

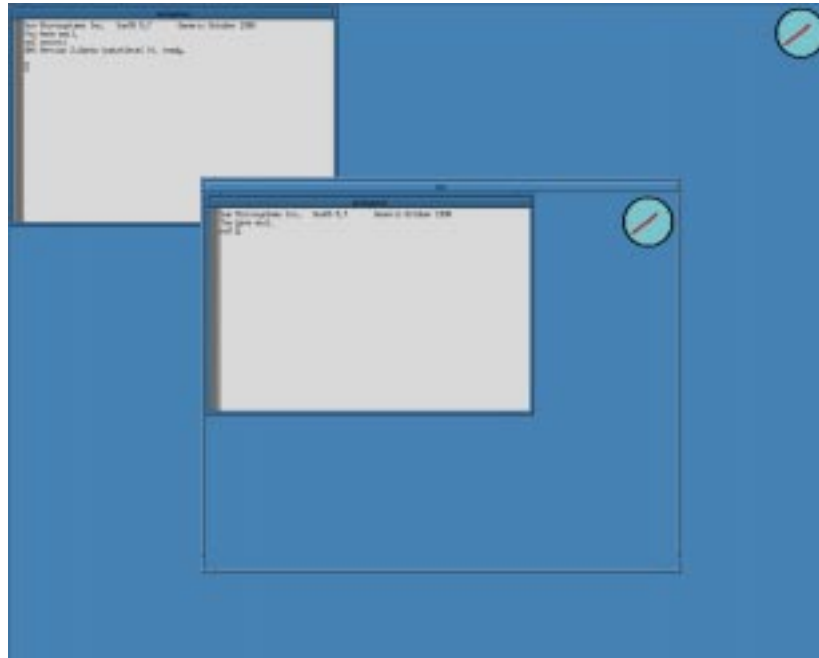
Sharing X Applications With XMX

A Quick Tutorial

John Bazik
jsb@cs.brown.edu

The Virtual Root Window

X client applications draw on a root window, which is just the background upon which all other windows appear. XMX creates a *virtual root window*, which is a root-window-in-a-window. The virtual root window encapsulates an entire X session.



Run xmx

```
% xmx
```

Assuming you are in an X session and your `DISPLAY` environment variable is set correctly, you should see a virtual root window appear that is a third the size of your screen and has the familiar “root weave” pattern on the background. There are no X client applications running in the session, so there is nothing you can do. Kill it and try this

```
% xmxinit
```

Again, you should see the root weave virtual root window, but this time you will see a lone `xterm` in the upper left corner. Experienced X users will recognize that this is exactly what `xinit` does when a user lacks a `.xinitrc` file. `xinit` is used to start an X session. `xmxinit` is used to start an XMX session. A `.xinitrc` file is a per-user shell script that runs all the X client applications the user wants to see when starting an X session. Similarly, a `.xmxinitrc` file runs all the X client applications a user wants to see when starting up an XMX session.

The virtual X session is completely separate and self-contained. From the `xterm` in the virtual root window, run a window manager. Try running some of your favorite applications. Create a `.xmxinitrc` file. Use your `.xinitrc` file as a model, but avoid doing things that don't make sense in a nested X session, like running `xmodmap` or redirecting console output (by running `xconsole` or `xterm -C`). To exit the XMx session, type "exit" in the `xterm`. Rerun `xmxinit` to try out your new session.

Multiple Displays

A nested X server is a neat trick, but XMx is really about application sharing. Create a shared X session

```
% xmxinit -- -floor $DISPLAY
```

Two identical virtual root windows appear on your screen. Notice that as your pointer tracks through one, a *telepointer* echos the movement in the other. Anything you do in one appears in both. XMx provides a WYSIWIS (What You See Is What I See) shared environment.

Log onto a different host and start an X session (or ask a friend who is logged in to help). On the remote host, issue an `xhost` command

```
remote% xhost +myhost
```

Now, on myhost, run a shared session

```
myhost% xmxinit -- -floor remote:0
```

Again, two virtual root windows appear, but this time there is one on each of the two machines involved. Both users have control over the virtual X session.

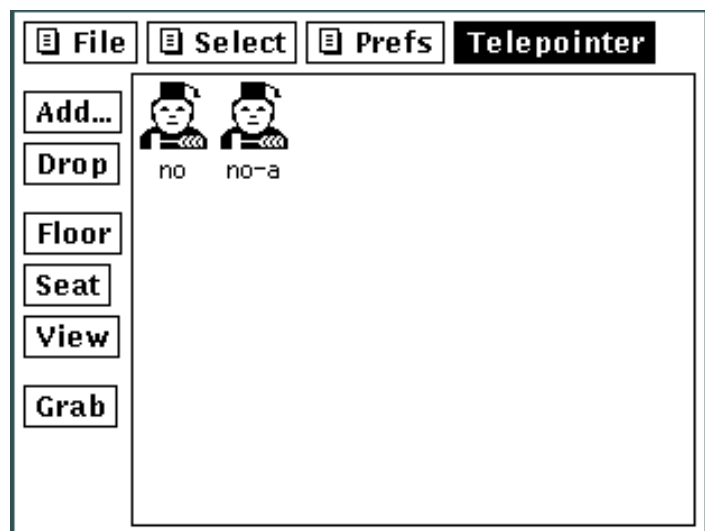
Controlling It

From within a shared X session, run

```
% xmtg
```

`xmtg` is an XMC (X Multiplexor Control) client application. It allows you to control XMx. The people icons represent participants in the session. Here you see two virtual root windows on a machine named "no" (`xmtg` added `-a` to disambiguate the second one).

Select an icon and click on the Drop button. The virtual root window disappears. Click on Add and enter the display string for the just-dropped display, and it reappears. Try adding a new display. Try clicking on Telepointer. Telepointers are optional. The buttons named Floor, Seat and View allow you to control which displays may provide keyboard and mouse input to shared applications.



Floor, Seat, View

Xmx provides three *input modes* for participants. These input modes dictate what happens when a user types or moves the mouse in a virtual root window. They are named for the roles one might have at a relatively formal meeting. The only mode we have encountered so far is Floor. When you have the Floor, your keyboard and mouse inputs control the shared applications in the virtual X session. Having the floor is like running a meeting, you're in control and everyone is watching what you do.

View mode is the exact opposite of floor mode. When a display is in view mode, the user may only watch passively what goes on in the session. All mouse and keyboard input is ignored. Think of view mode as being in the spectator's gallery.

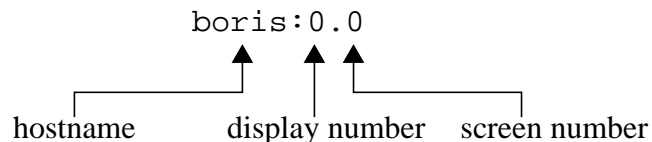
Seat mode is *inbetween* floor and view modes. When a display is in seat mode, the user's mouse and keyboard input is ignored by the X clients in the shared session, just like in view mode. But an XMC client, like xmtg, can choose to see input events from displays in seat mode. Think of seat mode as being a member of some organization that is having a meeting - it gives you the right to be recognized and participate, though you are not running things at the moment.

Try running a session with multiple displays (even if they are all on one machine). Try changing the input modes of the virtual root windows. First select an icon in xmtg, then click on either the Floor, Seat or View button to change it's mode. The icon changes to reflect the input mode: displays in floor mode have a gavel and a hat, displays in seat mode have only a hat, displays in view mode are bareheaded. Xmtg insists that you have at least one display in floor mode - otherwise you can do nothing!

When you have a display (a virtual root window) in seat mode, try clicking on the xmtg window there. Xmtg will see the mouse click and raise the hand of the corresponding seat-mode icon. All the other X clients ignore your inputs. In this way, an XMC client can coordinate giving the floor to, or sharing the floor among participants in the shared session.

DISPLAY Magic

X clients need to know what X server to talk to. They determine this from either the `-display` option provided on the command line, or from the `DISPLAY` environment variable. The *display string* has several components.



All running X servers can be located by a display string. So, too, can a running XMX. By convention, X servers usually default to display zero, as shown above. If there is more than one X server on a particular machine, it must have a different display number, so XMX, when it is acting like an X server, defaults to display number one to avoid conflicting with the local X server.

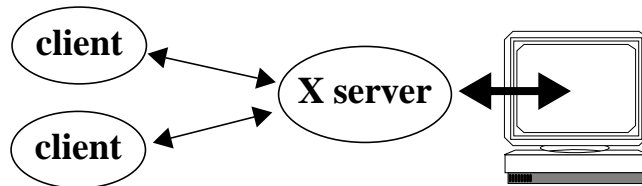
Note that X clients use a display string to find XMX, but XMX also uses display strings to find the X servers to which it connects.

```
% xmx -display :0 &
% xterm -display :1
```

In this example, XMX connects to the X server that is running on the local host at display number zero (since no screen number is given, it is assumed to be zero). Then an xterm connects to XMX, thinking it is an X server, and is displayed within XMX's virtual root window. These two commands are exactly what `xmxcinit` did when it found no `.xmxcinitrc` file.

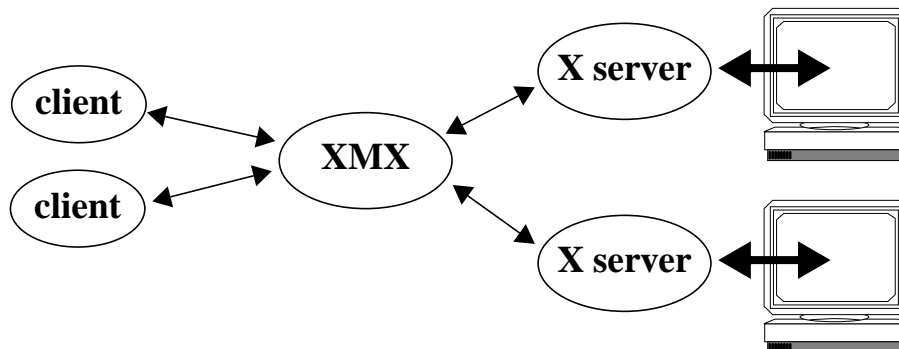
How Does XMX Work?

X applications or *clients*, like `xterm`, do not draw on your screen or watch your keyboard and mouse. They talk to an X server which does that for them. X clients talk to an X server over a data pipeline called a socket. A single X server does all the drawing for all the applications displayed on your machine, and it watches your keyboard and mouse for all the events that any of those applications are interested in.



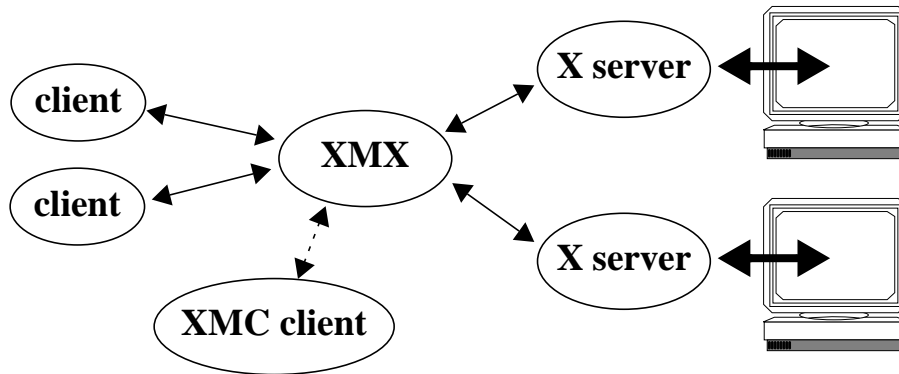
An X client does not need to be running on the same machine as the X server to which it is talking. You can run X clients on faraway machines, and display them on your desktop, and vice versa. And though few clients do, an X client can connect to, and draw on, and receive events from, as many displays as it likes. These are all features of X. X is a *network-transparent* windowing system.

XMX is an X client. But instead of connecting to only one X server, it connects to many X servers, drawing exactly the same thing on each of them. XMX also pretends to be an X server, so that other X clients can talk to it. In this way, it allows many people to share the same client applications, without those clients knowing that there are many people.



Because XMX is *interposed* between X clients and X servers, it cannot *jump into* an existing client-server session and begin sharing it. XMX can only share X sessions it was involved in from the start.

When sharing an X session like this, it is helpful to have participants (X servers) join and leave the session and to control which participant(s) can interact with the X clients while the others watch. This control is provided by a separate XMC (X Multiplexor Control) connection.



Because the XMC connection is completely separate from the X connection, an XMC client does not have to be displayed in the shared session. It can be displayed in a different, non-shared X session, or it can have no graphical user interface at all. In fact, there is a unix command-line XMC application called `xmc`, which is included in the `xmx` distribution.

Like the X Window System itself, XMX does not set any policy. The way a shared session works, the way that input modes are assigned, and users join or leave, is entirely up to the XMC clients that control it. `Xmtg` is merely a sample.