**The X Configuration File Format**  
The X configuration file’s name and location vary with the version of X being run:  
**X.org-X11** This server’s configuration file is called xorg.conf, and it’s usually stored in /  
etc/X11, although /etc and several other locations are also acceptable to the server.

Many modern X.org-X11 configurations omit the X configuration file  
entirely, instead relying on runtime auto-detection of hardware. This  
often works fine, but if X doesn’t work or if some of its features are set  
incorrectly, you may need to generate an xorg.conf file by typing Xorg  
-configure when X is *not* running and edit the file manually, as described  
in subsequent sections.

**XFree86 4.*x*** The XFree86 4.*x* configuration file is called XF86Config-4 or XF86Config,  
and is found in /etc/X11 or sometimes in /etc. This file’s format is the same as for the  
X.org-X11 configuration file.  
**XFree86 3.3.6 and earlier** The X configuration file’s name is XF86Config, and the file is  
most commonly located in /etc/X11 or /etc. Although the filename can be the same as for  
XFree86 4.*x*, the file format is slightly different. This book, like the exam, covers the newer  
format used by X.org-X11 and XFree86 4.*x*.

All three of these classes of X server use configuration files that are broken down into  
multiline sections, one section for each major feature. These sections begin with a line  
consisting of the keyword Section and the section name in quotes and end with the  
keyword EndSection:

Section "InputDevice"  
Identifier "Keyboard0"  
Driver "kbd"  
Option "XkbModel" "pc105"  
Option "XkbLayout" "us"  
Option "AutoRepeat" "500 200"  
EndSection  
This section tells X about the keyboard—its model, layout, and so on. Details for the  
sections you’re most likely to need to adjust are described shortly in “X Configuration  
Options.”  
For the most part, the different X servers support the same sections and most of the  
same option names. A few exceptions to this rule do exist, however:  
■ The Option keyword isn’t used in XFree86 3.3.6 and earlier. Instead, the option name  
(such as XkbLayout or AutoRepeat in the preceding example) appears without quotes as  
the first word on the line.  
■ XFree86 3.3.6 and earlier don’t use the ServerLayout section, described later in  
“Putting It All Together.”  
■ XFree86 3.3.6 and earlier lack the Identifier and Driver lines, which are common in  
the XFree86 4.*x* and X.org-X11 configuration files.  
■ Some section-specific features vary between versions. We describe the most important  
of these in the coming pages.

**The X Configure-and-Test Cycle**  
If your X configuration isn’t working correctly, you need to be able to modify that configuration and then test it. Many Linux distributions configure the system to start X automatically, but starting X automatically can make it difficult to test the X configuration. To a  
new Linux administrator, the only obvious way to test a new configuration is to reboot the  
computer.  
A better solution is to kick the system into a mode in which X is *not* started automatically. On Red Hat, Fedora, and similar distributions, this goal can be achieved by typing  
**telinit 3**. This action sets the computer to use runlevel 3, in which X normally doesn’t  
run. Chapter 5, “Booting Linux and Editing Files,” covers runlevels in more detail.  
Some distributions, such as Debian, Ubuntu, and Gentoo, don’t use runlevels as a signal  
for whether to start X. With such distributions, you must shut down the GUI login server  
by typing **/etc/init.d/xdm stop**. (You may need to change xdm to gdm, kdm, mdm, or  
lightdm, depending on your configuration.)  
Once the X session is shut down, you can log in using a text-mode login prompt and  
tweak your X settings manually, or you can use text-based X configuration programs. You  
can then type **startx** to start the X server again. If you get the desired results, quit from X  
(typically by selecting a “log out” option in your desktop environment) and type **telinit**  
**5** (**/etc/init.d/xdm start** in Debian and other distributions that don’t use runlevels to

start the GUI login prompt) to restore the system to its normal X login screen. If after  
typing **startx** you don’t get the results you want, you can end your X session and try  
modifying the system further.  
If X is working minimally but you want to modify it using X-based configuration tools,  
you can do so after typing **startx** to get a normal X session running. Alternatively, you  
can reconfigure the system before taking it out of the X-enabled runlevel.  
Another approach to restarting X is to leave the system in its X-enabled runlevel and  
then kill the X server. The Ctrl+Alt+Backspace keystroke does this on many systems, or you  
can do it manually with the kill command after finding the appropriate process ID with  
the ps command, as shown here:  
# **ps ax | grep X**  
1375 ? S 6:32 /usr/bin/X -auth /var/gdm/:0.Xauth  
# **kill 1375**  
This approach works better on systems that don’t map the running of X to specific runlevels, such as Debian and its derivatives.