

NAME

`x11perf` - X11 server performance test program

SYNTAX

`x11perf` [-option ...]

DESCRIPTION

The *x11perf* program runs one or more performance tests and reports how fast an X server can execute the tests.

Many graphics benchmarks assume that the graphics device is used to display the output of a single fancy graphics application, and that the user gets his work done on some other device, like a terminal. Such benchmarks usually measure drawing speed for lines, polygons, text, etc.

Since workstations are not used as standalone graphics engines, but as super-terminals, *x11perf* measures window management performance as well as traditional graphics performance. *x11perf* includes benchmarks for the time it takes to create and map windows (as when you start up an application); to map a pre-existing set of windows onto the screen (as when you deiconify an application or pop up a menu); and to rearrange windows (as when you slosh windows to and fro trying to find the one you want).

x11perf also measures graphics performance for operations not normally used in standalone graphics displays, but are nonetheless used frequently by X applications. Such operations include CopyPlane (used to map bitmaps into pixels), scrolling (used in text windows), and various stipples and tiles (used for CAD and color half-toning, respectively).

x11perf should be used to analyze particular strengths and weaknesses of servers, and is most useful to a server writer who wants to analyze and improve a server. *x11perf* is meant to comprehensively exercise just about every X11 operation you can perform; it does not purport to be a representative sample of the operations that X11 applications actually use. While it can be used as a benchmark, it was written and is intended as a performance testing tool.

As such, *x11perf* DOES NOT whittle down measurements to a single ‘‘HeXStones’’ or ‘‘MeXops’’ number. We consider such numbers to be uninformative at best and misleading at worst. Some servers which are very fast for certain applications can be very slow for others. No single number or small set of numbers are sufficient to characterize how an X implementation will perform on all applications. However, by knowledge of your favorite application, you may be able to use the numbers *x11perf* reports to predict its performance on a given X implementation.

That said, you might also want to look at *x11perfcomp(1)*, a program to compare the outputs of

different *x11perf* runs. You provide a list of files containing results from *x11perf*, and it lays them out in a nice tabular format.

For repeatable results, *x11perf* should be run using a local connection on a freshly-started server. The default configuration runs each test 5 times, in order to see if each trial takes approximately the same amount of time. Strange glitches should be examined; if non-repeatable one might chalk them up to daemons and network traffic. Each trial is run for 5 seconds, in order to reduce random time differences. The number of objects processed per second is displayed to 3 significant digits, but you'll be lucky on most UNIX system if the numbers are actually consistent to 2 digits. *x11perf* moves the cursor out of the test window; you should be careful not to bump the mouse and move it back into the window. (A prize to people who correctly explain why!!).

Before running a test, *x11perf* determines what the round trip time to the server is, and factors this out of the final timing reported. It ensures that the server has actually performed the work requested by fetching a pixel back from the test window, which means that servers talking to graphics accelerators can't claim that they are done, while in the meantime the accelerator is painting madly.

By default *x11perf* automatically calibrates the number of repetitions of each test, so that each should take approximately the same length of time to run across servers of widely differing speeds. However, since each test must be run to completion at least once, some slow servers may take a very long time, particularly on the window moving and resizing tests, and on the arc drawing tests.

All timing reports are for the smallest object involved. For example, the line tests use a PolyLine request to paint several lines at once, but report how many lines per second the server can paint, not how many PolyLine requests per second. Text tests paint a line of characters, but report on the number of characters per second. Some window tests map, unmap, or move a single parent window, but report on how many children windows per second the server can map, unmap, or move.

The current program is mostly the responsibility of Joel McCormack. It is based upon the *x11perf* developed by Phil Karlton, Susan Angebrannt, Chris Kent, Mary Walker, and Todd Newman, who wanted to assess performance differences between various servers. Several tests were added in order to write and tune the PMAX (DECStation 3100) servers. For a general release to the world, *x11perf* was rewritten to ease making comparisons between widely varying machines, to cover most important (and unimportant) X functionality, and to exercise graphics operations in as many different orientations and alignments as possible.

OPTIONS

x11perf is solely Xlib based, and accepts the options listed below:

-display host:dpy

Specifies which display to use.

- sync** Runs the tests in synchronous mode. Normally only useful for debugging *x11perf*.
- pack** Runs rectangle tests so that they pack rectangles right next to each other. This makes it easy to debug server code for stipples and tiles...if the pattern looks ugly, you've got alignment problems.
- repeat <n>** Repeats each test *n* times (by default each test is run 5 times).
- time <s>** Specifies how long in seconds each test should be run (default 5 seconds).
- pause <s>** Specifies how long, in seconds, to pause for between each run.
- all** Runs all tests. This may take a while.
- range <test1>[,<test2>]**
Runs all the tests starting from the specified name *test1* until the name *test2*, including both the specified tests. The testnames should be one of the options starting from *-dot*. (eg) *-range line100* will perform the tests from the 100 pixel line test, and go on till the last test, *-range line100,dline10* will do the tests from *line100* to *dline10*.
- labels** Generates just the descriptive labels for each test specified. See *x11perfcomp* for more details.
- fg *color-or-pixel***
Specifies the foreground color or pixel value to use.
- bg *color-or-pixel***
Specifies the background color or pixel value to use.
- clips *default*** Default number of clip windows.
- ddbg *color-or-pixel***
Specifies the color or pixel value to use for drawing the odd segments of a DoubleDashed line or arc. This will default to the *bg* color.
- rop <rop0 rop1 ...>**
Use specified raster ops (default is *GXcopy*). This option only affects graphics benchmarks in which the graphics function is actually used.

- pm <pm0 pm1 ...>**
Use specified planemasks (default is ~0). This option only affects graphics benchmarks in which the planemask is actually used.
- depth <depth>** Use a visual with <depth> planes per pixel (default is the default visual).
- vclass <vclass>** Use a visual with of class <vclass>. <vclass> can be StaticGray, GrayScale, StaticColor, PseudoColor, TrueColor, or DirectColor. (default is the default visual).
- reps <n>** Specify the repetition count (Default is number that takes approx. 5 seconds)
- subs <s0 s1 ...>** Specify the number of sub windows to use in the Window tests. Default is 4, 16, 25, 50, 75, 100 and 200.
- v1.2** Perform only x11perf Version 1.2 tests using Version 1.2 semantics.
- v1.3** Perform only x11perf Version 1.3 tests using Version 1.3 semantics.
- su** Set the save_under window attribute to True on all windows created by x11perf. Default is False.
- bs <backing_store_hint>**
Set the backing_store window attribute to the given value on all windows created by x11perf. <backing_store_hint> can be WhenMapped or Always. Default is NotUseful.
- dot** Dot.
- rect1** 1x1 solid-filled rectangle.
- rect10** 10x10 solid-filled rectangle.
- rect100** 100x100 solid-filled rectangle.
- rect500** 500x500 solid-filled rectangle.
- srect1** 1x1 transparent stippled rectangle, 8x8 stipple pattern.
- srect10** 10x10 transparent stippled rectangle, 8x8 stipple pattern.

- srect100** 100x100 transparent stippled rectangle, 8x8 stipple pattern.
- srect500** 500x500 transparent stippled rectangle, 8x8 stipple pattern.
- osrect1** 1x1 opaque stippled rectangle, 8x8 stipple pattern.
- osrect10** 10x10 opaque stippled rectangle, 8x8 stipple pattern.
- osrect100** 100x100 opaque stippled rectangle, 8x8 stipple pattern.
- osrect500** 500x500 opaque stippled rectangle, 8x8 stipple pattern.
- tilerect1** 1x1 tiled rectangle, 4x4 tile pattern.
- tilerect10** 10x10 tiled rectangle, 4x4 tile pattern.
- tilerect100** 100x100 tiled rectangle, 4x4 tile pattern.
- tilerect500** 500x500 tiled rectangle, 4x4 tile pattern.
- oddsrect1** 1x1 transparent stippled rectangle, 17x15 stipple pattern.
- oddsrect10** 10x10 transparent stippled rectangle, 17x15 stipple pattern.
- oddsrect100** 100x100 transparent stippled rectangle, 17x15 stipple pattern.
- oddsrect500** 500x500 transparent stippled rectangle, 17x15 stipple pattern.
- oddosrect1** 1x1 opaque stippled rectangle, 17x15 stipple pattern.
- oddosrect10** 10x10 opaque stippled rectangle, 17x15 stipple pattern.
- oddosrect100** 100x100 opaque stippled rectangle, 17x15 stipple pattern.
- oddosrect500** 500x500 opaque stippled rectangle, 17x15 stipple pattern.
- oddtilerect1** 1x1 tiled rectangle, 17x15 tile pattern.
- oddtilerect10** 10x10 tiled rectangle, 17x15 tile pattern.

- oddtile****rect100** 100x100 tiled rectangle, 17x15 tile pattern.
- oddtile****rect500** 500x500 tiled rectangle, 17x15 tile pattern.
- big****srect1** 1x1 stippled rectangle, 161x145 stipple pattern.
- big****srect10** 10x10 stippled rectangle, 161x145 stipple pattern.
- big****srect100** 100x100 stippled rectangle, 161x145 stipple pattern.
- big****srect500** 500x500 stippled rectangle, 161x145 stipple pattern.
- big****osrect1** 1x1 opaque stippled rectangle, 161x145 stipple pattern.
- big****osrect10** 10x10 opaque stippled rectangle, 161x145 stipple pattern.
- big****osrect100** 100x100 opaque stippled rectangle, 161x145 stipple pattern.
- big****osrect500** 500x500 opaque stippled rectangle, 161x145 stipple pattern.
- big****tilerect1** 1x1 tiled rectangle, 161x145 tile pattern.
- big****tilerect10** 10x10 tiled rectangle, 161x145 tile pattern.
- big****tilerect100** 100x100 tiled rectangle, 161x145 tile pattern.
- big****tilerect500** 500x500 tiled rectangle, 161x145 tile pattern.
- eschertilerect1** 1x1 tiled rectangle, 215x208 tile pattern.
- eschertilerect10** 10x10 tiled rectangle, 215x208 tile pattern.
- eschertilerect100**
100x100 tiled rectangle, 215x208 tile pattern.
- eschertilerect500**
500x500 tiled rectangle, 215x208 tile pattern.
- seg1** 1-pixel thin line segment.

-seg10	10-pixel thin line segment.
-seg100	100-pixel thin line segment.
-seg500	500-pixel thin line segment.
-seg100c1	100-pixel thin line segment (1 obscuring rectangle).
-seg100c2	100-pixel thin line segment (2 obscuring rectangles).
-seg100c3	100-pixel thin line segment (3 obscuring rectangles).
-dseg10	10-pixel thin dashed segment (3 on, 2 off).
-dseg100	100-pixel thin dashed segment (3 on, 2 off).
-ddseg100	100-pixel thin double-dashed segment (3 fg, 2 bg).
-hseg10	10-pixel thin horizontal line segment.
-hseg100	100-pixel thin horizontal line segment.
-hseg500	500-pixel thin horizontal line segment.
-vseg10	10-pixel thin vertical line segment.
-vseg100	100-pixel thin vertical line segment.
-vseg500	500-pixel thin vertical line segment.
-whseg10	10-pixel wide horizontal line segment.
-whseg100	100-pixel wide horizontal line segment.
-whseg500	500-pixel wide horizontal line segment.
-wvseg10	10-pixel wide vertical line segment.
-wvseg100	100-pixel wide vertical line segment.

-wvseg500	500-pixel wide vertical line segment.
-line1	1-pixel thin (width 0) line.
-line10	10-pixel thin line.
-line100	100-pixel thin line.
-line500	500-pixel thin line.
-dline10	10-pixel thin dashed line (3 on, 2 off).
-dline100	100-pixel thin dashed line (3 on, 2 off).
-ddline100	100-pixel thin double-dashed line (3 fg, 2 bg).
-wline10	10-pixel line, line width 1.
-wline100	100-pixel line, line width 10.
-wline500	500-pixel line, line width 50.
-wdline100	100-pixel dashed line, line width 10 (30 on, 20 off).
-wddline100	100-pixel double-dashed line, line width 10 (30 fg, 20 bg).
-orect10	10x10 thin rectangle outline.
-orect100	100-pixel thin vertical line segment.
-orect500	500-pixel thin vertical line segment.
-worect10	10x10 wide rectangle outline.
-worect100	100-pixel wide vertical line segment.
-worect500	500-pixel wide vertical line segment.
-circle1	1-pixel diameter thin (line width 0) circle.

- circle10** 10-pixel diameter thin circle.
- circle100** 100-pixel diameter thin circle.
- circle500** 500-pixel diameter thin circle.
- dcircle100** 100-pixel diameter thin dashed circle (3 on, 2 off).
- ddcircle100** 100-pixel diameter thin double-dashed circle (3 fg, 2 bg).
- wcircle10** 10-pixel diameter circle, line width 1.
- wcircle100** 100-pixel diameter circle, line width 10.
- wcircle500** 500-pixel diameter circle, line width 50.
- wdcircle100** 100-pixel diameter dashed circle, line width 10 (30 on, 20 off).
- wddcircle100** 100-pixel diameter double-dashed circle, line width 10 (30 fg, 20 bg).
- pcircle10** 10-pixel diameter thin partial circle, orientation and arc angle evenly distributed.
- pcircle100** 100-pixel diameter thin partial circle.
- wpcircle10** 10-pixel diameter wide partial circle.
- wpcircle100** 100-pixel diameter wide partial circle.
- fcircle1** 1-pixel diameter filled circle.
- fcircle10** 10-pixel diameter filled circle.
- fcircle100** 100-pixel diameter filled circle.
- fcircle500** 500-pixel diameter filled circle.
- fpcircle10** 10-pixel diameter partial filled circle, chord fill, orientation and arc angle evenly distributed.
- fpcircle100** 100-pixel diameter partial filled circle, chord fill.

- fspcircle10** 10-pixel diameter partial filled circle, pie slice fill, orientation and arc angle evenly distributed.
- fspcircle100** 100-pixel diameter partial filled circle, pie slice fill.
- ellipse10** 10-pixel diameter thin (line width 0) ellipse, major and minor axis sizes evenly distributed.
- ellipse100** 100-pixel diameter thin ellipse.
- ellipse500** 500-pixel diameter thin ellipse.
- dellipse100** 100-pixel diameter thin dashed ellipse (3 on, 2 off).
- ddellipse100** 100-pixel diameter thin double-dashed ellipse (3 fg, 2 bg).
- wellipse10** 10-pixel diameter ellipse, line width 1.
- wellipse100** 100-pixel diameter ellipse, line width 10.
- wellipse500** 500-pixel diameter ellipse, line width 50.
- wdellipse100** 100-pixel diameter dashed ellipse, line width 10 (30 on, 20 off).
- wddellipse100** 100-pixel diameter double-dashed ellipse, line width 10 (30 fg, 20 bg).
- pellipse10** 10-pixel diameter thin partial ellipse.
- pellipse100** 100-pixel diameter thin partial ellipse.
- wpellipse10** 10-pixel diameter wide partial ellipse.
- wpellipse100** 100-pixel diameter wide partial ellipse.
- fellipse10** 10-pixel diameter filled ellipse.
- fellipse100** 100-pixel diameter filled ellipse.
- fellipse500** 500-pixel diameter filled ellipse.

- fcpellipse10** 10-pixel diameter partial filled ellipse, chord fill.
- fcpellipse100** 100-pixel diameter partial filled ellipse, chord fill.
- fspellipse10** 10-pixel diameter partial filled ellipse, pie slice fill.
- fspellipse100** 100-pixel diameter partial filled ellipse, pie slice fill.
- triangle1** Fill 1-pixel/side triangle.
- triangle10** Fill 10-pixel/side triangle.
- triangle100** Fill 100-pixel/side triangle.
- trap1** Fill 1x1 trapezoid.
- trap10** Fill 10x10 trapezoid.
- trap100** Fill 100x100 trapezoid.
- trap300** Fill 300x300 trapezoid.
- strap1** Fill 1x1 transparent stippled trapezoid, 8x8 stipple pattern.
- strap10** Fill 10x10 transparent stippled trapezoid, 8x8 stipple pattern.
- strap100** Fill 100x100 transparent stippled trapezoid, 8x8 stipple pattern.
- strap300** Fill 300x300 transparent stippled trapezoid, 8x8 stipple pattern.
- ostrap1** Fill 10x10 opaque stippled trapezoid, 8x8 stipple pattern.
- ostrap10** Fill 10x10 opaque stippled trapezoid, 8x8 stipple pattern.
- ostrap100** Fill 100x100 opaque stippled trapezoid, 8x8 stipple pattern.
- ostrap300** Fill 300x300 opaque stippled trapezoid, 8x8 stipple pattern.
- tiletrap1** Fill 10x10 tiled trapezoid, 4x4 tile pattern.

- tiletrap10** Fill 10x10 tiled trapezoid, 4x4 tile pattern.
- tiletrap100** Fill 100x100 tiled trapezoid, 4x4 tile pattern.
- tiletrap300** Fill 300x300 tiled trapezoid, 4x4 tile pattern.
- oddstrap1** Fill 1x1 transparent stippled trapezoid, 17x15 stipple pattern.
- oddstrap10** Fill 10x10 transparent stippled trapezoid, 17x15 stipple pattern.
- oddstrap100** Fill 100x100 transparent stippled trapezoid, 17x15 stipple pattern.
- oddstrap300** Fill 300x300 transparent stippled trapezoid, 17x15 stipple pattern.
- oddostrap1** Fill 10x10 opaque stippled trapezoid, 17x15 stipple pattern.
- oddostrap10** Fill 10x10 opaque stippled trapezoid, 17x15 stipple pattern.
- oddostrap100** Fill 100x100 opaque stippled trapezoid, 17x15 stipple pattern.
- oddostrap300** Fill 300x300 opaque stippled trapezoid, 17x15 stipple pattern.
- oddtiletrap1** Fill 10x10 tiled trapezoid, 17x15 tile pattern.
- oddtiletrap10** Fill 10x10 tiled trapezoid, 17x15 tile pattern.
- oddtiletrap100** Fill 100x100 tiled trapezoid, 17x15 tile pattern.
- oddtiletrap300** Fill 300x300 tiled trapezoid, 17x15 tile pattern.
- bigstrap1** Fill 1x1 transparent stippled trapezoid, 161x145 stipple pattern.
- bigstrap10** Fill 10x10 transparent stippled trapezoid, 161x145 stipple pattern.
- bigstrap100** Fill 100x100 transparent stippled trapezoid, 161x145 stipple pattern.
- bigstrap300** Fill 300x300 transparent stippled trapezoid, 161x145 stipple pattern.
- bigostrap1** Fill 10x10 opaque stippled trapezoid, 161x145 stipple pattern.

- bigostrap10** Fill 10x10 opaque stippled trapezoid, 161x145 stipple pattern.
- bigostrap100** Fill 100x100 opaque stippled trapezoid, 161x145 stipple pattern.
- bigostrap300** Fill 300x300 opaque stippled trapezoid, 161x145 stipple pattern.
- bigtiletrap1** Fill 10x10 tiled trapezoid, 161x145 tile pattern.
- bigtiletrap10** Fill 10x10 tiled trapezoid, 161x145 tile pattern.
- bigtiletrap100** Fill 100x100 tiled trapezoid, 161x145 tile pattern.
- bigtiletrap300** Fill 300x300 tiled trapezoid, 161x145 tile pattern.
- eschertiletrap1** Fill 1x1 tiled trapezoid, 216x208 tile pattern.
- eschertiletrap10** Fill 10x10 tiled trapezoid, 216x208 tile pattern.
- eschertiletrap100**
Fill 100x100 tiled trapezoid, 216x208 tile pattern.
- eschertiletrap300**
Fill 300x300 tiled trapezoid, 216x208 tile pattern.
- complex10** Fill 10-pixel/side complex polygon.
- complex100** Fill 100-pixel/side complex polygon.
- 64poly10convex**
Fill 10x10 convex 64-gon.
- 64poly100convex**
Fill 100x100 convex 64-gon.
- 64poly10complex**
Fill 10x10 complex 64-gon.
- 64poly100complex**
Fill 100x100 complex 64-gon.

- ftext** Character in 80-char line (6x13).
- f8text** Character in 70-char line (8x13).
- f9text** Character in 60-char line (9x15).
- f14text16** 2-byte character in 40-char line (k14).
- tr10text** Character in 80-char line (Times-Roman 10).
- tr24text** Character in 30-char line (Times-Roman 24).
- polytext** Character in 20/40/20 line (6x13, Times-Roman 10, 6x13).
- polytext16** 2-byte character in 7/14/7 line (k14, k24).
- fitext** Character in 80-char image line (6x13).
- f8itext** Character in 70-char image line (8x13).
- f9itext** Character in 60-char image line (9x15).
- f14itext16** 2-byte character in 40-char image line (k14).
- f24itext16** 2-byte character in 23-char image line (k24).
- tr10itext** Character in 80-char image line (Times-Roman 10).
- tr24itext** Character in 30-char image line (Times-Roman 24).
- scroll10** Scroll 10x10 pixels vertically.
- scroll100** Scroll 100x100 pixels vertically.
- scroll500** Scroll 500x500 pixels vertically.
- copywinwin10** Copy 10x10 square from window to window.
- copywinwin100** Copy 100x100 square from window to window.

- copywinwin500** Copy 500x500 square from window to window.
- copypixwin10** Copy 10x10 square from pixmap to window.
- copypixwin100** Copy 100x100 square from pixmap to window.
- copypixwin500** Copy 500x500 square from pixmap to window.
- copywinpix10** Copy 10x10 square from window to pixmap.
- copywinpix100** Copy 100x100 square from window to pixmap.
- copywinpix500** Copy 500x500 square from window to pixmap.
- copypixpix10** Copy 10x10 square from pixmap to pixmap.
- copypixpix100** Copy 100x100 square from pixmap to pixmap.
- copypixpix500** Copy 500x500 square from pixmap to pixmap.
- copyplane10** Copy 10x10 1-bit deep plane.
- copyplane100** Copy 100x100 1-bit deep plane.
- copyplane500** Copy 500x500 1-bit deep plane.
- putimage10** PutImage 10x10 square.
- putimage100** PutImage 100x100 square.
- putimage500** PutImage 500x500 square.
- putimagexy10** PutImage XY format 10x10 square.
- putimagexy100** PutImage XY format 100x100 square.
- putimagexy500** PutImage XY format 500x500 square.
- shmput10** PutImage 10x10 square, MIT shared memory extension.

- shmput100** PutImage 100x100 square, MIT shared memory extension.
- shmput500** PutImage 500x500 square, MIT shared memory extension.
- shmputxy10** PutImage XY format 10x10 square, MIT shared memory extension.
- shmputxy100** PutImage XY format 100x100 square, MIT shared memory extension.
- shmputxy500** PutImage XY format 500x500 square, MIT shared memory extension.
- getimage10** GetImage 10x10 square.
- getimage100** GetImage 100x100 square.
- getimage500** GetImage 500x500 square.
- getimagexy10** GetImage XY format 10x10 square.
- getimagexy100** GetImage XY format 100x100 square.
- getimagexy500** GetImage XY format 500x500 square.
- compwinwin10** Composite 10x10 from window to window.
- compwinwin100**
Composite 100x100 from window to window.
- compwinwin500**
Composite 500x500 from window to window.
- comppixwin10** Composite 10x10 from pixmap to window.
- comppixwin100**
Composite 100x100 from pixmap to window.
- comppixwin500**
Composite 500x500 from pixmap to window.
- magpixwin10** Scale 5x5 from pixmap to 10x10 window.

- magpixwin100** Scale 50x50 from pixmap to 100x100 window.
- magpixwin500** Scale 250x250 from pixmap to 500x500 window.
- minpixwin10** Scale 10x10 from pixmap to 5x5 window.
- minpixwin100** Scale 100x100 from pixmap to 50x50 window.
- minpixwin500** Scale 500x500 from pixmap to 250x250 window.
- noop** X protocol NoOperation.
- atom** GetAtomName.
- pointer** QueryPointer.
- prop** GetProperty.
- gc** Change graphics context.
- create** Create child window and map using MapSubwindows.
- ucreate** Create unmapped window.
- map** Map child window via MapWindow on parent.
- unmap** Unmap child window via UnmapWindow on parent.
- destroy** Destroy child window via DestroyWindow parent.
- popup** Hide/expose window via Map/Unmap popup window.
- move** Move window.
- umove** Moved unmapped window.
- movetree** Move window via MoveWindow on parent.
- resize** Resize window.

- uresize** Resize unmapped window.
- circulate** Circulate lowest window to top.
- ucirculate** Circulate unmapped window to top.

X DEFAULTS

There are no X defaults used by this program.

SEE ALSO

X(7), x11perfcomp(1)

AUTHORS

Joel McCormack
Phil Karlton
Susan Angebrannt
Chris Kent
Keith Packard
Graeme Gill