

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour ✕

Find a free X11 display number



I have some unit tests that need an X11 display so I plan to start Xvfb before running them, but to start Xvfb I will need a free display number to connect it to. My best guess is to see what's free in `/tmp/.X11-unix` but I'm not sure how to handle the race if many tests try to start simultaneously.

sshd must do this, does anyone know how?

x11

asked Mar 26 '10 at 1:48



James

292 1 3 14

4 Answers

There's no point in trying to find a free display number. As you have guessed, between the time you find a free one and the time Xvfb starts, another X server might have taken the port you thought was free. So, better to just try to launch Xvfb, handle the failure if the port is taken, and then retry at the next port until you succeed or run out of ports to try.

```
#!/bin/bash
DISPLAY_NUM=0
unset TEST_HAS_RUN
until [ $TEST_HAS_RUN ] || (( $DISPLAY_NUM > 10 ))
do
  Xvfb :$DISPLAY_NUM &
  jobs
  sleep 2 # assumption here is that Xvfb will exit quickly if it can't launch
  if jobs | grep Xvfb
  then
    echo launching test on :$DISPLAY_NUM
    xterm -display :$DISPLAY_NUM
    TEST_HAS_RUN=1
    kill %-
  else
    let DISPLAY_NUM=$DISPLAY_NUM+1
  fi
done
```

answered Mar 26 '10 at 2:58



karunski

2,157 8 9

Work on work you love. From home.



stackoverflowcareers

Why not exploit the fact that every X11 server puts a "Lock" file into /tmp?

This is called `/tmp/.Xn-lock` where "n" is the Display id. (Also note the leading `.` in the filename).

This is the mechanism that Xserver itself uses to check for duplication, and it appears to be consistent on all *nix platforms I have tried (HP-UX, Linux, ...)

So you could adapt your script thus (forgive me for syntax errors, I'm more accustomed to C shell than Bourne/Korn shell scripting)

```
DISPLAY_NUM=0
do
  if ( -e /tmp/.X$DISPLAY_NUM-lock ) then
    let DISPLAY_NUM=$DISPLAY_NUM+1
```

```

else
    Xvfb :$DISPLAY_NUM -screen 0 1280x1024x24 -ac (or whatever args take your fancy)
fi
done

```

edited Feb 15 '13 at 10:53



lesmana
9,276 4 33 55

answered Nov 9 '11 at 10:51



Christopher Bell
61 1 2

With full credit to [this answer](#) to the related question [How high do X11 display numbers go?](#):

Recent X servers as of version 1.13 (`xvfb` , too) support the `-displayfd <fd>` command line option: It will make the X server choose the display itself and write the display number back to the file descriptor `<fd>` . It's a bit convoluted, but this would be a safe and race-condition-free way to ask `xvfb` to use any free display. A `bash` example:

```

exec 6>display.log
Xvfb -displayfd 6
# The display number of the new Xvfb instance has been written to display.log
# Kill Xvfb
exec 6>&-

```

edited Aug 10 '13 at 21:45



answered Aug 10 '13 at 21:40
krtmlr
6,271 2 21 57

Just a note: In Debian `xvfb` v1.14 is available in [jessie](#) , that means that in average distro should be quite recent (2nd half 2013 or later). – [dma_k](#) Jan 15 '14 at 9:48

Based in the answer of [@karunski](#).

Using `Xvfb` to probe the displays, and `lsof` to check if are unix sockets in the `Xvfb` process, is more effective, notice the `sleep 0.5` , can be variable depends on the machine.

```

#!/bin/bash
DISPLAY=0

until [ $DISPLAY_NUM > 10 ]; do
    echo -n "Looking for display on $DISPLAY..."
    Xvfb :$DISPLAY > /dev/null 2>&1 &
    pid=$!
    sleep 0.5
    lsof -a -U -p $pid > /dev/null 2>&1

    notfound="$?"
    kill $pid > /dev/null 2>&1

    wait $pid

    [ "$notfound" == "0" ] && echo "found" && break

    echo "fail"
    let DISPLAY=DISPLAY+1
done

```

answered Aug 14 '13 at 17:58



Felipe Alcacibar
5,837 2 10 16