HDSshare Operation

When HDSshare starts, it puts up a window with a name tile for the local display and for each remote ViewStation display you’ve specified. You can use the keyboard and mouse for any of those displays just by moving your local mouse to the appropriate tile and clicking on it.

This moves the mouse and keyboard control to the remote ViewStation’s monitor screen (which may be on your desktop or may be viewable through a clean-room window, etc.). It is just as if the keyboard and mouse were connected to the remote ViewStation base. It is not necessary for the remote display to have either a keyboard or a mouse; if they are present, the HDSshare display overrides their function when it has control. The HDSshare window with its name tiles appears on the new display so you can go to any other HDSshare ViewStation from the new display.

If you use the -nobuttons command line option, HDSshare doesn’t put up tiles for each display but allows you to move the cursor smoothly from one display to another as if they were a single screen. If you use the -nobuttons option, you must use the -md option to locate the master display; the other displays are assumed to be left-to-right as they appear on the command line.

For example the command line:

```
hdsshare -nobuttons -rd slave1:0 -rd slave2:0 -md -rd
slave3:0&
```

specifies the physical arrangement as slave1, slave2, master, slave3. In this example you move the cursor from one display to another without use of the name tiles.

2.6.19 Xpmroot Manual Page

NAME

xpmroot - Sets the root window of the current X display to an Xpm pixmap

SYNOPSIS

xpmroot file

DESCRIPTION

xpmroot reads the Xpm file specified in the command line and displays it in the root window.

BUGS

Repeated use of xpmroot with different xpm pixmaps will use up slots in your color table pretty darn fast.

AUTHOR

Rob Nation

ACKNOWLEDGMENTS

Similar to functionality of “xhydra” by Robert Chesler, but developed independently before the author was aware of xhydra.

SEE ALSO

X(1), xtrap(1)

2.6.18.1 Using the HDSshare Client

HDSshare is a local ViewStation client that lets you share a keyboard and mouse between several ViewStations. This is sometimes convenient where ViewStations are used without their own input devices (such as process control installations or in isolated environments such as clean rooms, etc.).

Starting HDSshare

HDSshare is included with the HDSperx clients in the HDSware 3.2 distribution but is not included as a default client on the HDSperx menu. You must make two entries to the clients configuration file in order to run HDSshare.

Make these changes in the ./hds-fx/config/clients.config file:

1. Load the Xtrap downloadable extension:

Your line should look like this (this may vary slightly by HDSware version)

```
hds.serverExtensions: /FXUtilities/hds-fx/extensions/MIT-SHM
/FXUtilities/hds-fx/extensions/DEC-XTRAP
```
(use a single line for the extensions, use a space between entries)

2. Add a line for the HDSshare client:

```
hds.client.hdsshare.path: /FXUtilities/hds-fx/bin/hdsshare
```

3. Reboot ALL of the HDSshare ViewStations so the server extension is loaded on each.

Refer to the hdsshare man page for details of HDSshare operation.

HDSshare requires that the Xtrap extension be running on all of the ViewStations controlled by HDSshare, so they should all use this clients.config file and all be rebooted with the extension.

Start HDSshare from the ViewStation’s Console window

You must specify one or more remote displays for HDSshare to manage. Suppose your ViewStation is named “hds40” and you want to control ViewStations named “hds42” and “hds46”. In the Console window enter:

```
hdsshare -rd hds42:0 -rd hds46:0 &
```

which gives you three name tiles (hds40, hds42, and hds46) in the HDSshare window. Note that HDSshare should run in the background.

The local ViewStation is the default controller; you can also specify another ViewStation as the controller with the -display argument:

```
hdsshare -display local:0 -rd remote1:0 -rd remote2:0 -rd
remote3:0 &
```

2.6.18 HDSshare Manual Page

This client is available with HDSware version 3.2. You must use the special configuration files /HDSware/hds-fx/config/config.ext (or /HDSware/hds-fx/config/config.ext.dsk) and /HDSware/hds-fx/config/clients.fx in order the hdsshare client.

NAME

hdsshare - a simple multi-headed ViewStation

SYNTAX

hdsshare [toolkit options] [-nobuttons] [-md] -rd RemoteDisplay [-rd RemoteDisplay ...]

DESCRIPTION

This Motif utility is used to logically control multiple X servers from one X server's keyboard and mouse, and switch between X servers. This utility requires the XTrap extension to be running on both the master X server and on all remote (slave) X servers.

OPTIONS

-no buttons

Do not use Motif buttons to hop the cursor from one display to another. If this option is specified, the -md option must also be present to identify the left-right order of the displays.

-md

This option enables automatic cursor hopping by moving the cursor to the left or right edge of the screen. The displays are assumed to be physically ordered from left to right in the same order that they are given on the command line. The -md option is a place holder for the master display's position in the display order. For example,

```
hdsshare -rd slave1:0 -rd slave2:0 -md -rd slave:0
```

specifies that the order of the display screens, from left to right, is slave1, slave2, the master display, and then slave3.

-rd RemoteDisplay

Specified in the same manner as the control display, for each remote display.

Example usage

```
hdsshare -rd remote:0
hdsshare -display local:0 -rd remote:0 -md
hdsshare -nobuttons -md -rd near:0 -rd far:0
```

Verbose output. ICMP packets other than ECHO_RESPONSE that are received are listed.

When using ping for fault isolation, it should first be run on the local host, to verify that the local network interface is up and running. Then, hosts and gateways further and further away should be “pinged”. Round-trip times and packet loss statistics are computed. If duplicate packets are received, they are not included in the packet loss calculation, although the round trip time of these packets is used in calculating the minimum/average/maximum round-trip time numbers. When the specified number of packets have been sent (and received) or if the program is terminated with a SIGINT, a brief summary is displayed.

Refer also to Section 1.4.5.3 for information about the ViewStation’s local route command.

2.6.17 PING Manual Page

Ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams ("pings") have an IP and ICMP header, followed by a "struct timeval" and then an arbitrary number of "pad" bytes used to fill out the packet. The options are as follows:

-c count

Stop after sending (and receiving) count ECHO_RESPONSE packets.

-d

Set the SO_DEBUG option on the socket being used.

-i wait

Wait wait seconds between sending each packet. The default is to wait for one second between each packet.

-l preload

If preload is specified, ping sends that many packets as fast as possible before falling into its normal mode of behavior.

-n

Numeric output only. No attempt will be made to lookup symbolic names for host addresses.

-p pattern

You may specify up to 16 "pad" bytes to fill out the packet you send. This is useful for diagnosing data-dependent problems in a network. For example, "-p ff" will cause the sent packet to be filled with all ones.

-q

Quiet output. Nothing is displayed except the summary lines at startup time and when finished.

-r

Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it (e.g., after the interface was dropped by routed(8)).

-s packetsize

Specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

-v

Determines the host that xscope will use to find its X11 server.

-d<display>

Defines the display number. The display number is added to the input and output port to give the actual ports which are used by xscope.

-q

Quiet output mode. Gives only the names of requests, replies, errors, and events, but does not indicate contents.

-v<print-level>

Determines the level of printing which xscope will provide. The print-level can be 0 (same as quiet mode), 1, 2, 3, 4. The larger numbers give more and more output. For example, a successful setup returns a string which is the name of the vendor of the X11 server. At level 1, the explicit field giving the length of the string is suppressed since it can be inferred from the string. At level 2 and above the length is explicitly printed.

EXAMPLES

xscope -v4 -hcleo -d0 -o0 -i1

This command would have xscope communicate with an X11 server on host ‘‘cleo’’, display 0; xscope itself would be available on the current host as display 1 (display of 0 plus the 1 of -i1). Verbose level 4.

xscope -q -d1 -o1 -o3

The X11 server for the current host, display 2 (1 for -d1 plus 1 for -o1) would be used by xscope which would run as display 4 (1 for -d1 plus 3 for -o3). Quiet mode (verbose level 0).

SEE ALSO

X(1), X11 Protocol document (doc/Protocol/X11.protocol)

2.6.16 Xscope Manual Page

This is the manual page for xscope. It runs as a local client on the ViewStation.

NAME

xscope - X Window Protocol Viewer

SYNOPSIS

xscope [option] ...

DESCRIPTION

xscope sits in-between an X11 client and an X11 server and prints the contents of each request, reply, error, or event that is communicated between them. This information can be useful in debugging and performance tuning of X11 servers and clients.

To operate, xscope must know the host, port, and display to use to connect to the X11 server. In addition, it must know the port on which it should listen for X11 clients.

Two cases are common:

(1) The X11 server is on the same host as xscope. In this case, the input port for xscope should be selected as an X11 server on a different display, and the client DISPLAY argument adjusted to select xscope. For example, if the X11 server is on port 6000, display 1, then xscope can use port 6002 as its input port. The client can use display 1 for direct access to X11 or display 2 for access to xscope. Note that running as a ViewStation local client xscope is on the same host (the Viewstation) as the server. The default host, input port, and output ports are already set correctly, so the user need only set the verbosity level desired. To connect to xscope the user would then set his DISPLAY environment variable to ViewStation IP:1 before running the client they wish to test.

(2) The X11 server is on a different host than xscope. In this case the same input and output ports can be used, and the host component of the DISPLAY is used to select xscope or X11.

ARGUMENTS

-i<input-port>

Specify the port that xscope will use to take requests from clients (defaults to 1). For X11, this port is automatically biased by 6000.

-o<output-port>

Determines the port that xscope will use to connect to X11 (defaults to 0). For X11, this port is automatically biased by 6000.

-h<host>

basePixel	16

Color 0

Index	Type	Red	Green	Blue
0	RgbFloat	0.0	0.0	0.0
1	RgbFloat	1.0	1.0	1.0
2	RgbFloat	1.0	0.0	0.0
3	RgbFloat	0.0	1.0	0.0
4	RgbFloat	0.0	0.0	1.0
5	RgbFloat	1.0	1.0	0.0
6	RgbFloat	0.0	1.0	1.0
7	RgbFloat	1.0	0.0	1.0

Light	1	
lightType		Ambient
direction		(0.0,0.0,0.0)
point		(0.0,0.0,0.0)
concentration		0.0
spreadAngle		0.0
attenuation1		0.0
attenuation2		0.0
lightColor type		RgbFloat
lightColor red		1.0
lightColor green		1.0
lightColor blue		1.0
DepthCue	0	
mode		Off
frontPlane		1.0
backPlane		0.0
frontScaling		1.0
backScaling		0.5
depthCueColor type		indexed
depthCueColor index		1
Color Approximation	0	
approxType		ColorSpace
approxModel		ColorApproxRGB
max1		5
max2		5
max3		5
dither		Off
mult1		1
mult2		6
mult3		36
weight1		1.0
weight2		1.0
weight3		1.0

bfSurfaceColor type		indexed
bfSurfaceColor index		1
reflectionAttr.ambient		1.0
reflectionAttr.diffuse		1.0
reflectionAttr.specular		1.0
reflectionAttr.specularConc		0.0
reflectionAttr.transmission		0.0
reflectionAttr.specularColor type		indexed
reflectionAttr.specularColor index		1
bfreflectionAttr.ambient		1.0
bfreflectionAttr.diffuse		1.0
bfreflectionAttr.specular		1.0
bfreflectionAttr.specularConc		0.0
bfreflectionAttr.transmission		0.0
bfreflectionAttr.specularColor type		indexed
bfreflectionAttr.specularColor ind		1
Edge Bundle	1	
edges		off
edgeType		SurfaceEdgeSolid
edgeWidth		1.0
edgeColor type		indexed
edgeColor index		1
TextFont	1	
numFonts		1
font		defaultFont (Roman)
(16 character set id's allowed per Font group)		
View	0	
clipFlags		ClipXY|ClipBack|ClipFront
clipLimits		(0.0,0.0,0.0),(1.0,1.0,1.0)
orientation		identity matrix
mapping		identity matrix

Line Type		1
polyline_interp		1
curve_approx		1
curve_tolerance		1.0
Line Width		1
line_color type		indexed
line_color index		1
MarkerBundle	0	
marker_type		asterisk
marker_scale		1.0
marker_color type		indexed
marker_color index		1
TextBundle	1	
textFontIndex		1
textPrecision		StringPrecision
charExpansion		1.0
charSpacing		0.0
textColor type		indexed
textColor index		1
InteriorBundle	1	
interiorStyle		InteriorStyleHollow
interiorStyleIndex		1
reflectionModel		ReflectionNoShading
surfaceInterp		SurfaceInterpNone
bfInteriorStyle		InteriorStyleHollow
bfInteriorStyleIndex		1
bfReflectionModel		ReflectionNoShading
surfaceApprox.approxMethod		1
surfaceApprox.uTolerance		1.0
surfaceApprox.vTolerance		1.0
surfaceColor type		indexed
surfaceColor index		1

Maximum Trim Curve Order	6
Best Color Approximation Values	ColorApproxAnyValues
Transparency support	False
Double Buffering support	True
Max Hits Event Support	True
ChromaticityRedU	0.628
ChromaticityRedV	0.346
LuminanceRed	1.0
ChromaticityGreenU	0.268
ChromaticityGreenV	0.588
LuminanceGreen	1.0
ChromaticityBlueU	0.150
ChromaticityBlueV	0.070
LuminanceBLue	1.0
ChromaticityWhiteU	0.313
ChromaticityWhiteV	0.329
LuminanceWhite	1.0

2.6.15.5 HDS PEX-3D Lookup Tables

Name	Maximum # of Entries	# Predefined
Line Bundle	20	1
Marker Bundle	20	1
TextBundle	20	1
Interior Bundle	20	1
Edge Bundle	20	1
Pattern	0	0
Color	256	8
TextFont	4	1
View	6	1
Light	16	1
Depth Cue	6	1
ColorApprox	6	1

2.6.15.6 HDS PEX-3D Lookup Table default predefined values

Name	Index	Values
LineBundle	0	

   3. VisualizeNone
   4. SimulateSome
   5. VisualizeWhenever
- Color Approximation Types - there are two types supported.
   1. ColorSpace
   2. ColorRange
- Color Approximation Models - there is one model supported.
   1. RGB
- Escapes - one escape is supported.
   1. “SetEchoColor”
- Rendering Color Models - there is one model supported.
   1. RGB
- Parametric Surface Characteristics - there are three types supported.
   1. None
   2. Implementation Dependent - HDS PEX-3D utilizes IsoparametricCurves with the number of u and v curves set to 20.
   3. IsoparametricCurves

2.6.15.4 HDS PEX-3D Implementation Dependent Constants

Name	**Value**
Dithering	True
Maximum Edge Width	~((unsigned long)0)
Minimum Edge Width	1
Nominal Edge Width	1
Supported Edge Widths	~((unsigned long)0)
Maximum Line Width	~((unsigned long)0)
Minimum Line Width	1
Nominal Line Width	1
Supported Line Widths	~((unsigned long)0)
Maximum Marker Size	~((unsigned long)0)
Minimum Marker Size	1
Nominal Marker Size	1
Supported Marker Sizes	~((unsigned long)0)
Model Clip Planes	64
Name Set Names	256
Max Non Ambient Lights	64
Maximum Nurb Order	10

2. Color

- Surface Approximation Methods - there are 6 types supported.
  1. Implementation Dependent - HDS PEX-3D utilizes the ConstantBetweenKnots method with u and v tolerances of 10.0
  2. ConstantBetweenKnots
  3. WCS_ChordalSize
  4. NPC_ChordalSize
  5. WCS_ChordalDev
  6. NPC_ChordalDev
- Trim Curve Approximation Methods - there are 2 methods supported.
  1. Implementation Dependent - HDS PEX-3D utilizes the ConstantBetweenKnots method with a tolerance of 10.0
  2. ConstantBetweenKnots
- Model Clip Operator - there are two types supported.
  1. Replace
  2. Intersection
- Light Types - there are 4 types supported.
  1. Ambient
  2. WCS_Vector
  3. WCS_Point
  4. WCS_Spot
- Color Type - there are two types supported.
  1. Indexed
  2. RGBFloat
- Float Format - there are two formats supported.
  1. IEEE_754_32
  2. DEC_F_Floating
- HLHSR modes - there are 2 modes supported.
  1. Off
  2. ZBuffer
- Prompt Echo Type - there is one type supported.
  1. EchoPrimitive.
- Display Update Modes - there are 5 modes supported.
  1. VisualizeEach
  2. VisualizeEasy

- Interior Style - there are three types supported.
    1. Hollow
    2. Solid
    3. Empty
- Line Type - there are four types supported.
    1. Solid
    2. Dashed
    3. Dotted
    4. DashDot
- Pick Device Type - there are two types supported.
    1. DC_HitBox
    2. NPC_HitVolume
- Pick One Methods - there are two types supported.
    1. 1
    2. Last
- Pick All Methods - there are two types supported.
    1. 1
    2. All
- Polyline Interpolation Methods - there are 2 types supported.
    1. None
    2. Color
- Curve Approximation Method - there are 6 types supported.
    1. Implementation Dependent - HDS PEX-3D utilizes the ConstantBetweenKnots method with a tolerance of 10.0
    2. ConstantBetweenKnots
    3. WCS_ChordalSize
    4. NPC_ChordalSize
    5. WCS_ChordalDev
    6. NPC_ChordalDev
- Reflection Methods - there are four types supported.
    1. NoShading
    2. Ambient
    3. Diffuse
    4. Specular
- Surface Interpolation Methods - there are two types supported.
    1. None

- PickOne methods “ClosestZ”, “VisibleAny”, and “VisibleClosest” are not supported. These methods will default to “Last”.
- PickAll method “Visible” is not supported. This method will default to “All”.
- Curve approximation methods “DC_ChordalSize”, “DC_ChordalDev”, “WCS_Relative”, “NPC_Relative”, and “DC_Relative” are not supported. These methods are rendered as Implementation Dependent which defaults to ConstantBetweenKnots with a tolerance of 10.0.
- Surface interpolation methods “DotProduct” and “Normal” are not supported. These methods default to the “Color” (Gouraud).
- Surface approximation methods “DC_ChordalSize”, “DC_ChordalDev”, “WCS_Relative”, “NPC_Relative”, and “DC_Relative” are not supported. These methods are rendered as Implementation Dependent which defaults to ConstantBetweenKnots with a tolerance of 10.0 for u and v.
- Color types “CIEFloat”, “HDSVFloat”, “HLSFloat”, “RGBInt8”, and “RGBInt16” are not supported. Attempts to use these types will result in a ColorTypeError.
- Float formats “IEEE_754_64”, and “DEC_D_Floating” are not supported.
- HLHSR modes “ZBufferId”, “HiddenLineOnly”, “Scanline”, and “Painters” are not supported. These modes default to “Off”.
- Prompt echo types “EchoStructure” and “EchoNetwork” are not supported. The types default to “EchoPrimitive”.
- Color approximation types “CIE”, “HSV”, “HLS” and “YIQ” are not supported. Attempts to use these types will exit with a BadValue.
- “gdp”, “gdp3”, and “gse” types are not supported.
- Rendering color models “CIE”, “HSV”, and “HLS” are not supported. Requests to use these models will silently exit.
- Parametric surface characteristics “MC_LevelCurves”, and “WC_Levelcurves” are not supported. These types will default to “None”.
- Transparency is not supported.

2.6.15.3 HDS PEX-3D Functional Description

These are the HDS PEX-3D supported Enumerated Types

- Marker Type - there are 5 marker types supported.
    1. Dot
    2. Cross
    3. Asterisk
    4. Circle
    5. X
- Annotation text style - there are two types supported.
    1. Not Connected
    2. Connected

2.6.15 PEX 3D Server Extension

This section describes the PEX-3D server extension. Note that this is an extension of the server code rather than a local client. This extension is loaded with entries in the clients.config file, described in Section 2.1.6.

PEX-3D Extension V1.0

Introduction

This extension is based on the X Consortium PEX 5.1 protocol standard.

The PEX extension is a 3-D graphics extension to the HDSware X11 server.

2.6.15.1 Installation of the HDS PEX-3D Extension

The PEX extension is a downloaded extension controlled by the "hds.serverExtensions" resource in the */FXUtilities/hds-fx/config/clients.config* file. There are two files, named *X3D-PEX-Vnn* and *X3D-PEX-Vnn-D*, which are located in */FXUtilities/hds-fx/extensions*. These files are for use with ViewStation server versions Vnn.hds and Vnn-D.hds respectively. These two versions of the ViewStation's server code differ in their ability to download local clients (designated by the -D option). The appropriate PEX-3D extension files can be used with either version since these extensions are loaded as an integral part of the server code.

HDS PEX-3D utilizes the PEX_FONTPATH environment variable to set the PEX font path which is separate from the X11 font path. The default value is set to */FXUtilities/hds-fx/fonts/PEX* which contains two compressed PEX fonts, Roman and Roman_m.

The PEX-3D extension requires the NFS file transfer protocol in order to read fonts. In addition the extension needs to read the Roman font in order to initialize. This means that the user would need to set the HDSware Host Access Protocol to NFS prior to starting the server.

ViewStation Requirements

The PEX-3D extensions are available with the HDSware 3.1.1.

HDS PEX-3D is supported on all Viewstation architectures. While it is possible to load the HDS PEX-3D extension with the Vnn.hds (non-downloadable) server version into a ViewStation containing only 4 MB of memory there is only about 500 KB of memory left. It is recommended that at least 8MB of memory be installed on the ViewStation when using the PEX-3D extension.

2.6.15.2 HDS PEX-3D Limitations

- Interior styles "Pattern" and "Hatch" are not supported. These styles will be rendered using the "Solid" interior style.

AUTHOR

Bob Scheifler, MIT Laboratory for Computer Science
David Krikorian, MIT Project Athena (X11 version)

2.6.14 FDFORMAT Disk Format Utility

This is the manual page for the ViewStation's floppy disk formatting utility.

NAME

fdformat - format floppy diskettes

SYNOPSIS

fdformat [-l] [-n] [-v label] device

DESCRIPTION

fdformat is a utility for formatting floppy diskettes. Al new blank diskettes must be formatted before they can be used. fdformat formats and verifies each track on the diskette, and terminates if it finds any bad sectors. All existing data on the diskette, if any, is destroyed by formatting.

By default, fdformat formats a high density diskette with a capacity of 1.44 megabytes. Use the -l option to format low density diskettes (720K capacity). Note: it is not possible to put a low density format onto a high density floppy diskette. High density diskettes can be recognized by the high density detect hole in the lower right corner of the diskette, opposite the write protect hole in the lower left corner.

A MS-DOS file system is installed on the diskette after the format is done.

OPTIONS

-v	Put a MS-DOS label on the disk after formatting it.
-l	Format a low density (720K) diskette.
-n	Skip verifying the floppy diskette after formatting.

NOTES

The MS-DOS file system installed by fdformat does not have the necessary system files to make it bootable on a PC.

FILES

/dev/fd1

/dev/fd2

etc.

BUGS

Currently bad sector mapping is not supported on floppy diskettes. Therefore, a diskette is unusable if fdformat finds an error (bad sector).

one is given, it will be interpreted as the acceleration. If no parameters or the flag ‘default’ is used, the system defaults will be set.

p

The p option controls pixel color values. The parameters are the color map entry number in decimal, and a color specification. The root background colors may be changed on some servers by altering the entries for BlackPixel and WhitePixel. Although these are often 0 and 1, they need not be. Also, a server may choose to allocate those colors privately, in which case an error will be generated. The map entry must not be a read-only color, or an error will result.

r

The r option controls the autorepeat. If a preceding dash or the ‘off’ flag is used, autorepeat will be disabled. If no parameters or the ‘on’ flag is used, autorepeat will be enabled.

s

The s option lets you set the screen saver parameters. This option accepts up to two numerical parameters, a ‘blank/noblank’ flag, an ‘expose/noexpose’ flag, an ‘on/off’ flag, or the ‘default’ flag. If no parameters or the ‘default’ flag is used, the system will be set to its default screen saver characteristics. The ‘on/off’ flags simply turn the screen saver functions on or off. The ‘blank’ flag sets the preference to blank the video (if the hardware can do so) rather than display a background pattern, while ‘noblank’ sets the preference to display a pattern rather than blank the video. The ‘expose’ flag sets the preference to allow window exposures (the server can freely discard window contents), while ‘noexpose’ sets the preference to disable screen saver unless the server can regenerate the screens without causing exposure events. The length and period parameters for the screen saver function determines how long the server must be inactive for screen saving to activate, and the period to change the background pattern to avoid burn in. The arguments are specified in seconds. If only one numerical parameter is given, it will be used for the length.

q

The q option gives you information on the current settings.

These settings will be reset to default values when you log out.

Note that not all X implementations are guaranteed to honor all of these options.

SEE ALSO

X(1), Xserver(1), xmodmap(1), xrdb(1), xsetroot(1)

COPYRIGHT

c

The c option controls key click. This option can take an optional value, a preceding dash(-), or an 'on/off' flag. If no parameter or the 'on' flag is given, the system defaults will be used. If the dash or 'off' flag is used, keyclick will be disabled. If a value from 0 to 100 is given, it is used to indicate volume, as a percentage of the maximum. The X server will set the volume to the nearest value that the hardware can support.

fp= path,...

The fp= sets the font path to the directories given in the path argument. The directories are interpreted by the server, not by the client, and are server-dependent. Directories that do not contain font databases created by mkfontdir will be ignored by the server.

fp default

The default argument causes the font path to be reset to the server's default.

fp rehash

The rehash argument causes the server to reread the font databases in the current font path. This is generally only used when adding new fonts to a font directory (after running mkfontdir to recreate the font database).

-fp or fp-

The -fp and fp- options remove elements from the current font path. They must be followed by a comma-separated list of directories.

+fp or fp+

This +fp and fp+ options prepend and append elements to the current font path, respectively. They must be followed by a comma-separated list of directories.

led

The led option controls the keyboard LEDs. This controls the turning on or off of one or all of the LEDs. It accepts an optional integer, a preceding dash(-) or an 'on' flag is given, all LEDs are turned on. If a preceding dash or the flag 'off' is given, all LEDs are turned off. If a value between 1 and 32 is given, that LED will be turned on or off depending on the existence of a preceding dash. A common LED which can be controlled is the "Caps Lock" LED. "xset led 3" would turn led #3 on. "xset -led 3" would turn it off. The particular LED values may refer to different LEDs on different hardware.

m

The m option controls the mouse parameters. The parameters for the mouse are 'acceleration' and 'threshold'. The acceleration can be specified as an integer, or as a simple fraction. The mouse, or whatever pointer the machine is connected to, will go 'acceleration' times as fast when it travels more than 'threshold' pixels in a short time. This way, the mouse can be used for precise alignment when it is moved slowly, yet it can be set to travel across the screen in a flick of the wrist when desired. One or both parameters for the m option can be omitted, but if only

Jim Fulton, MIT X Consortium, rewritten from an earlier version by David Rosenthal of Sun Microsystems.

2.6.13 Xset Manual Page

This is the ViewStation xset local client man page.

NAME

xset - user preference utility for X

SYNOPSIS

xset [-display display] [-b] [b on/off] [b [volume [pitch [duration]]] [[-]bc] [-c] [c on/off] [c [volume]] [[-+]fp[-+=] path[,path[,...]]] [fp default] [fp rehash] [[-]led [integer]] [led on/off] [m[ouse] [accel_mult[/accel_div] [threshold]]] [m[ouse] default] [p pixel color] [[-]r] [r on/off] [s [length [period]]] [s blank/noblank] [s expose/noexpose] [s on/off] [s default] [q]

DESCRIPTION

This program is used to set various user preference options of the display.

OPTIONS

-display display

This option specifies the server to use; see X(1).

b

The b option controls bell volume, pitch and duration. This option accepts up to three numerical parameters, a preceding dash(-), or a 'on/off' flag. If no parameters are given, or the 'on' flag is used, the system defaults will be used. If the dash or 'off' are given, the bell will be turned off. If only one numerical parameter is given, the bell volume will be set to that value, as a percentage of its maximum. Likewise, the second numerical parameter specifies the bell pitch, in hertz, and the third numerical parameter specifies the duration in milliseconds. Note that not all hardware can vary the bell characteristics. The X server will set the characteristics of the bell as closely as it can to the user's specifications.

bc

The bc option controls bug compatibility mode in the server, if possible; a preceding dash(-) disables the mode, otherwise the mode is enabled. Various pre-R4 clients pass illegal values in some protocol requests, and pre-R4 servers did not correctly generate errors in these cases. Such clients, when run against an R4 server, will terminate abnormally or otherwise fail to operate correctly. Bug compatibility mode explicitly reintroduces certain bugs into the X server, so that many such clients can still be run. This mode should be used with care; new application development should be done with this mode disabled. The server must support the MIT- SUNDRY-NONSTANDARD protocol extension in order for this option to work.

```
!
! 101 Backspace
! 55 Caps
! 14 Ctrl
! 15 Break/Reset
! 86 Stop
! 89 F5
!
keycode 101 = Delete
keycode 55 = Control_R
clear Lock
add Control = Control_R
keycode 89 = Escape
keycode 15 = Caps_Lock
add Lock = Caps_Lock
```

ENVIRONMENT

DISPLAY to get default host and display number.

SEE ALSO

X(1), xev(1), Xlib documentation on key and pointer events

BUGS

Every time a keycode expression is evaluated, the server generates a MappingNotify event on every client. This can cause some thrashing. All of the changes should be batched together and done at once. Clients that receive keyboard input and ignore MappingNotify events will not notice any changes made to keyboard mappings.

Xmodmap should generate "add" and "remove" expressions automatically whenever a keycode that is already bound to a modifier is changed.

There should be a way to have the remove expression accept keycodes as well as keysyms for those times when you really mess up your mappings.

COPYRIGHT

Copyright 1988, 1989, 1990 Massachusetts Institute of Technology.

Copyright 1987 Sun Microsystems, Inc.

See X(1) for a full statement of rights and permissions.

AUTHOR

resource in xterm is set as well, all terminal emulator windows will use the same key for erasing characters:

```
% xmodmap -e “keysym BackSpace = Delete”

% echo “XTerm*ttyModes: erase ^?” | xrdb -merge
```

Some keyboards do not automatically generate less than and greater than characters when the comma and period keys are shifted. This can be remedied with xmodmap by resetting the bindings for the comma and period with the following scripts:

```
!

! make shift-, be < and shift-. be >

!

keysym comma = comma less

keysym period = period greater
```

One of the more irritating differences between keyboards is the location of the Control and Shift Lock keys. A common use of xmodmap is to swap these two keys as follows:

```
!

! Swap Caps_Lock and Control_L

!

remove Lock = Caps_Lock

remove Control = Control_L

keysym Control_L = Caps_Lock

keysym Caps_Lock = Control_L

add Lock = Caps_Lock

add Control = Control_L
```

The keycode command is useful for assigning the same keysym to multiple keycodes. Although unportable, it also makes it possible to write scripts that can reset the keyboard to a known state. The following script sets the backspace key to generate Delete (as shown above), flushes all existing caps lock bindings, makes the CapsLock key be a control key, make F5 generate Escape, and makes Break/Reset be a shift lock.

```
!

! On the HP, the following keycodes have key caps as listed:
```

This adds all keys containing the given keysyms to the indicated modifier map. The keysym names are evaluated after all input expressions are read to make it easy to write expressions to swap keys (see the EXAMPLES section).

```
remove MODIFIERNAME = KEYSYMNAME ...
```

This removes all keys containing the given keysyms from the indicated modifier map. Unlike add, the keysym names are evaluated as the line is read in. This allows you to remove keys from a modifier without having to worry about whether or not they have been reassigned.

```
pointer = default
```

This sets the pointer map back to its default settings (button 1 generates a code of 1, button 2 generates a 2, etc.).

```
pointer = NUMBER ...
```

This sets to pointer map to contain the indicated button codes. The list always starts with the first physical button.

Lines that begin with an exclamation point (!) are taken as comments.

If you want to change the binding of a modifier key, you must also remove it from the appropriate modifier map.

EXAMPLES

Many pointers are designed such that the first button is pressed using the index finger of the right hand. People who are left-handed frequently find that it is more comfortable to reverse the button codes that get generated so that the primary button is pressed using the index finger of the left hand. This could be done on a 3 button pointer as follows:

```
% xmodmap –e “pointer = 3 2 1”
```

Many editor applications support the notion of Meta keys (similar to Control keys except that Meta is held down instead of Control). However, some servers do not have a Meta keysym in the default keymap table, so one needs to be added by hand. The following command will attach Meta to the Multi-language key (sometimes labeled Compose Character). It also takes advantage of the fact that applications that need a Meta key simply need to get the keycode and don’t require the keysym to be in the first column of the keymap table. This means that applications that are looking for a Multi_key (including the default modifier map) won’t notice any change.

```
% xmodmap –e “keysym Multi_key = Multi_key Meta_L”
```

One of the more simple, yet convenient, uses of xmodmap is to set the keyboard’s “rubout” key to generate an alternate keysym. This frequently involves exchanging Backspace with Delete to be more comfortable to the user. If the ttyModes

This option indicates that the current modifier map should be printed on the standard output.

-pk

This option indicates that the current keymap table should be printed on the standard output.

-pke

This option indicates that the current keymap table should be printed on the standard output in the form of expressions that can be fed back to xmodmap.

-pp

This option indicates that the current pointer map should be printed on the standard output.

-

A lone dash means that the standard input should be used as the input file.

The filename specifies a file containing xmodmap expressions to be executed. This file is usually kept in the user’s home directory with a name like .xmodmaprc.

EXPRESSION GRAMMAR

The xmodmap program reads a list of expressions and parses them all before attempting to execute any of them. This makes it possible to refer to keysyms that are being redefined in a natural way without having to worry as much about name conflicts.

```
keycode NUMBER = KEYSYMNAME ...
```

The list of keysyms is assigned to the indicated keycode (which may be specified in decimal, hex or octal and can be determined by running the xev program in the examples directory).

```
keysym KEYSYMNAME = KEYSYMNAME ...
```

The KEYSYMNAME on the left hand side is translated into matching keycodes used to perform the corresponding set of keycode expressions. The list of keysym names may be found in the header file <X11/keysymdef.h> (without the XK_ prefix) or the keysym database /usr/lib/X11/XKeysymDB. Note that if the same keysym is bound to multiple keys, the expression is executed for each matching keycode.

```
clear MODIFIERNAME
```

This removes all entries in the modifier map for the given modifier, where valid name are: Shift, Lock, Control, Mod1, Mod2, Mod3, Mod4, and Mod5 (case does not matter in modifier names, although it does matter for all other names). For example, ‘‘clear Lock’’ will remove all any keys that were bound to the shift lock modifier.

```
add MODIFIERNAME = KEYSYMNAME ...
```

2.6.12 Xmodmap Manual Page

This is the ViewStation xmodmap local client man page.

NAME

xmodmap - utility for modifying keymaps in X

SYNOPSIS

xmodmap [-options ...] [filename]

DESCRIPTION

The xmodmap program is used to edit and display the keyboard modifier map and keymap table that are used by client applications to convert event keycodes into keysyms. It is usually run from the user′s session startup script to configure the keyboard according to personal tastes.

OPTIONS

The following options may be used with xmodmap:

-display display

This option specifies the host and display to use.

-help

This option indicates that a brief description of the command line arguments should be printed on the standard error channel. This will be done whenever an unhandled argument is given to xmodmap.

-grammar

This option indicates that a help message describing the expression grammar used in files and with -e expressions should be printed on the standard error.

-verbose

This option indicates that xmodmap should print logging information as it parses its input.

-quiet

This option turns off the verbose logging. This is the default.

-n

This option indicates that xmodmap should not change the mappings, but should display what it would do, like make(1) does when given this option.

-e expression

This option specifies an expression to be executed. Any number of expressions may be specified from the command line.

-pm

Access is restricted to only those machines on the list of allowed hosts (i.e. access control is turned on).

nothing

If no command line arguments are given, the list of hosts that are allowed to connect is printed on the standard output along with a message indicating whether or not access control is currently enabled. This is the only option that may be used from machines other than the one on which the server is running.

FILES

/etc/X*.hosts

SEE ALSO

X(1), Xserver(1)

ENVIRONMENT

DISPLAY to get the default host and display to use.

BUGS

You can’t specify a display on the command line because -display is a valid command line argument (indicating that you want to remove the machine named “display” from the access list).

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.

See X(1) for a full statement of rights and permissions.

AUTHORS

Bob Scheifler, MIT Laboratory for Computer Science,

Jim Gettys, MIT Project Athena (DEC).

waitforserver takes no command-line options, but will use the DISPLAY environment variable to specify a display other than the local X Server.

2.6.11 Xhost Manual Page

This is the ViewStation's xhost local client man page.

NAME

xhost - server access control program for X

SYNOPSIS

xhost [[+-]hostname ...]

DESCRIPTION

The xhost program is used to add and delete hosts to the list of machines that are allowed to make connections to the X server. This provides a rudimentary form of privacy control and security. It is only sufficient for a workstation (single user) environment, although it does limit the worst abuses. Environments which require more sophisticated measures should use the hooks in the protocol for passing authentication data to the server.

The server initially allows network connections only from programs running on the same machine or from machines listed in the file /etc/X*.hosts (where * is the display number of the server). The xhost program is usually run either from a startup file or interactively to give access to other users.

Hostnames that are followed by two colons (::) are used in checking DECnet connections; all other hostnames are used for TCP/IP connections.

OPTIONS

Xhost accepts the following command line options described below. For security, the options that effect access control may only be run from the same machine as the server.

[+]hostname

The given hostname (the plus sign is optional) is added to the list of machines that are allowed to connect to the X server.

-hostname

The given hostname is removed from the list of machines that are allowed to connect to the server. Existing connections are not broken, but new connection attempts will be denied. Note that the current machine is allowed to be removed; however, further connections (including attempts to add it back) will not be permitted. Resetting the server (thereby breaking all connections) is the only way to allow local connections again.

+

Access is granted to everyone, even if they aren't on the list of allowed hosts (i.e. access control is turned off).

-

2.6.10 Waitforserver Manual Page

This is the ViewStation’s waitforserver local client man page.

NAME

waitforserver - utility to insure X server is accepting connections

SYNOPSIS

waitforserver

DESCRIPTION

waitforserver is a trivial X client which attempts repeatedly to open a connection to the default server, using XOpenDisplay(“”). Each attempt will try for the “network timeout” time period, and if the call returns a NULL Display, the XOpenDisplay() function will be called again. Once a successful connection is made to the X Server, the client will exit with a status of 0.

This client is useful in ViewStation FX Startup Command Files, in order to wait for the X Server to get up and running before attempting to start any clients in the background. This is necessary since the Startup Command File is executed before the X Server is started.

A “chicken and egg” problem exists if you try to include waitforserver directly in a Startup Command File, since if it is started in the foreground it will never succeed in making an X connection, and the ViewStation will appear to “hang”. This is because the X Server is not started until after the Startup Command File finishes executing, and the waitforserver client will not terminate (and therefore the Startup Command File will not finish executing) until a successful X connection is made. The solution to this problem is to make another shell script which consists of a single command to invoke the original Startup Command File in the background, and use this new shell script as the Startup Command File.

For example, if you wish to invoke the mwm and hdsdclock clients after the X Server is running, create a file with the following lines:

```
waitforserver

mwm &

hdsdclock &
```

You could name this file start_clients, and then create a Startup Command File called startup which consists of the single line:

```
sh <path>/start_clients &
```

where <path> is the local filesystem path to the start_clients shell script.

OPTIONS

COPYRIGHT

AUTHOR

Michael S. Hartman, Human Designed Systems, Inc.

XmLabel locator

XmScrolledList toc

XmScrolledWindow pageview

Core clip

hdsview page

TopLevelShell viewzoom

XmForm form

hdsview page

XmPushButton dismiss

TopLevelShell information

XmForm form

XmText infotext

XmPushButton dismiss

XmMessageBox error

XmMessageBox copyright

ENVIRONMENT

LPDEST The LPDEST environment variable gives the default printer destination.

LIMITATIONS

If the document does not begin with "%!PS-Adobe-", it does not claim conformance to the document structuring convention. When these documents are encountered, the functionality of hdsview is limited to giving you scroll bars and a next page capability. Because there is no table of contents, skipping around the document and marking pages is impossible.

If there is no table of contents for the document, the popup zoom window will always show the first page.

There is no visual indication of when an orientation or media is being "forced".

The NFS file access protocol must be used to view remote multiple page documents.

There is no support for local printing.

Remote printing works only when hdsview has been started by rshing from a remote host.

Saving the document to a file is not supported.

The zoomview window does not popup under the cursor as it should.

BUGS

Please report all bugs to Technical Support at Human Designed Systems, Inc. Bug reports can be e-mailed to support@hds.com. Be sure to include the hdsview version and the ViewStation server code release version in the bug report.

XmToggleButton zero
XmToggleButton one
XmToggleButton two
XmToggleButton three
XmToggleButton four
XmToggleButton five
XmRowColumn orientationButton
XmToggleButton portrait
XmToggleButton landscape
XmToggleButton upsidedown
XmToggleButton seascape
XmSeparator
XmToggleButton swap
XmRowColumn pagemediaButton
XmToggleButton Letter
XmToggleButton Tabloid
XmToggleButton Ledger
XmToggleButton Legal
XmToggleButton Statement
XmToggleButton Executive
XmToggleButton A3
XmToggleButton A4
XmToggleButton A5
XmToggleButton B4
XmToggleButton B5
XmToggleButton Folio
XmToggleButton Quarto
XmToggleButton 10x14
XmRowColumn helpButton
XmPushButton generalhelp
XmPushButton filehelp
XmPushButton pagehelp
XmPushButton magstephelp
XmPushButton orienthelp
XmPushButton mediahelp

-ncdwm

Equivalent to setting “HDSview.ncdwm: True”.

-noncdwm

Equivalent to setting “HDSview.ncdwm: False”.

WIDGET HIERARCHY

The hierarchy of the hdsview application:

HDSview hdsview
    XmForm form
    XmRowColumn menuBar
    XmRowColumn fileButton
    XmPushButton open
    XmPushButton reopen
    XmPushButton printwhole
    XmPushButton printmarked
    XmPushButton save
    XmSeparator
    XmPushButton copyright
    XmPushButton quit
    XmRowColumn pageButton
    XmPushButton next
    XmPushButton show
    XmPushButton prev
    XmSeparator
    XmPushButton center
    XmSeparator
    XmPushButton mark
    XmPushButton unmark
    XmRowColumn magstepButton
    XmToggleButton -5
    XmToggleButton -4
    XmToggleButton -3
    XmToggleButton -2
    XmToggleButton -1

-legal

Equivalent to setting "HDSview.pageMedia: Legal".

-statement

Equivalent to setting "HDSview.pageMedia: Statement".

-executive

Equivalent to setting "HDSview.pageMedia: Executive".

-a3

Equivalent to setting "HDSview.pageMedia: A3".

-a4

Equivalent to setting "HDSview.pageMedia: A4".

-a5

Equivalent to setting "HDSview.pageMedia: A5".

-b4

Equivalent to setting "HDSview.pageMedia: B4".

-b5

Equivalent to setting "HDSview.pageMedia: B5".

-folio

Equivalent to setting "HDSview.pageMedia: Folio".

-quarto

Equivalent to setting "HDSview.pageMedia: Quarto".

-10x14

Equivalent to setting "HDSview.pageMedia: 10x14".

-forcemedia

Equivalent to setting "HDSview.forcePageMedia: True".

-force

Equivalent to setting "HDSview.Force: True".

-swap

Equivalent to setting "HDSview.swapLandscape: True".

-noswap

Equivalent to setting "HDSview.swapLandscape: False".

-openwindows

Equivalent to setting "HDSview.openWindows: True".

-noopenwindows

Equivalent to setting "HDSview.openWindows: False".

Equivalent to setting "HDSview.Labels: False".

-quiet

Equivalent to setting "*hdsview.quiet: True".

-noquiet

Equivalent to setting "*hdsview.quiet: False".

-preload file

Equivalent to setting "*hdsview.preload: file".

-xdpi dpi

Equivalent to setting "*hdsview.xdpi: dpi".

-ydpi dpi

Equivalent to setting "*hdsview.ydpi: dpi".

-resolution dpi

Equivalent to setting "*hdsview.Resolution: dpi".

-dpi dpi

Equivalent to setting "*hdsview.Resolution: dpi".

-magstep magstep

Equivalent to setting "HDSview.magstep: magstep".

-portrait

Equivalent to setting "HDSview.orientation: Portrait".

-landscape

Equivalent to setting "HDSview.orientation: Landscape".

-upsidedown

Equivalent to setting "HDSview.orientation: Upside-down".

-seascape

Equivalent to setting "HDSview.orientation: Seascape".

-forceorientation

Equivalent to setting "HDSview.forceOrientation: True".

-page label

Equivalent to setting "HDSview.page: label".

-letter

Equivalent to setting "HDSview.pageMedia: Letter".

-tabloid

Equivalent to setting "HDSview.pageMedia: Tabloid".

-ledger

Equivalent to setting "HDSview.pageMedia: Ledger".

72. The default xdpi and ydpi are 300. The height will default to the width if the height is omitted. The ydpi will default to the xdpi if the ydpi is omitted.

OPTIONS

-staticgray

Equivalent to setting "*Visual: StaticGray".

-grayscale

Equivalent to setting "*Visual: GrayScale".

-staticcolor

Equivalent to setting "*Visual: StaticColor".

-pseudocolor

Equivalent to setting "*Visual: PseudoColor".

-truecolor

Equivalent to setting "*Visual: TrueColor".

-directcolor

Equivalent to setting "*Visual: DirectColor".

-install

Equivalent to setting "HDSview.installStdCmap: True".

-noinstall

Equivalent to setting "HDSview.installStdCmap: False".

-private

Equivalent to setting "HDSview.privateCmap: True".

-noprivate

Equivalent to setting "HDSview.privateCmap: False".

-center

Equivalent to setting "HDSview.autoCenter: True".

-nocenter

Equivalent to setting "HDSview.autoCenter: False".

-locator

Equivalent to setting "HDSview.showLocator: True".

-nolocator

Equivalent to setting "HDSview.showLocator: False".

-labels

Equivalent to setting "HDSview.Labels: True".

-nolabels

HDSVIEW WIDGET RESOURCES

Certain resources in the hdsview widget may be set by the user. These selected resources are presented below.

busyCursor (class Cursor)

The cursor shown when hdsscript is rendering to the window. The busy cursor is set to the "target" by the application defaults.

cursor (class Cursor)

The cursor shown when hdsscript is idle. The default cursor is the "crosshair".

interpreter (class Interpreter)

The name of the executable to call to render the PostScript. It is convenient to set this resource to the path of an alternate version of hdsscript for testing. The default value is "hdsscript".

preload (class Preload)

Files to load before the PostScript program. It is convenient to name files that preload fonts here for PostScript programs that continually reload fonts while rendering a page. The default is to not preload any files.

quiet (class Quiet)

Tells hdsscript whether to produce informational messages. The default value is "false".

useBackingPixmap (class UseBackingPixmap)

Tells whether to use a backing pixmap. If this resource is false, backing store is requested on the hdsview window. Some X servers have limited resources for large pixmaps. Also, some X servers' backing store is much faster than using a backing pixmap. You should reset this resource if your X server is one of the server types mentioned. The default value is "true".

xdpi (class Resolution)

Sets the X resolution of the window in dots per inch. You can use this resource to affect the main window. Zoom windows have their X dpi set explicitly in the program. The default value is calculated from the screen metrics.

ydpi (class Resolution)

Sets the Y resolution of the window in dots per inch. You can use this resource to affect the main window. Zoom windows have their Y dpi set explicitly in the program. The default value is calculated from the screen metrics.

HDSVIEW WIDGET ACTIONS

notify(width height xdpi ydpi)

The notify event is used by the hdsview application for the locator and popup zoom windows. If the width and height are 0, the event is used for the locator. Otherwise, it triggers a popup zoom window. The default width and height are

forceOrientation (class Force)

Tells whether to force the orientation on the document. The default is ''false''.

forcePageMedia (class Force)

Tells whether to force the page media on the document. The default is ''false''.

swapLandscape (class SwapLandscape)

Tells whether to swap the meaning of Landscape and Seascape. The default is ''false''.

printCommand (class PrintCommand)

Sets the command used for printing. The printer environment variable is set to the desired printer and then this command is executed using popen. This command should read from ''stdin'' and send the file to the appropriate printer. The default value is ''lp''.

printerVariable (class PrinterVariable)

Gives the name of the printer environment variable. The default value is ''LPD-EST''.

defaultPrinter (class DefaultPrinter)

Gives the printer name to use when the printer environment variable is not set. The default value is NULL.

printPrompt (class PrintPrompt)

Sets the prompt used to ask for the printer name. The default value is ''Printer Name: ''.

printFail (class printFail)

Sets the string used to inform the user that the printer command failed. The default is ''''lpr'' command failed.''.

openPrompt (class OpenPrompt)

Sets the prompt used to ask for a file name to open. The default value is ''Open File: ''.

openFail (class OpenFail)

Sets the string used to inform the user that the open failed. The default value is ''Cannot open file: ''.

savePrompt (class SavePrompt)

Sets the prompt used to ask for a file name to save. The default value is ''Save File: ''.

saveFail (class SaveFail)

Sets the string used to inform the user that the save failed. The default value is ''Cannot save file: ''.

Shift-Left (arrow)	Bound to hdsviewForce() hdsviewSetOrientation(seascape) which is equivalent to pushing Seascape with the second mouse button on the Orientation menu.

APPLICATION RESOURCES

The following application resources may be set to control the default behavior of hdsview.

showLocator (class Labels)

Tells whether to display the locator. The default is "true".

installStdCmap (class InstallStdCmap)

Tells whether to attempt to install a standard colormap. The default is "false".

privateCmap (class PrivateCmap)

Tells whether to use a standard colormap that is not the default colormap. The default is "false".

autoCenter (class AutoCenter)

Tells whether to center the page within the viewport whenever the page size changes. The default is "true".

horizonalMargin (class Margin)

Tells how many pixels hdsview should reserve for window decorations in the horizontal direction. The default value is "20".

verticalMargin (class Margin)

Tells how many pixels hdsview should reserve for window decorations in the vertical direction. The default value is "44".

minimumMagstep (class Magstep)

Tells the smallest magstep to display. The default is "-5".

maximumMagstep (class Magstep)

Tells the largest magstep to display. The default is "5".

magstep (class Magstep)

Sets the default magstep. The default is "0".

orientation (class Orientation)

Sets the default orientation. The default is "Portrait".

page (class Page)

Gives the initial page to display. This resource only affects the display of the file listed on the command line. The default is NULL.

pageMedia (class PageMedia)

Sets the default page media. The default is "Letter".

hdsviewEraseLocator()

Used to erase the locator when leaving an hdsview widget.

hdsviewCheckFile()

Checks to see if the file changed and refreshes the screen if necessary.

KEYBOARD TRANSLATIONS

The pulldown menu entry must be sensitive (i.e. not grayed out) for the action to have effect. Here is the default keyboard binding:

+	Bound to hdsviewIncreaseMagstep() which increases the magstep by 1.
-	Bound to hdsviewDecreaseMagstep() which decreases the magstep by 1.
G	Bound to hdsviewLeft() which scrolls the main viewport left.
J	Bound to hdsviewDown() which scrolls the main viewport down.
K	Bound to hdsviewUp() which scrolls the main viewport up.
L	Bound to hdsviewRight() which scrolls the main viewport right.
Up (arrow)	Bound to hdsviewDefault()hdsviewSetOrientation(portrait) which is equivalent to pushing Portrait with the first mouse button on the Orientation menu.
Right (arrow)	Bound to hdsviewDefault() hdsviewSetOrientation(landscape) which is equivalent to pushing Landscape with the first mouse button on the Orientation menu.
Down (arrow)	Bound to hdsviewDefault() hdsviewSetOrientation(upside-down) which is equivalent to pushing Upside-down with the first mouse button on the Orientation menu.
Left (arrow)	Bound to hdsviewDefault() hdsviewSetOrientation(seascape) which is equivalent to pushing Seascape with the first mouse button on the Orientation menu.
Shift-Up (arrow)	Bound to hdsviewForce() hdsviewSetOrientation(portrait) which is equivalent to pushing Portrait with the second mouse button on the Orientation menu.
Shift-Right (arrow)	Bound to hdsviewForce() hdsviewSetOrientation(landscape) which is equivalent to pushing Landscape with the second mouse button on the Orientation menu.
Shift-Down (arrow)	Bound to hdsviewForce() hdsviewSetOrientation(upside-down) which is equivalent to pushing Upside-down with the second mouse button on the Orientation menu.

Equivalent to pushing the Unmark menu button on the Page menu.

hdsviewSetMagstep(magstep)

Sets the magstep. The parameter must be an integer.

hdsviewIncreaseMagstep()

Increases magstep by one.

hdsviewDecreaseMagstep()

Decreases magstep by one.

hdsviewSetOrientation(orientation)

Set the orientation to the passed parameter. The parameter must be portrait, landscape, upside-down, or seascape.

hdsviewSwapLandscape()

Equivalent to pushing the Swap Landscape menu button on the Orientation menu.

hdsviewSetPageMedia(media-name)

Sets the media. The parameter should be either a media defined in the document or a standard media.

hdsviewDefault()

The orientation or media being set is not forced on the document. This action is called before the action that sets the orientation or media.

hdsviewForce()

The orientation or media being set is forced on the document. This action is called before the action that sets the orientation or media.

hdsviewDeleteWindow()

Destroy the current window. This provides a way to implement the Delete Window protocol for window managers.

hdsviewDismiss()

Pop down the current window. This provides a way to implement the Delete Window protocol for window managers.

hdsviewScrollUp()

Scroll the main viewport up.

hdsviewScrollDown()

Scroll the main viewport down.

hdsviewScrollLeft()

Scroll the main viewport left.

hdsviewScrollRight()

Scroll the main viewport right.

3	Equivalent to pushing the 3 menu button on the Magstep menu.
4	Equivalent to pushing the 4 menu button on the Magstep menu.
5	Equivalent to pushing the 5 menu button on the Magstep menu.

ACTIONS

Most of the pulldown menu commands have an equivalent action that can be used in a translation. The pulldown menu entry must be sensitive (i.e. not grayed out) for the action to have effect. Here is the list of actions:

hdsviewCopyright()

Equivalent to pushing the Copyright... menu button on the File menu.

hdsviewQuit()

Equivalent to pushing the Quit menu button on the File menu.

hdsviewOpen()

Equivalent to pushing the Open... menu button on the File menu.

hdsviewReopen()

Equivalent to pushing the Reopen menu button on the File menu.

hdsviewSave()

Equivalent to pushing the Save marked pages... menu button on the File menu.

hdsviewPrintWhole()

Equivalent to pushing the Print... menu button on the File menu.

hdsviewPrintMarked()

Equivalent to pushing the Print marked pages... menu button on the File menu.

hdsviewPrevious()

Equivalent to pushing the Previous menu button on the Page menu.

hdsviewShow()

Equivalent to pushing the Redisplay menu button on the Page menu.

hdsviewNext()

Equivalent to pushing the Next menu button on the Page menu.

hdsviewCenter()

Equivalent to pushing the Center menu button on the Page menu.

hdsviewMark()

Equivalent to pushing the Mark menu button on the Page menu.

hdsviewUnmark()

Most of the pulldown menu commands have an equivalent action that can be invoked from the keyboard. The pulldown menu entry must be sensitive (i.e. not grayed out) for the accelerator to have effect. Here are the default keyboard accelerators:

Q	Equivalent to pushing the Quit menu button on the File menu.
O	Equivalent to pushing the Open... menu button on the File menu.
R	Equivalent to pushing the Reopen menu button on the File menu.
S	Equivalent to pushing the Save marked pages... menu button on the File menu.
P	Equivalent to pushing the Print marked pages... menu button on the File menu.
Shift-P	Equivalent to pushing the Print... menu button on the File menu.
BackSpace	Equivalent to pushing the Previous menu button on the Page menu.
space	Equivalent to pushing the Next menu button on the Page menu.
Ctrl-L	Equivalent to pushing the Redisplay menu button on the Page menu.
M	Equivalent to pushing the Mark menu button on the Page menu.
N	Equivalent to pushing the Unmark menu button on the Page menu.
0	Equivalent to pushing the 0 menu button on the Magstep menu.
1	Equivalent to pushing the 1 menu button on the Magstep menu.
2	Equivalent to pushing the 2 menu button on the Magstep menu.

Set the orientation to Landscape. This implies a clockwise rotation of the paper by 90 degrees. A landscape orientation indicates that the longest edge of the paper is parallel to the horizontal (x) axis.

Upside-down

Set the orientation to Upside-Down.

Seascape

Set the orientation to Seascape. This implies a counterclockwise rotation of the paper by 90 degrees.

Swap Landscape

Swap the meaning of Landscape and Seascape, since some documents need to be rotated the other way. The swap landscape button allows hdsview to automatically rotate the document the right way in response to the %%Orientation comment in the PostScript file.

Media

The entries on the Media menu set the page media. Media defined in the document appear at the beginning of the menu separated by a line from the standard media. A "diamond" appears in front of the current media.

Here are the standard media names and their sizes. The size is given as the width and height in PostScript points.

Letter 612 x 792 (8.5 x 11 in.)
Tabloid 792 x 1224 (11 x 17 in.)
Ledger 1224 x 792 (17 x 11 in.)
Legal 612 x 1008 (8.5 x 14 in.)
Statement 396 x 612 (5.5 x 8.5 in.)
Executive 540 x 720 (7.5 x 10 in.)
A3 842 x 1190
A4 595 x 842
A5 420 x 595
B4 729 x 1032
B5 516 x 729
Folio 612 x 936 (8.5 x 13 in.)
Quarto 610 x 780
10x14 720 x 1008 (10 x 14 in.)

KEYBOARD ACCELERATORS

Pops up a dialog widget to ask for the file name. Saves the marked pages in the selected file. If no pages have been marked, the mark procedure is called before saving and the unmark procedure is called after saving.

Copyright...

Pops up the copyright window.

Quit

Causes hdsview to exit.

Page

This popup menu controls page access:

Next

Display the next page.

Redisplay

Display the current page.

Previous

Display the previous page.

Center

Center the page in the viewport.

Mark

Mark the pages that have been selected in the table of contents. If no pages have been selected, mark the current page.

Unmark

Unmark the pages that have been selected in the table of contents. If no pages have been selected, unmark the current page.

Magstep

The Magstep menu controls the magnification at which the document is viewed. The default magstep of 0 implies no magnification (i.e. the size on the screen should match the size on paper). Hdsview borrows the notion of magstep from TeX. The magnification is defined to be 1.2**magstep. At magstep 1, the document is magnified by 1.2. At magstep -1, the document is reduced by 1.2. The Magstep menu lists values from -5 to 5.

Orientation

The Orientation menu controls the display orientation. A "diamond" appears in front of the current orientation.

Portrait

Set the orientation to Portrait. This implies no rotation. A portrait orientation indicates that the longest edge of the paper is parallel to the vertical (y) axis.

Landscape

Clicking anywhere within the viewport will popup a zoom window. The window is centered about the location that was clicked. Clicking with the first mouse button pops up a low resolution zoom window. Clicking with the second mouse button pops up a medium resolution zoom window. Clicking with the third mouse button pops up a high resolution zoom window. The cursor in the zoom window will display a "target" or "cross hair" depending on the state of hdsscript. The locator reports the position of the cursor in the zoom windows as well as the main viewport.

In the table of contents, the first and third mouse button functions exactly as they do in a Text widget. That is the first mouse button selects text and the third mouse button extends selections. However, clicking on a page label with the second mouse button will cause that page to be shown. The page being displayed is marked with a '<' in the right margin of the table of contents.

POPUP WINDOWS

The are several additional windows that may appear. The copyright window will appear whenever copyright information is requested. The information window appears whenever the hdsscript process writes to "stdout" or "stderr". Whenever there is an error in the PostScript program, the stack trace will appear in this window. When hdsscript is invoked without the quiet option, informational message regarding the state of hdsscript will appear in the information window. The copyright and information windows can be dismissed by pushing the "Dismiss" button.

MENUS

There are six buttons on the menu bar. They are the File, Page, Magstep, Orientation, Media, and Help buttons. There are keyboard accelerators for most of the menu items. Be sure to read the keyboard accelerator section.

File

This popup menu controls file access:

Open...

Pops up a dialog widget to ask for the file name. The widget is in total control until it pops down. Opens the file for viewing.

Reopen

Reopens the current file.

Print...

Pops up a dialog widget to ask for the printer name. Sends the whole document to the printer.

Print marked pages...

Pops up a dialog widget to ask for the printer name. Sends the marked pages to the printer. If no pages have been marked, the mark procedure is called before printing and the unmark procedure is called after printing.

Save marked pages...

Dan Greening wrote most of xtetris 2.0-2.3, based on earlier work by Didier Tallot, Phill Everson, Martyn Shortley, and Adam Marguilies.

2.6.9 HDSview Manual Page

This is the HDS view man page.

NAME

hdsview - View PostScript documents using hdsscript

SYNOPSIS

hdsview [filename]
or
hdsview [-staticgray] [-grayscale] [-staticcolor] [-pseudocolor] [-truecolor] [-directcolor] [-[no]install] [-[no]private] [-[no]center] [-[no]title] [-[no]date] [-[no]locator] [-[no]labels] [-resolution dpi] [-dpi dpi] [-xdpi dpi] [-ydpi dpi] [-[no]quiet] [-preload file] [-magstep n] [-portrait] [-landscape] [-upsidedown] [-seascape] [-letter] [-tabloid] [-ledger] [-legal] [-statement] [-executive] [-a3] [-a4] [-a5] [-b4] [-b5] [-folio] [-quarto] [-10x14] [-force] [-forceorientation] [-forcemedia] [-[no]swap] [-[no]openwindows] [-page label] [-toolkitoption ...] [filename]

DESCRIPTION

The hdsview program provides a Motif user interface for the hdsscript interpreter. Hdsview and hdsscript function as two cooperating programs. Hdsview creates the viewing window and hdsscript draws in it.

Don't be alarmed by the number of options. Generally, one invokes hdsview with just one parameter, the name of the file to be previewed. If the filename is "-", hdsview will read from "stdin". The options provide a way to set X resources from the command line for a single invocation of hdsview. For that reason, discussion of the options is delayed until after the X resources are discussed.

MAIN WINDOW

The main viewport is on the right side of the main window. There are scroll bars along the bottom and right edges of the viewport. To the left of the viewport is the table of contents. If the PostScript file has document structuring convention (DSC) comments, the table of contents will display the page labels (i.e. usually page numbers). At the top of the application is the menu bar, consisting of push buttons. Each push button brings up a pulldown menu. Over the table of contents there is one optional label that contains the locator. The locator shows the location of the cursor in the viewport. The location is expressed in the default user coordinate system. The locator is useful for measuring bounding boxes.

Within the main viewport the mouse cursor is a "target" when hdsscript is doing work. The cursor is a "crosshair" when hdsscript is idle. When moving to another page in a document, it is generally best to wait for hdsscript to become idle. Otherwise, the current hdsscript process must be killed and the overhead of reading the prologue is incurred again.

2.6.8 HDStetris Manual Page

This is the HDS tetris man page.

NAME

hdstetris - X Window block dropping game

SYNOPSIS

hdstetris [X options]

DESCRIPTION

hdstetris is a game involving dropping blocks. As they drop you can move them to the left or to the right by clicking the left or right mouse buttons respectively. Pressing the shift key while clicking the button causes the falling shape to rotate anti-clockwise (left mouse button) or clockwise (right mouse button). Pressing the middle mouse button causes the shape to quickly drop. Note that the mouse must be in the window in which the shapes are dropping for these actions to work correctly.

You can also use the keyboard: h for moving to the left, l for moving to the right, j to rotate clockwise, k to rotate counter-clockwise and space for dropping quickly..

You can also use the direction pad, if your keyboard has one: (Left) to move left, (Down) to rotate clockwise, (Up) to rotate counter clockwise, and (Right) to move right.

Points are scored for each block that comes to rest on the gradually building up pile of blocks. Different blocks in different orientations have different point values. When the pile reaches the top of the screen and no further blocks can be dropped the game ends. When a row of blocks across the screen is completely filled, that row is removed and all the blocks above it drop down. As more rows are deleted the blocks drop faster.

Almost everything about hdstetris can be configured via the application defaults file.

OPTIONS

hdstetris takes standard X options (see X(1)). It adds the following:

-speed value

This sets the game's speed. By default the speed is 10. 20 causes the game to be 2 times faster, 5 makes it 2 times slower. "-speed 50" makes for an extremely fast game. Keep in mind that as you knock out rows, the game's speed increases.

SEE ALSO

X(1)

AUTHORS

William Pohlmann (Human Designed Systems), converted from Athena to Motif start hdsscreensaver with a -display argument, the programs which hdsscreensaver launches will draw on the same display.)

When the screensaver turns off, or is killed, care is taken to restore the “real” virtual root window if there is one. Because of this, it is important that you not kill the screensaver process with kill -9 if you are running a virtual-root window manager. If you kill it with -9, you may need to restart your window manager to repair the damage. This isn’t an issue if you aren’t running a virtual- root window manager.

ENVIRONMENT

DISPLAY to get the default host and display number.

XENVIRONMENT to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

SEE ALSO

hdslogo(1), attraction(1), blitspin(1), greynetic(1),

helix(1), hopalong(1), hypercube(1), imsmap(1), noseguy(1),

pyro(1), qix(1), random(1), rocks(1), rorschach(1),

slidescreen(1), xroger(1).

BUGS

Apparently there are some problems with “XView” programs getting confused and thinking that the screensaver window is the real root window even when the screensaver is not active: ClientMessages intended for the window manager are sent to the screensaver window instead. This could be solved by making HDS saver forward all unrecognised ClientMessages to the real root window, but there may be other problems as well.

COPYRIGHT

Copyright 1993, Human Designed Systems, Inc.

AUTHORS

Joseph H Durant (Human Designed Systems)

Modified merge of xscreensaver and xnlock clients ported as a local client to the ViewStation FX.

Please let me know if you find any bugs or make any improvements.

nographics (class Boolean)

If this is true the graphics user interface widget will not be displayed. up. Default False.

lock (class Boolean)

If true the display will be locked immediately. The user will be prompted for a locking password if one has not already been provided. The default saver is used unless a different one is specified.

lockalways (class Boolean)

If this is true each screensaver will be invoked with the lock option set. Note that the screensaver password must also be set for this to work. up. Default False.

password (class Char)

The screensaver password used when the lock option is set. Default Null.

programs (class Programs)

The graphics demo programs which hdsscreensaver runs when the user is idle. The value of this resource is a string, one screensaver command per line. Each line must contain exactly one command -- no semi-colons.

If the value of this resource is empty, then no programs will be run; the screen will simply be made black.

Note that you must escape the newlines; here is an example of how you might set this in your .Xdefaults file:

```
hdsscreensaver.programs: \

rocks \n\

qix \n\

rocks \n\

hdslogo
```

HOW IT WORKS

When it is time to activate the screensaver, a full-screen black window is created. This window is given the appropriate properties so that, to any subsequently-created programs, it will appear to be a "virtual root" window. The currently selected graphics screensaver program is then run.

When the user becomes active again, the screensaver window is unmapped and the running subprocess is killed by sending it SIGTERM. This is also how the subprocesses are killed when the screensaver decides that it's time to run a different demo if random is selected: the old one is killed and a new one is launched.

Before launching a subprocess, hdsscreensaver stores an appropriate value for $DISPLAY in the environment that the child will recieve. (This is so that if you

-nounfade

If set, the current screensaver will just blank before the screen is restored.

-nographics

If set, the user interface widget will not be displayed.

-password password

The password used if screensaver is set as lock.

-lock

If set the display will be locked immediately. The user will be prompted for a locking password if one has not already been provided. The default saver is used unless a different one is specified.

-lockalways

If set, every screensaver will be invoked with the locking option set. The password option MUST be used in conjunction for this option to be valid.

X DEFAULTS

hdsscreensaver understands the following resources:

program (class Program)

Same as the -program command-line option. Default HDSlogo.

timeout (class Time)

Same as the -timeout command-line option. Default 120 seconds.

cycle (class Time)

Same as the -cycle command-line option. Default 30 seconds.

fade (class Boolean)

Same as the -fade command-line option. If this is true, then when the screensaver activates, the current contents of the screen will fade to black instead of simply winking out. This only works on displays with writable colormaps. Default true. A fade will also be done when switching graphics screensavers (when the cycle timer expires and random is selected.)

unfade (class Boolean)

Same as the -unfade command-line option. If this is true, then when the screensaver deactivates, the original contents of the screen will fade in from black instead of appearing immediately. This only works on displays with writable colormaps. Default True.

fadeSeconds (class Time)

If fade or unfade is true, this is how long the fade will be in seconds (default 1.)

fadeTicks (class Integer)

If fade is true, this is how many times a second the colormap will be changed to effect a fade. Higher numbers yield smoother fades, but may make the fades take longer if your server isn't fast enough to keep up. Default 75.

noseguy	A guy with a big nose wanders around the screen saying things.
pyro	Fireworks.
qix	Flying string art.
random	Cycle through list randomly.
rocks	Flying through an asteroid field.
rorschach	Random inkblot patterns.
slidescreen	Permute the screen image like an 8-puzzle
xroger	Blinking jolly roger flag.

See the individual man pages for more details.

Provisions are made to allow the user to add and delete from this list, modify timeout, cycle, and fade times, as well as instantly view each graphics program.

OPTIONS

hdssaver accepts the following options:

-program screensaver

The screensaver will activate this program after the keyboard and mouse have been idle for the time set by timeout. This program must be a set as a local client of type screensaver as defined by the user's client.config file. Please refer to the ViewStation User's manual for more information concerning local clients. Default value is hdslogo (a blinking HDS logo).

-timeout seconds

The screensaver will activate after the keyboard and mouse have been idle for this many seconds. Default value is 120 seconds with a minimum of ten.

-cycle seconds

After the screensaver has been running for this many seconds, the currently running screensaver will be killed (with SIGTERM), and a new one started. This is only applicable when random is selected and has a default value of 30 seconds with a minimum of 10.

-fade

If set, the current screen will fade out before the screensaver graphics program is run. Default set.

-nofade

If set, the current screen will just blank before the screensaver graphics program is run.

-unfade

If set, the screensaver graphics program will fade out before the screen is restored. Default set.

2.6.7 HDSscreensaver Manual Page

This is the HDS screensaver manual page.

NAME

hdssaver - local ViewStation FX screensaver, chooser, and locking client.

SYNOPSIS

hdssaver [-display host:display.screen] [-program screensaver] [-timeout int] [-cycle int] [-fade] [-nofade] [-unfade] [-nounfade] [-nographics] [-lock] [-lockalways] [-password password] [-xrm resources]

OPTION

hdssaver provides the ViewStation user with a list of locally run screensaver graphics programs using a standard Motif file selection box widget. This program waits until the keyboard and mouse have been idle for a period, and then run the currently chosen graphics demo. It turns off the demo as soon as there is any mouse or keyboard activity. Its purpose is to display pretty pictures on your screen when it is not in use, in keeping with the philosophy that unattended monitors should always be doing something interesting, just like they do in the movies. By running these programs locally, network load is diminished.

This program also acts as a ViewStation screen locker. When lock is selected the user is prompted for a password (maximum of ten characters). Once the password is verified, the currently chosen graphics demo will run. Any further use of the keyboard or mouse causes the ViewStation to prompt for the password. If the correct password is entered within 30 seconds the graphics demo is turned off and the ViewStation returned to its previous state.

The following graphics demo programs are available and can be customized either through command line options, or resource settings:

hdslogo	Blinking HDS logo
attraction	A bouncing ball demo, or a qix-like demo, or a wild color-cycling thing, with some odd rules.
blitspin	Rotate a bitmap using bitblts.
greynetic	Random colored/stippled rectangles.
helix	Generates spirally “stringart” patterns.
hopalong	Fractals.
hypercube	2D projection of a 4D object.
imsmap	Generates random maps or cloud formations

WIDGETS

In order to specify resources, it is useful to know the hierarchy of the widgets which compose *hdsxclock*. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

XClock hdsxclock

Clock clock

ENVIRONMENT

DISPLAY

to get the default host and display number.

XENVIRONMENT

to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

SEE ALSO

X(1), xrdb(1), time(3C), Athena Clock widget

BUGS

Hdsxclock determines the current time (GMT) by broadcasting to the internet *time* service provided by local UNIX hosts. The first valid reply is used. The conversion to local time is done by one of the following methods: If the environment variable TZ is a POSIX style time zone specification, it will be parsed to obtain the normal and daylight-savings offsets from GMT. If the environment variable TZ is a path to a time zone information file (typically /etc/localtime or /usr/share/lib/zoneinfo/localtime on a UNIX host), this file will be used to perform local time conversion. If the previous two methods fail, an approximation to local time conversion will be obtained by querying the internet *daytime* service on the same host which replied to the original *time* service broadcast. This calibration of system time to GMT and local time is local to each instance of hdsxclock, so multiple clocks may show different times if the system clocks of UNIX hosts on the local network are not exactly synchronized.

When in digital mode, the string should be centered automatically.

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.

See X(1) for a full statement of rights and permissions.

AUTHORS

Tony Della Fera (MIT-Athena, DEC)

Dave Mankins (MIT-Athena, BBN)

Ed Moy (UC Berkeley)

-padding *number*

This option specifies the width in pixels of the padding between the window border and clock text or picture. The default is 10 on a digital clock and 8 on an analog clock.

X DEFAULTS

The application class name is XClock. This program uses the *Athena Clock* widget. It understands all of the core resource names and classes as well as:

width (class Width)

Specifies the width of the clock. The default for analog clocks is 164 pixels; the default for digital clocks is whatever is needed to hold the clock when displayed in the chosen font.

height (class Height)

Specifies the height of the clock. The default for analog clocks is 164 pixels; the default for digital clocks is whatever is needed to hold the clock when displayed in the chosen font.

update (class Interval)

Specifies the frequency in seconds at which the time should be redisplayed.

foreground (class Foreground)

Specifies the color for the tic marks. The default is depends on whether *reverseVideo* is specified. If *reverseVideo* is specified the default is *white*, otherwise the default is *black*.

hands (class Foreground)

Specifies the color of the insides of the clock's hands. The default is depends on whether *reverseVideo* is specified. If *reverseVideo* is specified the default is *white*, otherwise the default is *black*.

highlight (class Foreground)

Specifies the color used to highlight the clock's hands. The default is depends on whether *reverseVideo* is specified. If *reverseVideo* is specified the default is *white*, otherwise the default is *black*.

analog (class Boolean)

Specifies whether or not an analog clock should be used instead of a digital one. The default is True.

chime (class Boolean)

Specifies whether or not a bell should be rung on the hour and half hour.

padding (class Margin)

Specifies the amount of internal padding in pixels to be used. The default is 8.

font (class Font)

Specifies the font to be used for the digital clock. Note that variable width fonts currently will not always display correctly.

2.6.6 HDSxclock Manual Page

This is the HDS xclock man page.

NAME

hdsxclock - analog / digital clock for X

SYNOPSIS

hdsxclock [-*toolkitoption* ...] [-help] [-analog] [-digital] [-chime] [-hd *color*] [-hl *color*] [-update *seconds*] [-padding *number*]

DESCRIPTION

The *hdsxclock* program displays the time in analog or digital form. The time is continuously updated at a frequency which may be specified by the user. This program is nothing more than a wrapper around the Athena Clock widget.

OPTIONS

Hdsxclock accepts all of the standard X Toolkit command line options along with the additional options listed below:

-help

This option indicates that a brief summary of the allowed options should be printed on the standard error.

-analog

This option indicates that a conventional 12 hour clock face with tick marks and hands should be used. This is the default.

-digital

This option indicates that a 24 hour digital clock should be used.

-chime

This option indicates that the clock should chime once on the half hour and twice on the hour.

-hd ***color***

This option specifies the color of the hands on an analog clock. The default is *black*.

-hl ***color***

This option specifies the color of the edges of the hands on an analog clock, and is only useful on color displays. The default is *black*.

-update ***seconds***

This option specifies the frequency in seconds at which *hdsxclock* should update its display. If the clock is obscured and then exposed, it will be updated immediately. A value of less than 30 seconds will enable a second hand on an analog clock. The default is 60 seconds.

COLORS

The application class name is Clock. Below are the default colors:

Clock*Background: grey
Clock*BorderColor: light blue
Clock*hour: yellow
Clock*jewel: yellow
Clock*minute: yellow

SEE ALSO

X(1), X Toolkit documentation

BUGS

Hdsoclock determines the current time (GMT) by broadcasting to the internet *time* service provided by local UNIX hosts. The first valid reply is used. The conversion to local time is done by one of the following methods: If the environment variable TZ is a POSIX style time zone specification, it will be parsed to obtain the normal and daylight-savings offsets from GMT. If the environment variable TZ is a path to a time zone information file (typically /etc/localtime or /usr/share/lib/zoneinfo/localtime on a UNIX host), this file will be used to perform local time conversion. If the previous two methods fail, an approximation to local time conversion will be obtained by querying the internet *daytime* service on the same host which replied to the original *time* service broadcast. This calibration of system time to GMT and local time is local to each instance of hdsoclock, so multiple clocks may show different times if the system clocks of UNIX hosts on the local network are not exactly synchronized.

COPYRIGHT

Copyright 1989, Massachusetts Institute of Technology.

See X(1) for a full statement of rights and permissions.

AUTHOR

Keith Packard, MIT X Consortium

2.6.5 HDSoclock Manual Page

This is the HDS oclock manual page.

NAME

hdsoclock - display time of day

SYNOPSIS

hdsoclock [-option ...]

DESCRIPTION

Hdsoclock simply displays the current time on an analog display

OPTIONS

-fg ***foreground color***

choose a different color for the both hands and the jewel of the clock

-bg ***background color***

choose a different color for the background.

-jewel ***jewel color***

choose a different color for the jewel on the clock.

-minute ***minute color***

choose a different color for the minute hand of the clock.

-hour ***hour color***

choose a different color for the hour hand of the clock.

-geometry ***geometry***

define the initial window geometry; see X(1).

-display ***display***

specify the display to use; see X(1).

-bd ***border color***

choose a different color for the window border.

-bw ***border width***

choose a different width for the window border. As the Clock widget changes its border around quite a bit, this is most usefully set to zero.

-noshape

causes the clock to not reshape itself and ancestors to exactly fit the outline of the clock.

-transparent

causes the clock to consist only of the jewel, the hands, and the border.

X(1), xrdb(1), time(3C).

BUGS

Hdsdclock determines the current time (GMT) by broadcasting to the internet *time* service provided by local UNIX hosts. The first valid reply is used. The conversion to local time is done by one of the following methods: If the environment variable TZ is a POSIX style time zone specification, it will be parsed to obtain the normal and daylight-savings offsets from GMT. If the environment variable TZ is a path to a time zone information file (typically /etc/localtime or /usr/share/lib/zoneinfo/localtime on a UNIX host), this file will be used to perform local time conversion. If the previous two methods fail, an approximation to local time conversion will be obtained by querying the internet *daytime* service on the same host which replied to the original *time* service broadcast. This calibration of system time to GMT and local time is local to each instance of hdsdclock, so multiple clocks may show different times if the system clocks of UNIX hosts on the local network are not exactly synchronized.

Scrolling from 9 to 10 O'Clock seems weird, but chances are you won't notice it.

If the window is too small, the seconds are tough to read. This should be recognized by the program and should display seconds using a font rather than the bitmaps used by the clock itself.

COPYRIGHT

Copyright (c) 1988, Dan Heller.

AUTHOR

Dan Heller -- <argv@sun.com> or <dheller@cory.berkeley.edu>

Fade & tails by Steve Reinhardt, <steven@fts.webo.dg.com>

foreground (class Foreground)

Specifies the color for the tic marks. Using the class specifies the color for all things that normally would appear in the foreground color. The default is ‘‘black’’ since the core default for background is ‘‘white.’’

font (class Font)

Specifies the font to be used for the date.

miltime (class Boolean)

Specifies whether the military (24hr) time format should be used.

reverseVideo (class ReverseVideo)

Specifies that the foreground and background colors should be reversed.

scroll (class Boolean)

Specifies whether the digits should scroll or not.

fade (class Boolean)

Specifies whether the digits should fade or not.

fadeRate (class Time)

The number of milliseconds of delay between iterations of fading.

seconds (class Boolean)

Specifies whether the seconds should be displayed or not.

bell (class Boolean)

Specifies whether the bell should sound on the half hour and on the hour.

tails (class Boolean)

Specifies whether tails should be put on the digits 6 and 9.

alarm (class Boolean)

Specifies whether the alarm should go off at the specified time (alarmTime).

alarmTime (class String)

Specifies the time alarm goes off if alarm is set.

ENVIRONMENT

DISPLAY

to get the default host and display number.

XENVIRONMENT

to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

SEE ALSO

-alarmTime *HH:MM*

If alarm is set, it goes off at time specified in 24-hour format.

-seconds

This option will update the clock every second and display the time including the seconds.

-bg *color*

This option specifies the color to use for the background of the window. The default is "white."

-bd *color*

This option specifies the color to use for the border of the window. The default is "black."

-bw *number*

This option specifies the width in pixels of the border surrounding the window.

-fg *color*

This option specifies the color to use for displaying text. The default is "black".

-fn *font*

This option specifies the font to be used for displaying normal text. The default is "Fixed."

-rv

This option indicates that reverse video should be simulated by swapping the foreground and background colors.

-geometry *geometry*

This option specifies the prefered size and position of the clock window.

-display *host:display*

This option specifies the X server to contact.

-xrm *resourcestring*

This option specifies a resource string to be used. This is especially useful for setting resources that do not have separate command line options.

X DEFAULTS

The application class name is DClock. Hdsdclock understands all of the core resource names and classes as well as:

width (class Width)

Specifies the width of the clock.

height (class Height)

Specifies the height of the clock.

-miltime

This option causes the clock to display the time in 24 hour (military) time format.

-tails

This option puts "tails" on the digits 6 and 9.

-scroll

-noscroll

When the time changes, the digits scroll from the previous digit to the next digit. Since this is on by default, the -noscroll option can turn it off.

-fade

This option causes the display to fade from one digit to the next instead of jumping or scrolling. If fade is specified, the scroll attribute will be forced to off.

-fadeRate ***n***

The fadeRate attribute takes an integer value which specifies the number of milliseconds to wait between iterations when fading from one digit to the next. A larger value will make the fade slower and more pronounced. The default is 50. The fadeRate attribute also controls the scroll rate if the scroll attribute is on and the fade attribute is off.

-date ***"format"***

The date is printed under the time in the specified font. The string displayed is in the "format" argument. If the string contains a formatting character (%), then the next character is examined and a value is inserted into the string. Example:

```
hdsdclock -date "Today is %W"
```

The date string will print "Today is Friday" if the weekday name happens to be friday. The formatting characters that are understood are:

%W	Full weekday name
%w	Three-char weekday name (Sun, Mon, Tue...)
%M	Full month name
%m	Three-char month name (Jan, Feb, Mar...)
%d	The date (numerical day number of the month)
%Y	Full year (4 digits)
%y	2-digit year number

-alarm

-noalarm

The alarm is turned on or off. Alarm rings bell and toggles the reverse video five times.

2.6.4 HDSdclock Manual Page

This is the HDS digital clock man page.

NAME

hdsdclock - digital clock for X

SYNOPSIS

hdsdclock [-*toolkitoption* ...] [-option ...]

DESCRIPTION

The *hdsdclock* program displays the time in digital format only. The time is updated on a per second basis or on a per minute basis. This program is nothing more than a wrapper around the dclock widget not associated with any particular widget set.

When the clock is running, the user may change attributes by typing:

r	Toggles **Reverse Video**.
s	Toggles the **seconds** display.
b	Toggles the **bell** attribute.
j	Toggles the **jump/scroll** attribute.
f	Toggles the **fade** attribute.
d	Toggles the **date** format.
m	Toggles the **military time** format.
a	Toggles the **alarm** clock.
t	Toggles the **tails** attribute.
q	quit the program.

The third mouse button puts the clock in the mode where the alarm clock can be set. The alarm must be set in 24-hour format to distringuish between am and pm time. Digits can be changed by selecting the digit with the left or middle mouse button. The Left button advances the time while the Middle button moves the time backwards. Using the left or middle button over the text at the bottom of the clock toggles the alarm from actually being set (or, use the 'a' key).

OPTIONS

Hdsdclock accepts all of the standard X Toolkit command line options along with the additional options listed below:

-help

This option indicates that a brief summary of the allowed options should be printed on the standard error.

-bell

This option indicates that the bell will beep once on the half hour and twice on the hour.

SEE ALSO

X(1) and xdm(1)

COPYRIGHT

AUTHORS

William Pohlmann(Human Designed Systems)

X Version 11 Last change: 1/7/93

2.6.3 HDS Chooser Client

This is the HDS Chooser man page.

NAME

HDS Chooser - XDMCP host menu

SYNOPSIS

Only available by setting run local chooser to yes in the XDMCP options in setup when the xdmcp access mode is broadcast or indirect.

DESCRIPTION

The HDS Chooser program presents a list of all hosts which are willing to manage the session. The user may connect to any of the hosts presented by selecting the host and pressing the Connect button. A host may be selected via the mouse or by typing the desired selection into the selection box. If a host is typed into the selection box which is not listed, pressing the Connect button will result in a warning beep and the connection will not be attempted.

The list will grow as any new hosts respond that they are willing to manage. The chooser will send out the request with an exponential backoff algorithm so that there will be no network traffic when the chooser is up and the display is idle. The retransmissions start at 2 seconds and back off by factors of 2 to 32 seconds and discontinue after 126 seconds. The transmissions will restart after any keyboard or mouse activity.

The user may also press the Refresh button to discard the current list of willing hosts and begin collecting again. The refresh button is most useful when returning to a display that has been idle for some time. Since the list was compiled over time a host which in the past was willing to manage may no longer be. While the user will be informed of any new hosts that are willing to manage the moment the mouse is touched, hosts which are no longer willing are never removed from the list. By discarding the list and starting again with the refresh button, the list will be made current.

If a host is chosen that is no longer willing or is no longer on the network, the chooser will restart. There could be a 126 second delay before the chooser is run if the host was removed from the network.

Unlike the chooser provided with X11 Release 5 this chooser can work both with X11R4 xdm and X11R5 xdm hosts since it does not work via a forwarded request.

WIDGETS

In order to specify resources, it is useful to know the hierarchy of the widgets which compose HDS Chooser. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

Chooser chooser

XmSelectionBox selection

XmPushButton button39

XmPushButton button40

APPLICATION RESOURCES

rpn (Class Rpn)

Specifies that the rpn mode should be used. The default is TI mode.

stipple (Class Stipple)

Indicates that the background should be stippled. The default is “on” for monochrome displays, and “off” for color displays.

cursor (Class Cursor)

The name of the symbol used to represent the pointer. The default is “hand2”.

SEE ALSO

X(1), xrdb(1), the Motif Widget Set

COPYRIGHT

Copyright 1993, Human Designed Systems, Inc. Adapted from xcalc, copyright 1988, 1989, Massachusetts Institute of Technology.

AUTHORS

William Pohlmann (Human Designed Systems)

xcalc by

John Bradley, University of Pennsylvania

Mark Rosenstein, MIT Project Athena

Donna Converse, MIT X Consortium

X Version 11 Last change: 1/6/93

as keyboard accelerators. Because these resources are not specified in the source code, you can create a customized calculator by writing a private application defaults file, using the Motif PushButton and Form widget resources to specify the attachments of the buttons (determining relative position and size), the label for each button, and the function of each button.

The foreground and background colors of each calculator key can be individually specified. For the TI calculator, a classical color resource specification might be:

```
XCalc.ti.XmPushButton.background:  gray50
XCalc.ti.XmPushButton.foreground:  white
```

For each of buttons 20, 25, 30, 35, and 40, specify:

```
XCalc.ti.button20.background: black
XCalc.ti.button20.foreground: white
```

For each of buttons 22, 23, 24, 27, 28, 29, 32, 33, 34, 37, 38, and 39:

```
XCalc.ti.button22.background: white
XCalc.ti.button22.foreground: black
```

WIDGET HIERARCHY

In order to specify resources, it is useful to know the hierarchy of the widgets which compose xcalc. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

```
XCalc xcalc
    XmForm  ti  or  hp    (the name depends on the mode)
        XmFrame  bevel
            XmForm  screen
                XmLabel  M
                XmToggleButton  LCD
                XmLabel  INV
                XmLabel  DEG
                XmLabel  RAD
                XmLabel  GRAD
                XmLabel  P
        XmPushButton  button1
        XmPushButton  button2
        XmPushButton  button3
  and so on, ...
        XmPushButton  button38
```

TI Key	HP Key	Keyboard Accelerator	TI Function	HP Function
SQRT	SQRT	r	squareRoot()	squareRoot()
AC	ON	space	clear()	clear()
AC	<-	Delete	clear()	back()
AC	<-	Backspace	clear()	back()
AC	<-	Control-H	clear()	back()
AC		Clear	clear()	
		q	quit()	quit()
		Control-C	quit()	quit()
INV	i	i	inverse()	inverse()
sin	s	s	sine()	sine()
cos	c	c	cosine()	cosine()
tan	t	t	tangent()	tangent()
DRG	DRG	d	degree()	degree()
e		e	e()	
ln	ln	l	naturalLog()	naturalLog()
y^x	y^x	^	power()	power()
PI	PI	p	pi()	pi()
x!	x!	!	factorial()	factorial()
(		(	leftParen()	
)		)	rightParen()	
/	/	/	divide()	divide()
*	*	*	multiply()	multiply()
-	-	-	subtract()	subtract()
+	+	+	add()	add()
=		=	equal()	
0..9	0..9	0..9	digit()	digit()
.	.	.	decimal()	decimal()
+/-	CHS	n	negate()	negate()
	x:y	x		XexchangeY()
	ENTR	Return		enter()
	ENTR	Linefeed		enter()

CUSTOMIZATION

The application class name is XCalc.

hdscalc has an enormous application defaults file which specifies the position, label, and function of each key on the calculator. It also gives translations to serve

INV

Inverts the meaning of the function keys. This would be the f key on an HP calculator, but xcalc does not display multiple legends on each key. See the individual function keys for details.

10^x

Raises "10.0" to the number in the top of the stack. When inverted, it calculates the log (base 10) of the number in the display. The corresponding action procedure is tenpower().

e^x

Raises "e" to the number in the top of the stack. When inverted, it calculates the log (base e) of the number in the display. The action procedure is epower().

STO

Copies the number in the top of the stack to a memory location. There are 10 memory locations. The desired memory is specified by following this key with a digit key.

RCL

Pushes the number from the specified memory location onto the stack.

SUM

Adds the number on top of the stack to the number in the specified memory location.

x:y

Exchanges the numbers in the top two stack positions, the X and Y registers. The corresponding action procedure is XexchangeY().

R v

Rolls the stack downward. When inverted, it rolls the stack upward. The corresponding action procedure is roll().

blank

These keys were used for programming functions on the HP-10C. Their functionality has not been duplicated in xcalc.

Finally, there are four additional action procedures/callbacks: bell(), which rings the bell; selection(), which performs a cut on the entire number in the calculator's "liquid crystal" display; paste(), which pastes the current select buffer into the display if it can be interpreted as a number; and quit(), which turns off the calculator (exits the program).

ACCELERATORS

Accelerators are shortcuts for entering commands. xcalc provides some sample keyboard accelerators; also users can customize accelerators. The numeric keypad accelerators provided by xcalc should be intuitively correct. The accelerators defined by xcalc on the main keyboard are given below:

*****

Multiplication. The corresponding action procedure is multiply().

-

Subtraction. The corresponding action procedure is subtract().

+

Addition. The corresponding action procedure is add().

=

Perform calculation. The TI-specific action procedure is equal().

STO

Copies the number in the display to the memory location. The corresponding action procedure is store().

RCL

Copies the number from the memory location to the display. The corresponding action procedure is recall().

SUM

Adds the number in the display to the number in the memory location. The corresponding action procedure is sum().

EXC

Swaps the number in the display with the number in the memory location. The corresponding action procedure for the TI calculator is exchange().

+/-

Negate; change sign. The corresponding action procedure is negate().

.

Decimal point. The action procedure is decimal().

Calculator Key Usage (RPN mode): The number keys, CHS (change sign), +, -, *, /, and ENTR keys all do exactly what you would expect them to do. Many of the remaining keys are the same as in TI mode. The differences are detailed below. The action procedure for the ENTR key is enter().

<-

This is a backspace key that can be used if you make a mistake while entering a number. It will erase digits from the display. (See BUGS). Inverse backspace will clear the X register. The corresponding action procedure is back().

ON

Clears the display, the state, and the memory. To clear state, the action procedure is off.

radians, and in ’GRAD’ mode, numbers are in grads. When inverted, the DRG key has a feature of converting degrees to radians to grads and vice-versa. Example: put the calculator into ’DEG’ mode, and enter "45 INV DRG". The display should now show something along the lines of ".785398", which is 45 degrees converted to radians. The corresponding action procedure is degree().

e

The constant ’e’. (2.7182818...). The corresponding action procedure is e().

EE

Used for entering exponential numbers. For example, to get "-2.3E-4" you’d enter "2 . 3 +/- EE 4 +/-". The corresponding action procedure is scientific().

log

Calculates the log (base 10) of the number in the display. When inverted, it raises "10.0" to the number in the display. For example, entering "3 INV log" should result in "1000". The corresponding action procedure is logarithm().

ln

Calculates the log (base e) of the number in the display. When inverted, it raises "e" to the number in the display. For example, entering "e ln" should result in "1". The corresponding action procedure is naturalLog().

y^x

Raises the number on the left to the power of the number on the right. For example "2 y^x 3 =" results in "8", which is 2^3. For a further example, "(1+2+3) y^x (1+2) =" equals "6 y^x 3" which equals "216". The corresponding action procedure is power().

PI

The constant ’pi’. (3.1415927....) The corresponding action procedure is pi().

x!

Computes the factorial of the number in the display. The number in the display must be an integer in the range 0-500, though, depending on your math library, it might overflow long before that. The corresponding action procedure is factorial().

(

Left parenthesis. The corresponding action procedure for TI calculators is leftParen().

)

Right parenthesis. The corresponding action procedure for TI calculators is rightParen().

/

Division. The corresponding action procedure is divide().

tor function as a translation to that button. Each calculator function has both a translation and a callback with the same name. The only exception is for the digit keys where the action used is digit(n), where n is the corresponding digit, 0..9 while the callbacks are digit0, digit1, ..., digit9. Note that when used as an action the name is followed by () whereas when use as a callback only the name is needed.

1/x

Replaces the number in the display with its reciprocal. The corresponding action procedure is reciprocal().

x^2

Squares the number in the display. The corresponding action procedure is square().

SQRT

Takes the square root of the number in the display. The corresponding action procedure is squareRoot().

CE/C

When pressed once, clears the number in the display without clearing the state of the machine. Allows you to re-enter a number if you make a mistake. Pressing it twice clears the state, also. The corresponding action procedure for TI mode is clear().

AC

Clears the display, the state, and the memory. The action procedure to clear the state is off().

INV

Invert function. See the individual function keys for details. The corresponding action procedure is inverse().

sin

Computes the sine of the number in the display, as interpreted by the current DRG mode (see DRG, below). If inverted, it computes the arcsine. The corresponding action procedure is sine().

cos

Computes the cosine, or arccosine when inverted. The corresponding action procedure is cosine().

tan

Computes the tangent, or arctangent when inverted. The corresponding action procedure is tangent().

DRG

Changes the DRG mode, as indicated by 'DEG', 'RAD', or 'GRAD' at the bottom of of the calculator "liquid crystal" display. When in 'DEG' mode, numbers in the display are taken as being degrees. In 'RAD' mode, numbers are in

2.6.2 HDScalc Manual Pages

This is the HDScalc man page.

NAME

hdscalc - scientific calculator for X

SYNOPSIS

hdscalc [-stipple] [-rpn] [-toolkitoption...]

DESCRIPTION

hdscalc is a scientific calculator desktop accessory that can emulate a TI-30 or an HP-10C. It is based on the X11 Release5 **xcalc**, but has been rewritten with added customizability and functionality using the Motif toolkit.

OPTIONS

hdscalc accepts all of the standard toolkit command line options along with two additional options:

-stipple

This option indicates that the background of the calculator should be drawn using a stipple of the foreground and background colors. On monochrome displays this improves the appearance.

-rpn

This option indicates that Reverse Polish Notation should be used. In this mode the calculator will look and behave like an HP-10C. Without this flag, it will emulate a TI-30.

OPERATION

Pointer Usage: Operations may be performed with pointer button 1, or in some cases, with the keyboard. Many common calculator operations have keyboard accelerators. To quit, press pointer button 3 on the display of the calculator.

Calculator Key Usage (TI mode): The numbered keys, the +/- key, and the +, -, *, /, and = keys all do exactly what you would expect them to. It should be noted that the operators obey the standard rules of precedence. Thus, entering "3+4*5=" results in "23", not "35". The parentheses can be used to override this. For example, "(1+2+3)*(4+5+6)=" results in "6*15=90".

The entire number in the calculator display can be selected, in order to paste the result of a calculation into text. A number may also be pasted into the display.

The following is a list of all calculator functions available for use in defining a custom calculator. In departure from the normal scheme of only being able to specify actions in a resource file, this program will also allow callbacks to be specified for any calculator button via the user data resource for that button. The calculator function for any button may therefore be specified as a translation for that button or as the user data for that button, and it would be used as an action or callback respectively. Callbacks are being used so that the translations that implement default Motif button behavior are not affected by adding the calcula-

HDS Serial - HDSterm, a VT320 terminal emulator. This client is described fully in Section 2.2.

HDS Memory - a client that reports current available memory, fragments, and largest available fragment of ViewStation memory. Its operation is described in Section 1.5.5.4.

HDS Console Window - a ViewStation shell and Diagnostic message window which uses HDSterm as its interface. This client is described in Section 1.2.2

MWM - Motif Window Manager. This client is described in Section 2.4. Its full OSF man pages are found on the HDSware tape in the */hds-fx/man* directory.

OLVWM - OPEN LOOK Virtual Window Manager. This client is described in Section 2.5. Its complete man pages, including sample files, are found on the HDSware tape in the */hds-fx/man* directory.

FVWM - FVWM Virtual Window Manager. This client and its accompanying modules is described in Section 2.8.

HDS3270 - IBM 3270 terminal emulator. This client is described in Section 2.3.2.

Multimedia Clients - These audio and video clients have a separate section describing their operation and containing their man pages; see Section 2.7.

The Console Window displays a current list of HDSperx clients with the “list” command. Many of these clients are common UNIX or X clients and do not have individual man pages in this manual.

HDSperx Clients

This chapter describes the HDSperx local clients for the ViewStation.

Section 2.0 - HDS ViewStation Local Clients

2.6 HDSperx Clients

The ViewStation can run X clients locally because of the power of its processor and its Unix micro-kernel architecture. HDSware supplies a number of these clients. You can also compile your own clients to run on the ViewStation with the HDS ViewStation Developer's Kit, a set of software development tools. The Developer's kit is described in Section 5.0.

2.6.1 HDSperx Manual Pages

This note contains printed copies of the manual pages for each of the HDSperx clients. These manual pages are available on the HDSware tape as nroff files in the *./hds-fx/man* directory. This means that you can add these nroff files to your other man pages and have help for the HDSperx clients available with your normal man utility. If you want to see the file as text output, you can use the command:

```
nroff -man <filename.1>
```

The man pages contain all the command line arguments and resources for the HDSperx clients. Man pages for the maze and ico clients are not reprinted here since they are identical to the man pages available on X systems.

The following clients have printed man pages in separate sections of this manual:

HDS Telnet - HDSterm, a VT320 terminal emulator. This client is described fully in Section 2.2, with the man page in Section 2.2.4..