitor) of the given name. (RandR 1.2-supporting drivers only)

Option "RightOf" "*output***"**
This optional entry specifies that the monitor should be positioned to the right of the output (not monitor) of the given name. (RandR 1.2-supporting drivers only)

Option "Above" "*output***"**
This optional entry specifies that the monitor should be positioned above the output (not monitor) of the given name. (RandR 1.2-supporting drivers only)

Option "Below" "*output***"**
This optional entry specifies that the monitor should be positioned below the output (not monitor) of the given name. (RandR 1.2-supporting drivers only)

Option "Enable" "*bool***"**
This optional entry specifies whether the monitor should be turned on at startup. By default, the server will attempt to enable all connected monitors. (RandR 1.2-supporting drivers only)

Option "DefaultModes" "*bool***"**
This optional entry specifies whether the server should add supported default modes to the list of modes offered on this monitor. By default, the server will add default modes; you should only disable this if you can guarantee that EDID will be available at all times, or if you have added custom modelines which the server can use. (RandR 1.2-supporting drivers only)

Option "MinClock" " *frequency***"**
This optional entry specifies the minimum dot clock, in kHz, that is supported by the monitor.

Option "MaxClock" " *frequency***"**
This optional entry specifies the maximum dot clock, in kHz, that is supported by the monitor.

Option "Ignore" "*bool***"**
This optional entry specifies that the monitor should be ignored entirely, and not reported through RandR. This is useful if the hardware reports the presence of outputs that don't exist. (RandR 1.2-supporting drivers only)

Option "Rotate" "*rotation***"**
This optional entry specifies the initial rotation of the given monitor. Valid values for rotation are "normal", "left", "right", and "inverted". (RandR 1.2-supporting drivers only)

MODES SECTION

The config file may have multiple **Modes** sections, or none. These sections provide a way of defining sets of video modes independently of the **Monitor** sections. **Monitor** sections may include the definitions provided in these sections by using the **UseModes** keyword. In most cases the **Modes** sections are not necessary because the built−in set of VESA standard modes will be sufficient.

Modes sections have the following format:

Section "Modes"

```
    Identifier "name"
    entries
    ...
  EndSection
```

The **Identifier** entry specifies the unique name for this set of mode descriptions. The other entries permitted in **Modes** sections are the **Mode** and **ModeLine** entries that are described above in the **Monitor** section.

SCREEN SECTION

The config file may have multiple **Screen** sections. There must be at least one, for the "screen" being used. A "screen" represents the binding of a graphics device (**Device** section) and a monitor (**Monitor** section). A **Screen** section is considered "active" if it is referenced by an active **ServerLayout** section or by the **−screen** command line option. If neither of those is present, the first **Screen** section found in the config file is considered the active one.

Screen sections have the following format:

```
  Section "Screen"
    Identifier "name"
    Device     "devid"
    Monitor    "monid"
    entries
    ...
    SubSection "Display"
      entries
      ...
    EndSubSection
    ...
  EndSection
```

The **Identifier** entry is mandatory. All others are optional.

The **Identifier** entry specifies the unique name for this screen. The **Screen** section provides information specific to the whole screen, including screen−specific **Options**. In multi−head configurations, there will be multiple active **Screen** sections, one for each head. The entries available for this section are:

Device "*device−id*"
: This entry specifies the **Device** section to be used for this screen. When multiple graphics cards are present, this is what ties a specific card to a screen. The *device−id* must match the **Identifier** of a **Device** section in the config file.

Monitor "*monitor−id*"
: specifies which monitor description is to be used for this screen. If a **Monitor** name is not specified, a default configuration is used. Currently the default configuration may not function as expected on all platforms.

VideoAdaptor "*xv−id*"
: specifies an optional Xv video adaptor description to be used with this screen.

DefaultDepth *depth*
: specifies which color depth the server should use by default. The **−depth** command line option can be used to override this. If neither is specified, the default depth is driver−specific, but in most cases is 8.

DefaultFbBpp *bpp*
: specifies which framebuffer layout to use by default. The **−fbbpp** command line option can be used to override this. In most cases the driver will chose the best default value for this. The only case where there is even a choice in this value is for depth 24, where some hardware supports both a packed 24 bit framebuffer layout and a sparse 32 bit framebuffer layout.

Options

Various **Option** flags may be specified in the **Screen** section. Some are driver−specific and are described in the driver documentation. Others are driver−independent, and will eventually be described here.

Option "Accel"

Enables 2D hardware acceleration. This option is on by default, but it may be necessary to turn it off if there are bugs in the driver. There are many options to disable specific accelerated operations, listed below. Note that disabling an operation will have no effect if the operation is not accelerated (whether due to lack of support in the hardware or in the driver).

Option "InitPrimary" "*boolean***"**

Use the Int10 module to initialize the primary graphics card. Normally, only secondary cards are soft-booted using the Int10 module, as the primary card has already been initialized by the BIOS at boot time. Default: false.

Option "NoInt10" "*boolean***"**

Disables the Int10 module, a module that uses the int10 call to the BIOS of the graphics card to initialize it. Default: false.

Option "NoMTRR"

Disables MTRR (Memory Type Range Register) support, a feature of modern processors which can improve video performance by a factor of up to 2.5. Some hardware has buggy MTRR support, and some video drivers have been known to exhibit problems when MTRR's are used.

Each **Screen** section may optionally contain one or more **Display** subsections. Those subsections provide depth/fbbpp specific configuration information, and the one chosen depends on the depth and/or fbbpp that is being used for the screen. The **Display** subsection format is described in the section below.

DISPLAY SUBSECTION

Each **Screen** section may have multiple **Display** subsections. The "active" **Display** subsection is the first that matches the depth and/or fbbpp values being used, or failing that, the first that has neither a depth or fbbpp value specified. The **Display** subsections are optional. When there isn't one that matches the depth and/or fbbpp values being used, all the parameters that can be specified here fall back to their defaults.

Display subsections have the following format:

SubSection "Display"
  Depth *depth*
  entries
  ...
EndSubSection

Depth *depth*

This entry specifies what colour depth the **Display** subsection is to be used for. This entry is usually specified, but it may be omitted to create a match−all **Display** subsection or when wishing to match only against the **FbBpp** parameter. The range of *depth* values that are allowed depends on the driver. Most drivers support 8, 15, 16 and 24. Some also support 1 and/or 4, and some may support other values (like 30). Note: *depth* means the number of bits in a pixel that are actually used to determine the pixel colour. 32 is not a valid *depth* value. Most hardware that uses 32 bits per pixel only uses 24 of them to hold the colour information, which means that the colour depth is 24, not 32.

FbBpp *bpp*

This entry specifies the framebuffer format this **Display** subsection is to be used for. This entry is only needed when providing depth 24 configurations that allow a choice between a 24 bpp packed framebuffer format and a 32bpp sparse framebuffer format. In most cases this entry should not be used.

Weight *red−weight green−weight blue−weight*
This optional entry specifies the relative RGB weighting to be used for a screen is being used at depth 16 for drivers that allow multiple formats. This may also be specified from the command line with the **−weight** option (see **Xorg(1)).**

Virtual *xdim ydim*
This optional entry specifies the virtual screen resolution to be used. *xdim* must be a multiple of either 8 or 16 for most drivers, and a multiple of 32 when running in monochrome mode. The given value will be rounded down if this is not the case. Video modes which are too large for the specified virtual size will be rejected. If this entry is not present, the virtual screen resolution will be set to accommodate all the valid video modes given in the **Modes** entry. Some drivers/hardware combinations do not support virtual screens. Refer to the appropriate driver−specific documentation for details.

ViewPort *x0 y0*
This optional entry sets the upper left corner of the initial display. This is only relevant when the virtual screen resolution is different from the resolution of the initial video mode. If this entry is not given, then the initial display will be centered in the virtual display area.

Modes "*mode−name*" ...
This optional entry specifies the list of video modes to use. Each *mode−name* specified must be in double quotes. They must correspond to those specified or referenced in the appropriate **Monitor** section (including implicitly referenced built−in VESA standard modes). The server will delete modes from this list which don't satisfy various requirements. The first valid mode in this list will be the default display mode for startup. The list of valid modes is converted internally into a circular list. It is possible to switch to the next mode with **Ctrl+Alt+Keypad−Plus** and to the previous mode with **Ctrl+Alt+Keypad−Minus**. When this entry is omitted, the valid modes referenced by the appropriate **Monitor** section will be used. If the **Monitor** section contains no modes, then the selection will be taken from the built-in VESA standard modes.

Visual "*visual−name*"
This optional entry sets the default root visual type. This may also be specified from the command line (see the **Xserver(1)** man page). The visual types available for depth 8 are (default is **PseudoColor**):

StaticGray
GrayScale
StaticColor
PseudoColor
TrueColor
DirectColor

The visual type available for the depths 15, 16 and 24 are (default is **TrueColor**):

TrueColor
DirectColor

Not all drivers support **DirectColor** at these depths.

The visual types available for the depth 4 are (default is **StaticColor**):

StaticGray
GrayScale
StaticColor
PseudoColor

The visual type available for the depth 1 (monochrome) is **StaticGray**.

Black *red green blue*
This optional entry allows the "black" colour to be specified. This is only supported at depth 1. The default is black.

White *red green blue*
This optional entry allows the "white" colour to be specified. This is only supported at depth 1. The default is white.

Options
Option flags may be specified in the **Display** subsections. These may include driver−specific options and driver−independent options. The former are described in the driver−specific documentation. Some of the latter are described above in the section about the **Screen** section, and they may also be included here.

SERVERLAYOUT SECTION

The config file may have multiple **ServerLayout** sections. A "server layout" represents the binding of one or more screens (**Screen** sections) and one or more input devices (**InputDevice** sections) to form a complete configuration. In multi−head configurations, it also specifies the relative layout of the heads. A **ServerLayout** section is considered "active" if it is referenced by the **−layout** command line option or by an **Option "DefaultServerLayout"** entry in the **ServerFlags** section (the former takes precedence over the latter). If those options are not used, the first **ServerLayout** section found in the config file is considered the active one. If no **ServerLayout** sections are present, the single active screen and two active (core) input devices are selected as described in the relevant sections above.

ServerLayout sections have the following format:

```
Section "ServerLayout"
  Identifier  "name"
  Screen      "screen−id"
  ...
  InputDevice "idev−id"
  ...
  options
  ...
EndSection
```

Each **ServerLayout** section must have an **Identifier** entry and at least one **Screen** entry.

The **Identifier** entry specifies the unique name for this server layout. The **ServerLayout** section provides information specific to the whole session, including session−specific **Options**. The **ServerFlags** options (described above) may be specified here, and ones given here override those given in the **ServerFlags** section.

The entries that may be used in this section are described here.

Screen *screen−num* **"screen−id"** *position−information*
One of these entries must be given for each screen being used in a session. The *screen−id* field is mandatory, and specifies the **Screen** section being referenced. The *screen−num* field is optional, and may be used to specify the screen number in multi−head configurations. When this field is omitted, the screens will be numbered in the order that they are listed in. The numbering starts from 0, and must be consecutive. The *position−information* field describes the way multiple screens are positioned. There are a number of different ways that this information can be provided:

x y

Absolute *x y*
These both specify that the upper left corner's coordinates are (*x*,*y*). The **Absolute** keyword is optional. Some older versions of XFree86 (4.2 and earlier) don't recognise the **Absolute** keyword, so it's safest to just specify the coordinates without it.

RightOf "*screen−id*"

LeftOf "*screen−id*"

Above "*screen−id*"

Below "*screen−id*"

Relative "*screen−id*" *x y*

These give the screen's location relative to another screen. The first four position the screen immediately to the right, left, above or below the other screen. When positioning to the right or left, the top edges are aligned. When positioning above or below, the left edges are aligned. The **Relative** form specifies the offset of the screen's origin (upper left corner) relative to the origin of another screen.

InputDevice "*idev−id*" "*option*" ...

One of these entries should be given for each input device being used in a session. Normally at least two are required, one each for the core pointer and keyboard devices. If either of those is missing, suitable **InputDevice** entries are searched for using the method described above in the **INPUTDEVICE** section. The *idev−id* field is mandatory, and specifies the name of the **InputDevice** section being referenced. Multiple *option* fields may be specified, each in double quotes. The options permitted here are any that may also be given in the **InputDevice** sections. Normally only session−specific input device options would be used here. The most commonly used options are:

"CorePointer"
"CoreKeyboard"
"SendCoreEvents"

and the first two should normally be used to indicate the core pointer and core keyboard devices respectively.

Options

In addition to the following, any option permitted in the **ServerFlags** section may also be specified here. When the same option appears in both places, the value given here overrides the one given in the **ServerFlags** section.

Option "IsolateDevice" "*bus−id*"

Restrict device resets to the specified *bus−id*. See the **BusID** option (described in **DEVICE SECTION**, above) for the format of the *bus−id* parameter. This option overrides **SingleCard**, if specified. At present, only PCI devices can be isolated in this manner.

Option "SingleCard" "*boolean*"

As **IsolateDevice**, except that the bus ID of the first device in the layout is used.

Here is an example of a **ServerLayout** section for a dual headed configuration with two mice:

```
Section "ServerLayout"
  Identifier  "Layout 1"
  Screen      "MGA 1"
  Screen      "MGA 2" RightOf "MGA 1"
  InputDevice "Keyboard 1" "CoreKeyboard"
  InputDevice "Mouse 1"    "CorePointer"
  InputDevice "Mouse 2"    "SendCoreEvents"
  Option      "BlankTime"  "5"
EndSection
```

DRI SECTION

This optional section is used to provide some information for the Direct Rendering Infrastructure. Details about the format of this section can be found on-line at <*http://dri.freedesktop.org/*>.

VENDOR SECTION

The optional **Vendor** section may be used to provide vendor−specific configuration information. Multiple **Vendor** sections may be present, and they may contain an **Identifier** entry and multiple **Option** flags. The data therein is not used in this release.

SEE ALSO

General: **X**(7), **Xserver**(1), **Xorg**(1), **cvt**(1), **gtf**(1).

Not all modules or interfaces are available on all platforms.

Display drivers: **apm**(4), **ati**(4), **chips**(4), **cirrus**(4), **cyrix**(4), **fbdev**(4), **glide**(4), **glint**(4), **i128**(4), **i740**(4), **imstt**(4), **intel**(4), **mga**(4), **neomagic**(4), **nv**(4), **openchrome**(4), **r128**(4), **radeon**(4), **rendition**(4), **savage**(4), **s3virge**(4), **siliconmotion**(4), **sis**(4), **sisusb**(4), **sunbw2**(4), **suncg14**(4), **suncg3**(4), **suncg6**(4), **sunffb**(4), **sunleo**(4), **suntcx**(4), **tdfx**(4), **trident**(4), **tseng**(4), **vesa**(4), **vmware**(4), **voodoo**(4), **wsfb**(4), **xgi**(4), **xgixp**(4).

Input drivers: **acecad**(4), **citron**(4), **elographics**(4), **evdev**(4), **fpit**(4), **joystick**(4), **kbd**(4), **mousedrv**(4), **mutouch**(4), **penmount**(4), **synaptics**(4), **vmmouse**(4), **void**(4), **wacom**(4).

Other modules and interfaces: **exa**(4), **fbdevhw**(4), **v4l**(4).

AUTHORS

This manual page was largely rewritten by David Dawes <*dawes@xfree86.org*>.