2.1.6.2 Loading X11 Server Extensions

These server extensions can be used by loading them with the clients.config file (in the default configuration, these clients are commented out of this file). You can include them in this file by removing the comment characters as you would one of the HDSperx clients and the extension will be loaded with the server code. Since they are built into the server code, they are actually loaded on reboot.

You should note that these extensions are installed as an integral part of the server code, not as independent clients. For this reason, you should always be careful to match the server code version with the extension files on the same HDSware tape. If you mix different file versions, they won't work. This is not generally a problem, since the extension files are included on the same tape; you just have to be careful not to mix tape sources.

2.1.6.3 XTrap Extension

The XTrap extension is used to test the operation of the X server and X software, typically with third party software testing application programs. It is often used with scripts that capture keypresses and mouse movements, or replays scripts to simulate sessions with user and application interaction. The XTest extension is part of the XTrap extension, though it is often installed separately. Refer to the X11 manual pages for a more complete specification.

2.1.6.4 Multi-Buffering Extension

The Multi-Buffering extension is used to enhance the display of graphics images. The displayed image is swapped for another buffered image which was drawn while it was off-screen, thus giving a fast replacement with the actual drawing invisible to the viewer. Refer to the X11 man pages for more detail.

2.1.6.5 Shared Memory Extension

The Shared Memory extension (MIT-SHM) is used to provide local downloaded clients with a mechanism to use high performance direct shared memory access for X pixmaps. This extension is loaded by default (refer to the clients.config file).

2.1.6.6 PEX-3D Extension

The PEX-3D extension is a 3D Protocol extension to the X Window System. It adds over 200 X protocol requests for defining and displaying 3D pictures.

Details on the PEX-3D Extension are found in Section 2.6.15.

ble. If ’forceLocal’ is false, then the Remote Shell protocol will be used to attempt to run the command remotely.

Usually forceLocal is ’yes’, since the user usually would prefer to run a local version of a command to a remote one.

Default: yes

stackSize

This attribute allows the stack size for the client to be increased. (Note: The HDS Console ’ps’ command shows the stack utilization for running clients.)

transparentMode

This attribute is intended to be used by clients (such as window managers) which are intended to be run locally but to act as though they were running on a remote host. It is only applicable when the client is started from a remote host via the Remote Shell protocol. In that case, if transparentMode is true, the local client will access the remote user’s home environment and use the remote host’s files as though the client were running on the remote host. Also, all commands started by the client will be run on the remote host (unless they are downloadable clients with the ’forceLocal’ attribute set to ’yes’).

If transparentMode is false, then the local client will use the ViewStation’s filesystem (where filenames must be entered relative to local mount points). The local client would also not use the Remote Shell protocol to run commands on the remote host.

Default: no

2.1.6 X11 Server Extensions

Extensions Integrated into the HDSware Server

The HDS ViewStation server software contains a number of built-in X11 server extensions. These are software modules that allow the server to do special functions which are not necessary for the standard X11 server functionality, but operate as optional extensions. One example of an extension of this type is the Shape extension, which allows the server to draw a round window, such as the window used for the "oclock" client application.

The X utility program, xdpyinfo, shows what extensions are available for the active display. Check the HDSware version for the extensions currently supported.

2.1.6.1 Downloadable X11 Server Extensions

There are some additional extensions that are not built in but are available for the server to download as needed from the HDSware location on the network. There are currently four extensions of this type included on the HDSware tape, the XTrap extension, the Multi-Buffering extension, the shared memory extension, and the PEX 3-D drawing extension.

shareFree

This optional attribute applies to clients that have the shareCode attribute set to yes. This attribute determines whether or not the client should be freed when the last instance of the client exits. When this attribute is set to yes, the client will be freed when the last client instance exits. When this attribute is set to no, the client will remain in memory and will not be freed.

Default value: no.

Example:

hds.client.mwm.shareFree: yes

preload

This optional attribute applies only to libraries and indicates the library should be loaded at startup. Libraries are normally loaded when a client is run that uses them. The "preload" attribute will be ignored if the library has the "shareFree" attribute set to yes or if the library is missing or has an empty "path" attribute.

Default value: no.

Example:

hds.library.motif.preload: yes

path

The "path" attribute is REQUIRED for a downloadable client description. This attribute lists the full path which the ViewStation can use to access the executable code for a client.

Example:

hds.client.xclock.path: /FXUtilities/hds-fx/bin/hdsxclock

Note: The same executable code can be referenced by several different client names. This allows users to rename or alias a single command to have multiple client names.

Example:

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

hds.client.xcalcrpn.path: /FXUtilities/hds-fx/bin/xcalc

hds.client.xcalcrpn.defaultArgs: -rpn
```

Other highly-specialized attributes are provided, but should NOT generally be useful.

forceLocal

This attribute controls whether the downloadable version of the client will be run in preference to the remote host version. It is only relevant when 'transparentMode' is active for the program which is launching the client, as it is for a local window manager. If 'forceLocal' is yes, then the local client will be run, if possi-

Almost all Console shell commands can be password protected, according to the following list.

You must set passwdRequired for these commands:

```
arp         ether       help
list        log         logpg
ls          memory      netstat
nvm         opendev     ports
ps          pwd         rehash
            version
```

The passwdRequired resource is set by default for these commands:

```
db          dw          flash
reboot      restart     kill
```

These commands cannot be used with the passwdRequired resource since they are integral parts of the shell itself:

```
cd          echo        env
exit        password    setenv
unsetenv
```

You can set this resource just like with any other HDSperx client, such as:

```
hds.client.list.passwdRequired: true
hds.client.ps.passwdRequired: true
hds.client.netstat.passwdRequired: true
hds.client.opendev.passwdRequired: true
```

Another way of extending password protection to local clients is to use the configuration file resource:

```
HDSsetup.password_is_global: yes
```

which protects all Setup Mode entries and all local clients.

shareCode

This optional attribute specifies that the client’s executable code should be shared among multiple instances of the client. Each instance of the client has its own stack and set of variables, but the executable code is shared.

Default value: no.

Example:

hds.client.mwm.shareCode: yes

defaultArgs

This optional attribute specifies the default arguments to use when starting the client, if no other arguments are specified. Arguments for each client are different and are described in the man pages for each client.

Default value: none.

Example:

```
hds.client.hdsxclock.defaultArgs: -bg red -name HDSxclock
```

defaultEnv

This optional attribute specifies the default environment to use when staring the client. Environment variables specified in this way are overridden by those set directly by users. The value consists of a list of variable specifications separated by the symbol \n. Each variable name may be preceded by whitespace.

Default value: none.

Example:

```
hds.client.hdsxclock.defaultEnv: XFILESEARCHPATH=/HDS$$ROM\n\
                                 XBMLANGPATH=/HDS$$ROM/%B
```

menuName

This optional attribute specifies the menu entry to be used for the client in the automatic local window manager menu. If there is no value given for this attribute, then the client will not show up in the window manager menu.

Default value: none.

Example:

```
hds.client.hdstelnet.menuName: HDS Telnet
```

minAvailMem

This optional attribute specifies the minimum amount of available ViewStation memory usually required by the client. If the ViewStation does not have enough memory available, then the client cannot be started locally.

The value is measured in bytes. The suffixes "kb", "mb", "k", and "m" are supported for convenience. The case of these suffixes is not important.

Default value: none.

Example:

```
hds.client.hdsxclock.minAvailMem: 250kb
```

passwdRequired

If "Yes", this optional attribute indicates that the client should only be run if the ViewStation user has been able to present the ViewStation's Configuration Password before attempting to start the client. The password is entered by using the 'password' command in the HDS Console.

Default value: No

```
hds.client.ico.defaultArgs:  -p8

#

#   xclock

#

hds.client.hdsxclock.menuName: HDS Clock (Analog)

hds.client.hdsxclock.path: /FXUtilities/hds-fx/bin/hdsxclock

hds.client.hdsxclock.defaultArgs: -name HDSxclock -u 1
```

and so on. You can look at the complete file in /FXUtilities/hds-fx/config/clients.config.

2.1.4.3 Pre-loading Motif Libraries

The HDSperx clients, as well as HDSterm, use the Motif libraries Xm and Xt in their operation. These libraries can be pre-loaded when the ViewStation starts up so the clients are ready when you select them.

There are resources in the *./hds-fx/config/clients.syntax.config* file to control this preloading feature:

```
hds.library.motif.preload: yes (default value is NO)
```

In addition, individual clients have options to control their own operation and code utilization:

```
hds.client.<name>.shareCode: [ no | yes ]

hds.client.<name>.shareFree: [ no | yes ]

hds.library.<name>.shareFree: [ no | yes ]
```

These resources indicate whether the client named should share its code with other instances of the same client, or whether a separate copy of the code should be loaded. They can also specify whether the code should be freed when the last instance of the client exits. The same free characteristic applies to the libraries loaded for the clients.

2.1.5 HDSperx Client Attributes

There are a number of Client Attributes that you may want to modify. The attributes which can be set in this file include:

critical

This optional attribute specifies that the client is critical to the operation of the ViewStation and should not be suspended during low-memory conditions. This attribute should be used only when absolutely necessary.

Default value: false

Example:

hds.client.mwm.critical: true

2.1.4 Customizing HDSperx Clients

The HDSperx client configuration file automatically lists the available downloadable clients and their attributes. If you set up the ViewStation with the HDSware configuration feature, you do not have to be concerned with these files, unless you want some form of customization for these clients.

The list of clients.config file can be set in the HDS Setup client, under the "Setup Options" menu and the "Startup Options" submenu. This list can also be set with the "HDSsetup.client.config.path" entry in the main ViewStation configuration file, config.fx.

You can add or delete clients from this list (and from the root menu list) by adding or deleting clients (and the X11 extensions provided as downloadable clients) from this file. The easiest way to do this is simply insert a "#" comment character at the beginning of the "menuName" resource line for that client so it is ignored when the file is read.

2.1.4.1 Format of Client Configuration Files

The format of the clients.config file is based on the format used for standard X resource databases. However, they are NOT X resources in the normal sense; they are used only by the ViewStation’s clients.config mechanism. These entries should not be placed in normal X resource files or set to be read by **xrdb** since they would have no effect. Similarly, your Xresources should not appear in the clients.config file.

Entries in the clients.config files are self-explanatory and appear in the form:

hds.client.<client_name>.<attribute_name>: <attribute_value>

Comment lines should begin with a ’#’ or ’!’ character. Note also that C-preprocessor style comments will not work in these files.

Refer to the *./hds-fx/config/clients.syntax.config* file for samples and more details on client attributes.

2.1.4.2 A clients.config Sample File

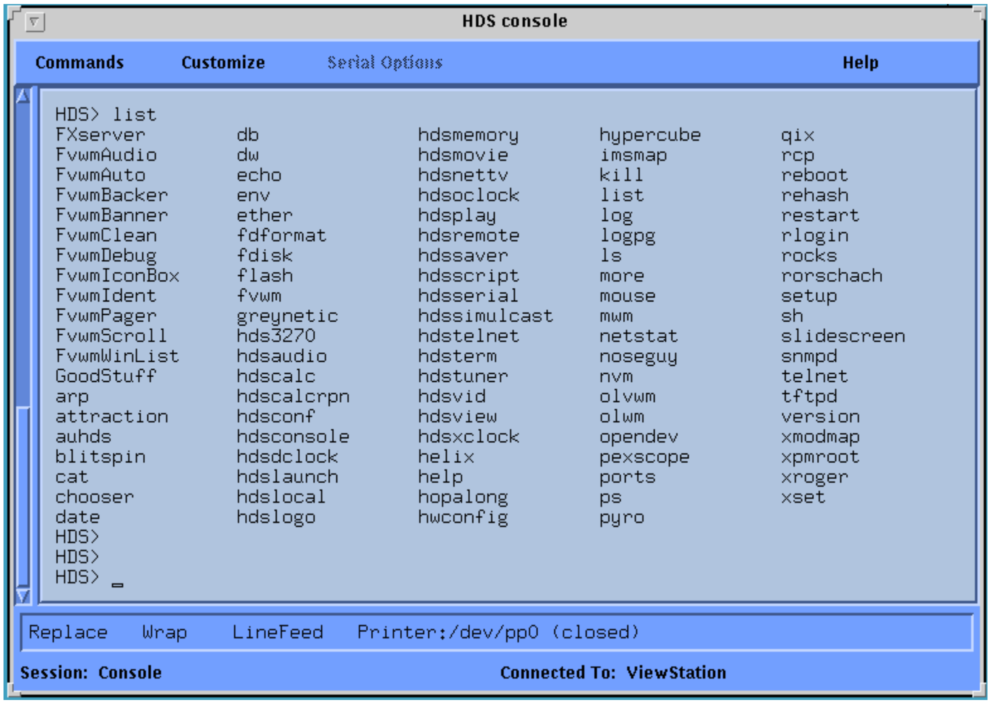
This set of entries describes the clients ico and HDS xclock. The first entry determines the name that will be used automatically for the clients in the local window manager menus. The second entry specifies the full remote host path to the client’s downloadable code. The third entry specifies the default arguments which will be used by the client if no other arguments are specified. Other resources are described below.

A section of the clients.config file is shown below:

```
#

#   ico

hds.client.ico.menuName:  Ico

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

2.1.2.2 HDSperx Client List

There are many more HDSperx clients, including some clients for Console Window operations. This list is available with the "list" command in the Console Window. It is subject to change.



```
HDS> list
FXserver        db              hdsmemory       hypercube       qix
FvwmAudio       dw              hdsmovie        imsmap          rcp
FvwmAuto        echo            hdsnettv        kill            reboot
FvwmBacker      env             hdsoclock       list            rehash
FvwmBanner      ether           hdsplay         log             restart
FvwmClean       fdformat        hdsremote       logpg           rlogin
FvwmDebug       fdisk           hdssaver        ls              rocks
FvwmIconBox     flash           hdsscript       more            rorschach
FvwmIdent       fvwm            hdsserial       mouse           setup
FvwmPager       greynetic       hdssimulcast    mwm             sh
FvwmScroll      hds3270         hdstelnet       netstat         slidescreen
FvwmWinList     hdsaudio        hdsterm         noseguy         snmpd
GoodStuff       hdscalc         hdstuner        nvm             telnet
arp             hdscalcrpn      hdsvid          olvwm           tftpd
attraction      hdsconf         hdsview         olwm            version
auhds           hdsconsole      hdsxclock       opendev         xmodmap
blitspin        hdsdclock       helix           pexscope        xpmroot
cat             hdslaunch       help            ports           xroger
chooser         hdslocal        hopalong        ps              xset
date            hdslogo         hwconfig        pyro
HDS>
HDS>
HDS> _
```

Replace   Wrap   LineFeed   Printer:/dev/pp0 (closed)

Session: Console   Connected To: ViewStation

2.1.3 HDSware Installation of HDSperx

The HDSware installation feature of the ViewStation is a simple way to utilize the HDSperx clients and menus. When you use HDSware to get started (specifying the HDSware host and path), the ViewStation automatically locates the *./hds-fx/config/clients.config* file and the HDSperx clients themselves in the HDSware directory, *./hds-fx/bin*, and automatically adds them to the window manager's menu file so they are easily accessible from a submenu on your local window manager's root menu. You must restart the ViewStation after you make the HDSware entries so the configuration file will be read; you probably also want to save these values in NVM.

2.1.3.1 Using the HDSperx Clients

When you use your window manager's root menu (move the pointer to the root window and hold down the left mouse button), you will see HDSperx as its first menu entry. When you move to this entry, another menu appears that contains a list of the HDSperx clients. Select the client you want and release the button. That's all it takes. The new HDSperx client starts and you can use it by moving the pointer to it.

You can add or delete HDSperx clients from the HDSperx menu by adding or deleting them from the *./hds-fx/config/clients.config* file. This process is discussed in detail below. For instance, you may choose not to load the tetris and maze clients, or you may want to show only one clock instead of three.

You can also modify the client attributes and extend password protection to any or all of these clients, as discussed below.

2.1.2.1 HDSperx Clients Menu

This list shows the clients normally included in the HDSperx menu. This list is subject to change.

HDS audio - a client to control the ViewStation’s audio feature

HDS calc - a TI-30 style calculator with all the normal arithmetic and trigonometric functions

HDS calc (HP style) - a HP 10C style calculator using Reverse Polish Notation

HDS clock (analog) - an analog clock with a square face

HDS clock (digital) - a digital clock

HDS clock (transparent) - an analog clock with a round face

HDS console - a Console shell window for the ViewStation

HDS memory - the Available Memory client to report ViewStation memory status

HDS saver - a screen saver and screen lock, with selectable programs for the screen display

HDS serial - an HDSterm window for serial connections

HDS telnet - an HDSterm window for Telnet connections

HDS video - a client to control the ViewStation’s video feature

HDS view - a client to display Postscript files

Clients with the “menuName” attribute in the clients.config file appear on the HDSperx menu. Names commented out of the clients.config file, such as tetris or maze, do not appear on the menu or on the list command.

HDS ViewStation Client Operations

This chapter discusses the operation of the HDS ViewStation's local clients, called HDSperx.

2.0 Section 2.0 - HDS ViewStation Local Clients

2.1 HDS ViewStation Client Operations

2.1.1 Using the HDSperx Clients

The HDS ViewStation has a powerful RISC processor and an internal Unix architecture that permits the ViewStation X terminal to run a number of important clients on its own display. These clients include the window managers, HDSterm, Setup Mode, Console window, and others.

In addition to these clients, the ViewStation can run other clients, such as a calculators and clocks. These are normal X clients that you may commonly run on your host for display on your X terminal. The HDS ViewStation lets you run these clients locally. This gives you the benefits of easy setup and fast response as well as the savings in both host CPU time and host memory, and the reduction of network traffic by having them run on the X terminal.

These HDSperx clients are included on the HDSware tape and are automatically accessible to the ViewStation when you use the HDSware start up feature. The HDSperx clients are available from the root menu of your window manager. All of this setup and configuration is automatic. The details are discussed below.

2.1.2 HDSperx Clients

There are a number of HDSperx client application programs. These are listed below with a brief description of their operation and features. They appear automatically on the root menu of your local window manager when you use the HDSware start up feature.