

# Getting Started [2]

---

This section is for new X Window System users; it describes the following procedures:

- Accessing X on your workstation
- Logging in to a Cray Research system from your workstation
- Setting environment variables
- Running a client on the Cray Research system and displaying the output on your X workstation

The general methods presented in this section apply to all workstations. If you are following along at your workstation, the prompt you see will be one that is unique to your workstation. In the sample sessions that follow, the workstation is a Sun X11 server named `mirror`, and the prompt is `mirror%`. Similarly, in the sample session, the name of the Cray Research system is `sn2003`.

## Accessing X on your workstation

### 2.1

To access X on your workstation, you must access the server by adding a path to the X11 binary files in your `.profile` (for the Bourne shell and Korn shell), `.login` (for the C shell), or `.cshrc` file (for the C shell), as follows:

```
set path=($path /usr/bin/X11)           (C shell)
PATH=$PATH:/usr/bin/X11                 (Bourne shell,
export PATH                             Korn shell)
```

You can also do this from the command line.

Next, from your `.profile`, `.login`, or `.cshrc` file, or from the command line (if you are not already in a window system), set the `DISPLAY` environment variable to your server name, as follows:

```
mirror% setenv DISPLAY mirror:0.0    (C shell)

mirror$ DISPLAY=mirror:0.0 (Bourne shell,
mirror$ export DISPLAY              Korn shell)
```

To start your environment, create a `.xinitrc` file. The following is a sample `.xinitrc` file that contains commands to start some of the tools that you will be using. This sample shows `xclock`, which displays time; `xload`, which displays the system load average; `xbiff`, which is a mail notification program; and `xterm`, which is a terminal emulator. It starts the `twm` (tab window manager, formerly known as Tom's window manager), as an example, but other window managers such as `mwm` may be substituted.

```
xhost + sn2003
#
# note: The preceding item enables access for host sn2003
#
twm &
xclock -analog -geometry 64x64-0-0 &
xload -rv -geometry 150x64-64-0 &
xbiff -rv -geometry 64x64-214-0 &
xterm -geometry 80x60-0+0 -fn 8x13 &
#
# note: the last item must not be run in background.
#
xterm -C -g 51x12+0+0
```

You can customize your X environment by changing your standard dot files and adding some new dot files. "Example Files," page 59, contains examples of all of the files you will need.

To bring up the X11 server, add the following entry in your `.login` or `.profile` file:

```
xinit
```

When you log in or execute this entry at the command line, you should see a terminal window in the upper left corner of the screen and additional windows, as specified in your `.xinitrc` file (see the previous sample file).

In addition to using the `xhost(1X)` program, you can use X authority files (`xauth(1X)`) to control access to your display. With authority files, you must create a key, which clients must use to access your display server.

If your workstation is not a UNIX workstation, see the X documentation that is specific to your workstation for instructions on starting your environment.

## Logging in to a Cray Research system from your workstation

2.2

In the following sample session, you will be logging in to a Cray Research system (`sn2003`) running UNICOS, to run the client named `xclock`. You will be displaying the clock on your workstation, `mirror`. To begin a remote login to the Cray Research system, use either `telnet(1B)` or `rlogin(1B)` in an `xterm` window, as follows:

```
mirror% telnet sn2003
```

```
mirror% rlogin sn2003
```

After you have provided a user ID and password, and standard login messages have appeared, a Cray prompt similar to the following appears:

```
sn2003%
```

## Setting environment variables

2.3

Now that you are logged in to a Cray Research system, you must set the environment variables by entering the commands shown in this subsection. You can enter these commands at the prompt, or put them into your `.profile`, `.login`, or `.cshrc` file.

The following commands put `/usr/bin/X11` in your directory search path:

```
sn2003% set path=($path /usr/bin/X11) (C shell)
sn2003$ PATH=$PATH:/usr/bin/X11      (Bourne shell,
sn2003$ export PATH                  Korn shell)
```

The following command sets the `DISPLAY` environment variable to your server name (`mirror`):

```
sn2003% setenv DISPLAY mirror:0.0 (C shell)
sn2003$ DISPLAY=mirror:0.0      (Bourne shell,
sn2003$ export DISPLAY          Korn shell)
```

The following command sets the terminal type environment variable, `TERM`, to `xterm`:

```
sn2003% set term xterm          (C shell)
sn2003$ TERM=xterm              (Bourne shell,
sn2003$ export TERM            Korn shell)
```

## Running a client on the Cray Research system

2.4

Now you are ready to run programs (known as *clients*) on the Cray Research system. The following example illustrates the use of a Cray Research system to produce graphics interactively on a server workstation. Run the `xclock` client on the Cray Research system, as follows:

```
sn2003% xclock &
```

A clock is displayed on your workstation. See `xclock(1X)` for available options; for more details about `xclock`, see the O'Reilly, volume 3, documentation, listed in the preface.

# Writing Your Own Clients [3]

---

This section describes the unique features of X programming on the Cray Research system, and it offers programming techniques to make the most efficient use of the Cray Research system. See the preface for a list of references that describe general X programming techniques.

The following topics are presented in this section:

- “Compiling a client” shows how to write a client.
- “Handling events,” page 10, describes how to minimize network traffic, thus maximizing efficiency.
- “Using colors,” page 11, discusses issues that pertain to color graphics.
- “Using fonts,” page 14, describes techniques for efficient use of fonts.
- “Using images,” page 15, discusses the use of client-side raster images.
- “Debugging tools,” page 17, describes the use of the `cdbx(1)` and `xscope` debugging tools.

## Compiling a client

3.1

When you build an X client on a Cray Research system, you must load the correct X libraries with the user-written code to create the executable binary file. The X11R5 libraries are as follows:

<code>libXext.a</code>	(Extension library)
<code>libXaw.a</code>	(Athena widget library)
<code>libXt.a</code>	(Intrinsics toolkit)
<code>libX11.a</code>	(also known as <code>Xlib</code> )
<code>libXmu.a</code>	(MIT utility library)