7 X startup

So far, we have just taken our X system as it came, once we had done whatever it took to start it up in section 2. This section will describe what typically happens at start up and will explain how a user can configure his or her own X environment.

The most accurate answer is, unfortunately, "it depends". X's almost infinite range of configuration options is going to bite hard here. Here are a couple of fairly common scenarios, the first covering X-terminals and workstations that display a login window, the second covering those that start up in text mode and require a command to be typed.

7.1 X startup with an X-terminal

When the machine booted it started running a process called `xdm` ("X display manager") running with `root` privileges. `xdm` then

1. creates a random number to act as a magic cookie,
2. starts an X server on its display with instructions to obey only X instructions that have the correct cookie,
3. directs the server to pop up the login window,
4. waits for a user to successfully authenticate him- or herself,
5. thinks up another cookie,
6. restarts the X server with the new server needing the new cookie,
7. starts up another `xdm` owned by the user which also knows the cookie.

The child `xdm` then

1. puts the cookie in the user's `.Xauthority` file,

2. runs a session start up script which typically calls the user's `.xsession` file if such exists,
3. instructs the server to close down any remaining windows,
4. instructs the server to shut down,
5. stops, passing control back to the parent `xdm`.

N.B.: All the user's work must be done while the `.xsession` *script is still running.* See section 7.3.

7.2 X startup from text mode

The `startx` program typically does the following:

1. creates a random cookie and places it in the user's `.Xauthority` file,
2. runs the `xinit` program,
3. puts the system back into text mode when `xinit` finishes.

The `xinit` program in turn does the following:

1. starts the `X` server with the arguments to pass it the cookie,
2. runs a startup script which typically calls the user's `.xinitrc` file if such exists,

   NB: All the user's work must be done while the `.xinitrc` script is still running. See section 7.3.
3. instructs the server to close down any remaining windows,
4. instructs the server to shut down.

Notes:

- In some systems, `startx` actively remove the cookie from the `.Xauthority` file. On others it leaves it there to be overwritten.
- On some older systems `startx` does not set up any cookies, leaving your session at the mercy of the `xhosts` mechanism.

7.3 The `.xsession/.xinitrc` file

This file is a shell script, and therefore must be executable. Here is an example of a simple `.xsession` file:

```
#!/bin/sh

umask 022
xhost - 1> /dev/null 2>&1

# Set resources.
[ -r $HOME/.Xresources ] && xrdb -load $HOME/.Xresources

# Start some clients.
xbiff -geometry -120+0&
xclock -geometry 112x112-0+0&

# Start window manager
twm||xterm -name "Session" -ls
```

Now we consider this line by line:

1. `#!/bin/sh`
   - This is a standard shell script.
   - It must, therefore, be executable.
   - It runs in an environment that has `DISPLAY` set.
2. `umask 022`
   - The default `umask` may need changing, however.

- The value `022` causes any files created not to be world- or group-writable.

3. `[ -r $HOME/.Xresources ] && xrdb -load $HOME/.Xresources`

   - This line sets the user's resources in the server.
   - The `DISPLAY` environment variable is set for `xrdb` to use.
   - The `HOME` environment variable is set for the shell to use.
   - The use of `xrdb` assumes that permissions are already set up.

4. `xbiff -geometry -120+0&`

5. `xclock -geometry 112x112-0+0&`

   - These two lines set up some basic `X` clients.
   - Note that both are backgrounded so the flow of control will pass on to subsequent lines.
   - Both commands are run locally so there is no need for use of `xon`.
   - If my mail arrived on a remote machine, the line might read

     ```
     xon remote xbiff -geometry -120+0
     ```

6. `twm||xterm -name "Session" -ls`

   - This line will attempt to start the `twm` `X` client, and if it can't will start an `xterm` instead.
   - Certain `X` servers supply local window managers (*e.g.* Vista-eXceed) and the launch of a second one will fail.
   - Whichever of `twm` or `xterm` is started is *not* backgrounded.
   - Once this last program finishes, the `X` server will shut down again.
   - Exiting the window manager or logging out of the "Session" `xterm` will end the entire `X` session.
   - There is more information about `twm` in section 6.

7.4 Troubleshooting

- `.xsession` not executable

  If `.xsession` is not executable, then it will be ignored and you will get the system default behaviour.

- `.xsession` stops too early

  If one of the local clients is not run in the background (*i.e.* the final ampersand is missing from the line) then the startup will stop early and the window manager will not be launched.

- `.xsession` does not stop

  If the last line of the `.xsession` is backgrounded, the flow on control will pass by that line, the `.xsession` script will terminate and the `X` server will shut down.

- *All the user's work must be done while* `.xsession` *is running.*

  While files like `.bash_profile` are configuration scripts, `.xsession` is a standard shell script.