5. Where you would normally start your window manager, substitute the ‘hdsrunwm’ command.

For example, if you start your ‘olwm’ in the .xsession file, replace the line

```
olwm -option1 -option2 -option3
```

with

```
hdsrunwm olwm -option1 -option2 -option3
```

2.5.17 HDSrunwm Launcher Program

The 'hdsrunwm' program provides an alternate way of starting a local window manager on an HDS ViewStation. If the 'hdsrunwm' program is used on the host to start the ViewStation window manager, any window manager menu commands will also be started by the 'hdsrunwm' program. This allows these menu commands to inherit the same environment as they would if the window manager were being run on the host.

The 'hdsrunwm' program requires HDSware Version 3.0.3 or later.

To use the 'hdsrunwm' program:

1. Check the *hds-fx/contrib/hdsrunwm/execs* directory for an executable suitable for your host type. If you find one, use the 'uncompress' command on the file. For example,

```
uncompress hdsrunwm.SUN4.Z

mv hdsrunwm.SUN4 hdsrunwm
```

You may need to set the 'executable' permissions for the file. For this, you can use the command:

```
chmod +x hdsrunwm
```

2. If a suitable executable does not already exist for your host type, set the current directory to *hds-fx/contrib/hdsrunwm/src*. The 'hdsrunwm.c' source can be used to compile an executable for your host.

You will need to know the name of the Remote Shell program on your host. For many machines, this is 'rsh'. On some System V machines the program is 'rcmd'. On some HP systems it is 'remsh'. Edit the 'hdsrunwm.c' source file and set the REMOTE_SHELL_PROGRAM definition appropriately.

To compile, execute the command:

```
cc -o hdsrunwm hdsrunwm.c
```

If you are using an alternate compiler, such as GNU/C, you may need to use a command such as 'gcc'. This will produce a 'hdsrunwm' executable file.

3. Copy the 'hdsrunwm' file into a directory which will be included in your execute search path when you want to start the window manager. Possibilities include /bin, /usr/bin/X11, /usr/local/bin, etc.

4. The 'hdsrunwm' program uses the DISPLAY environment variable to determine the hostname of the ViewStation on which to start the window manager.

The 'hdsrunwm' uses the Remote Shell program to start the command. Most Remote Shell programs require a valid hostname instead of a numeric IP address. Therefore, if your Remote Shell program requires a hostname (as do most), make sure your DISPLAY environment variable will contain a hostname and not simply an IP address.

The 'hdsrunwm' program will print a warning message if a number IP address is present in the DISPLAY variable.

ViewStation's local window manager from the Setup Mode menu, none of these commands will work because only the default resources are used. An error message "No remote host running" will appear in this case. The usage of these files is the same as when a host-based window manager is used.

Further, the permissions for the ViewStation must be the same as for the remote host that started the local window manager. This means that a process valid for the host will also be permitted for the ViewStation. This permission is set in the */etc/hosts.equiv* file, where you must enter the ViewStation's name as an equivalent to the initiating host.

2) The local window manager reads (for purposes of its internal **rcmd** function) the environment files for the user specified, so the environment settings appropriate for the process must be present in the user's .cshrc file. For Bourne and Korn shells, the corresponding files are .profile, though you should double-check that these files are always read when the new shell is spawned. All commands to the remote host are prepended with the ViewStation's display setting so the output is returned to the ViewStation.

3) Different versions of OLWM (versions 1.0, 2.0, and 3.0 are generally available) use different syntax for this file. For example, the PIN and DEFAULT entries in the file may have to appear at the beginning of an entry or at the end, depending on your version. Further, the TITLE line may require the fields to be reversed, that is: "Workspace" - TITLE. Check your version and its manual pages for the correct usage. If the syntax in this file is not correct, the menus will not be read and the process will fail.

If you have problems starting remote processes, look in the Console window for error messages and consult the checklist for remote starting at the end of this chapter. You may see messages that report the rhosts command failed because of incorrect permissions, or a file was not found, or that a host is not reachable. Your host machine must be configured with permission for the ViewStation to start remote processes, refer to your */etc/hosts.equiv* file for a ViewStation entry.

2.5.16 HDS OLVWM Extensions

The HDS Network Systems's ViewStation version of **olvwm** adds the following resources to support ViewStation specific extensions.

`AutoClientMenu` *(boolean)*

Indicates whether **olwm** should automatically add the HDSperx menu to the top of the Workspace Menu. The HDSperx menu lists tools which can run locally on the ViewStation.

Default value: true.

`AutoClientMenuName` *(string)*

Specifies the name of the HDSperx menu to be used if AutoClientMenu is true.

Default value: HDSperx.

HDS **olvwm** also adds a new Workspace menu customization keyword: HDS-MENU, which presents a menu of HDS ViewStation downloadable clients.

2.5.15 OLWM Version 1.0 Syntax Differences

This note discusses the syntax used with OLWM version 1.0 which was used with ViewStation server code prior to version 2.1. Note the differences and modify your menu configuration files appropriately. You can refer to the OLVWM version 3.0 manual pages included on the FX Utility tape for complete details.

The OLWM version 1.0 allows the user to start processes from window manager menus as defined in an .olwmmenu configuration file. The sample file below shows some simple examples. Details are available from the OLWM manual pages, including specifications for labeling and syntax.

```
# OLWM Menu items
# Note: this menu refers to OLWM version 1.0

TITLE                           "Workspace"
PIN

DEFAULT "Programs"              Menu
    DEFAULT "xterm"             "xterm -fn 10x20 -g 80x32 &"
    "Text Editor..."            textedit
    "File Manager..."           filemgr
    "Mail Tool..."              mailtool
END

"Utilities" MENU
    DEFAULT "Refresh"           REFRESH
    "Save Workspace"            SAVE_WORKSPACE
    "Lock Screen"               xlock
END

"Properties..."                 PROPERTIES

"WM Exit..."                    WMEXIT
"Exit..."                       EXIT
```

There are three things to note:

1) You must start the ViewStation local window manager from a remote host in order to use these resource specification files for the local window manager. This remote host and the user will be the host and the user for starting remote processes, so the appropriate permissions and accounts must exist. If you start the

files with both names and locations, and display error messages when they occur. These are some common errors:

1) "RCP failed" - this message typically points to problems with permissions. Check /etc/hosts.equiv and path variables, or perhaps nameservice failures.

2) "Error: Remote shell 'env' command to user@host contains an invalid line." - this message typically points to path or environment variable problems. It may be that the .cshrc file (or its equivalent) has a command that is not understood or has incorrect syntax.

3) "Error executing remote shell env command to user@host." - this message usually indicates that there is a problem with permissions or getting to the correct files.

4) "RCP protocol screwup" - this message can refer to any of the permission, path, or environment variable problems.

5) "Login incorrect" or "Permission denied" - these messages typically indicate a problem with the /etc/hosts or /etc/hosts.equiv files, or the .rhosts files if they are used.

- Suggestions for fixes

The basic task is to isolate the problem. Simplify the configuration as much as possible. Use a single host, single host name, and a single user. Rename the user's .cshrc and use a minimal file in its place. Rename the resource files and use a minimal file in their places. Use the Console window messages to track the loading process and file locations. Reduce your system configuration to the minimal configuration and try to run the ViewStation's window manager before you add the more complicated elements.

A quick and simple test for the "rsh" command is to enter:

```
rsh <localhost> env
```

which will show you the path variables and environments which are set from your .cshrc file. If variables are missing from this response, they will also be missing for the local window manager started by rsh.

Another simple test is to do an "rcp" between two hosts on the network to test its operation independent of the ViewStation and its file transfer requirements.

The ViewStation's requirements are simple: the ability to start a remote shell and do a remote copy. Added features, like nameservices and configurations, or special user environments, can make this simple requirement into a complicated set of tasks.

There are many situations and problems of this type. Almost without exception, the problems can be traced to incompatibilities within the system configuration. HDS Technical Support people will be glad to help you examine the problem, but they are not experts on your system. Use your system administrator and your network utilities to try to trace the problem.

3) Domain Name Service (and/or NIS or Yellow Pages) - The ViewStation must have a name (not just an IP address) on file and the nameservice lookup functions must work correctly from all devices and in all directions.

- Resource File Requirements

The “rsh” command (or its equivalent on other systems, such as “rcmd” on SCO systems and “remsh” on HP systems, etc.) starts a shell. This shell must have correct environment variables, paths, and so on, in place or it cannot locate the files it needs for operation. Different systems and different users may have this information in different places. You must insure that this information is available to the local window manager.

1) For the C shell - the .cshrc file should contain all the pertinent path and environment variables. The .cshrc file is read each time the “rsh” command is given, so this file is the desired location for them.

2) For the Bourne shell - the .profile file should contain all the pertinent path and environment variables. The .profile file is read when the “rsh” command is given, so this file is a good location for them. A .login file is not a good choice for these variables.

3) For the Korn shell - the .profile file should contain all the pertinent path and environment variables. You must also put the path and environment variables in some system-wide location where they will be read whenever the “rsh” command is given, perhaps invoking the command from a script which explicitly finds and reads them.

4) Resource files and locations - resource variables for the window managers can have many locations, such as .Xdefaults, /usr/lib/X11/app-defaults, .Xresources, .mwmrc, .openwin-menu, etc. You must insure that these files are read, which means both that the locations are correctly specified and also that the “rcp” command succeeds. The ViewStation attempts to read these files in a number of locations; use the Console window messages to see the file names and locations it looks for. These may be different than the locations your host used for host-based operation of the window managers.

Note that the shell startup files may not produce any output messages or the “rcp” command will fail and an “RCP protocol screwup” message will be sent. This is a restriction inherent in the “rcp” protocol.

5) Resource file contents - the contents of the resource file may cause problems. Comment lines, variable names, etc. must conform to the window manager’s format and syntax. For example, resources written as C code cannot be imported without some syntax adjustments. These are often tricky problems to find and may depend on syntax differences between window manager versions. Check the **man** pages of the window managers; the ViewStation uses full, licensed copies of the window managers, so all resources and syntax are fully supported.

- Common Error Messages

The ViewStation Diagnostic messages (or Console Window) are a valuable tool in tracing problems and errors. The Diagnostic messages report rcp attempts to read

These unique and powerful features all depend on the success of the initial “rsh” command and the successful operation of nameservice and other network services. If these processes fail, the local window managers cannot gain the access they require.

This checklist shows a few things you can check if you have problems with this process. It is suggestive of places to look for problems. It is not specific about many items because the names and syntax of files can vary from system to system.

- Necessary Network Services

1) /etc/hosts file - The ViewStation must have a name entry here on its “rsh” host.

For a Sun host, the /etc/hosts file looks like this:

```
# Sun host database
# If the NIS is running, this file is consulted only when booting
#
128.91.3.12    hdssun
#
128.91.6.5     mikegfx
128.91.6.10    davidfx
```

and so on.

2) /etc/hosts.equiv - The ViewStation must have specific equivalency with its primary host (the source of the “rsh” command). Often this *letc/hosts.equiv* file is empty or has only a “+” entry; it should have a specific entry for the ViewStation using the rsh command. Check your **man** pages for the correct form and syntax for this file. For a Sun host, the */etc/hosts.equiv* file looks like this:

```
# hosts.equiv file
# See /etc/hosts for a list of valid names
hdssun1
hds486
davidfx
+
```

If you are using an .rhosts file, exchanging permissions is a little more complicated and sensitive since "Trusted Access" is implemented in different ways by different systems. The system administrator should be familiar with how this works on your system. Each user may have an .rhosts file in their home directory which contains the names of hosts equivalent to the user. The file itself contains a host name and a user name for each user that is to be granted equivalent status with the home user.

"Quit..."	WMEXIT
"Exit..."	EXIT

and so on.

If you are making changes to the HDSperx clients' configuration, you can see the changes by using the "rehash" command in the Console shell window. This rehash command forces a reading of the configuration files, so olvwm responds to the new settings.

2.5.13.2 HDSperx Menu Contents

The HDSperx menu is a list of the ViewStation's local clients that you have selected in the ViewStation's *clients.config* file. You can add or delete clients from this menu by uncommenting or commenting their client menuName lines in the *clients.config* file.

Note that you must specify HDSperx processes with a "hdslocal" prefix to indicate that they should run on the ViewStation. In the example above, the xcalc will run on the host computer, but the hdsdclock will run on the ViewStation.

2.5.13.3 HDSperx Menu Handling

The menu itself is handled differently depending on your window manager startup procedure.

- **ViewStation window manager with remote start** - If you start the ViewStation's OPEN LOOK window manager using a remote shell command from the host computer, you get the OLWM menu specified in your *.openwin-menu* file with the addition of the HDSperx menu for the ViewStation's local clients. This merging of the menus is done automatically by the ViewStation.
  This is the best option, since it includes all your menus. You can use the remote shell startup mechanism from an *.xsession* file or from your *.profile*, which are read as you login. This is discussed in detail below.
- **ViewStation window manager with local start** - If you start the OPEN LOOK-window manager by clicking on the ViewStation's OLWM icon (or the Autostart button) on the Main Menu of Setup Mode, your OLWM root menu is the HDSperx menu (along with refresh, exit, and other commands). The customization options described here don't apply; you must use the rsh method of starting to get them.
- **Host-based window manager** - If you use the host to run the OPEN LOOK window manager, you get the OLWM menus specified in the *.openwin-menu* file. You could manually add the HDSperx menu to this file, but you would have to make sure the "exec" command syntax was correct for the ViewStation's local clients.

2.5.14 Window Manager Remote Starting Checklist

Starting the ViewStation's window managers from a remote host with the "rsh" command allows the local ViewStation window manager access to its resource files on the host and permits "rcp" and "rcmd" access to the network.

```
 “Print Tool...”          exec $OPENWINHOME/bin/xview/printtool
“Programs” END PIN

“Utilities” MENU
 “Refresh” DEFAULT        REFRESH
 “Reset Input”            POSTSCRIPT /resetinput ClassUI send
 “Save Workspace”         SAVE_WORKSPACE
 “Lock Screen”            exec xlock
 “Console...”             exec $OPENWINHOME/bin/xview/cmdtool -C
“Utilities” END

“Properties...”           PROPERTIES

“Restart...”              RESTART

“Exit WM...”              WMEXIT

“Exit...”                 EXIT
```

The OLVWM window manager can also start HDSperx clients explicitly, without using HDSMENU. For example, you can start a local HDSdclock with an .openwin-menu file containing menu entries:

```
# OPEN LOOK Virtual Window Manager Menu
"Workspace" TITLE
"Winmenu..."                    WINMENU
"Programs" MENU
   "XTerm..."DEFAULT            exec xterm
"Programs"                      ENDPIN

"Calculator"                    exec xcalc
"Clock"                         exec hdslocal hdsdclock
"HDS Local Clients..."          HDSMENU

"Properties..."                 PROPERTIES
"New Menu..."                   REREAD_MENU_FILE
"Restart..."                    RESTART
```

remote processes, refer to your */etc/hosts.equiv* file or the *.rhosts* files on each machine for a ViewStation entry.

2.5.13 Using HDSperx Clients with OLWM

The HDSperx clients are set up to run from a root menu submenu automatically. Their operation from the menu is simple, since they start with just a mouse click on their entry.

Control of the HDSperx submenu and its name is handled by two olvwm resources:

```
olvwm*autoClientMenu:  True
olvwm*autoClientMenuName:  HDSMENU
```

These are both olvwm resources and should be used with other olvwm resources in .Xdefaults or .olvwmrc. (Resources for the HDSperx clients themselves are HDS resources and should be used only in the HDS client configuration file.)

If you want to load the HDSperx submenu automatically, use "olvwm*autoClientMenu: True"; this is the default setting. If you want to move the menu from its top position, you should set the "autoClientMenu" to False and locate the tools menu where you want it. You can also use the "autoClientMenuName" to enter a new menu name, which might be "HDS Local Clients". By default, the ViewStation’s local OLVWM has an internally defined menu called "HDSMENU" which can be explicitly referenced in your .openwin-menu file menu table.

2.5.13.1 OLWM .openwin-menu Sample File

This is a sample .openwin-menu file. Use it as a reference when you construct your own menu file.

```
#
# openwin-menu - OpenWindows X11/NeWS Server default root menu file.
“Workspace” TITLE
“Programs” MENU
 “HDS tools”             HDSMENU
 “small xterm”           exec /usr/bin/X11/xterm -fn 8x13b
 “xterm”                 exec /usr/bin/X11/xterm -geom 96x35
 “big xterm”             exec /usr/bin/X11/xterm -fn 12x24
 “xload”                 exec /usr/bin/X11/xload
 “Text Editor...”        exec $OPENWINHOME/bin/xview/textedit
 “File Manager...”       exec $OPENWINHOME/bin/xview/filemgr
 “Mail Tool...”          exec $OPENWINHOME/bin/xview/mailtool
 “Calendar Manager...”   exec $OPENWINHOME/bin/xview/cm
 “Clock...”              exec $OPENWINHOME/bin/xview/clock
 “Calculator...”         exec $OPENWINHOME/bin/xview/calctool
```

openwin-menu-utilities, in the same directory). Just make these entries and save the modified file. It is good practice to save a copy of the original file under another name.

2.5.12.1 Starting Clients with OLWM

The OPEN LOOK Virtual Window Manager provides a facility allowing the user to start processes from window manager menus as defined in an .openwin-menu configuration file. The sample file shown in the next section gives some simple examples. Details are available from the OLWM manual pages, including specifications for labeling and syntax; refer also to the .olvwmrc configuration file.

There are three things to note about the openwin-menu file:

1) You must start the ViewStation local window manager from a remote host in order to use these resource specification files for the local window manager. This remote host and the user will be the host and the user for starting remote processes, so the appropriate permissions and accounts must exist. If you start the ViewStation's local window manager from the Setup Mode menu, none of these commands will work because only the default resources are used. An error message "No remote host running" will appear in this case. The usage of these files is the same as when a host-based window manager is used.

Further, the permissions for the ViewStation must be the same as for the remote host that started the local window manager. This means that a process valid for the host will also be permitted for the ViewStation. This permission is set in the */etc/hosts.equiv* file, where you must enter the ViewStation's name as an equivalent to the initiating host. If you are using an *.rhosts* file, the appropriate permissions must be entered there as well. Refer to Section 1.11 for more details.

2) The local window manager reads (for purposes of its internal **rcmd** function) the environment files for the user specified, so the environment settings appropriate for the process must be present in the user's .cshrc file. For Bourne and Korn shells, the corresponding files are .profile, though you should double-check that these files are always read when the new shell is spawned. All commands to the remote host are prepended with the ViewStation's display setting so the output is returned to the ViewStation.

3) Different versions of OLWM (versions 1.0, 2.0, and 3.0 are generally available) use different syntax for this file. For example, the PIN and DEFAULT entries in the file may have to appear at the beginning of an entry or at the end, depending on your version. Further, the TITLE line may require the fields to be reversed, that is: "Workspace" - TITLE. Check your version and its manual pages for the correct usage. If the syntax in this file is not correct, the menus will not be read and the process will fail.

If you have problems starting remote processes, look in the Console window for error messages and consult the checklist for remote starting at the end of this chapter. You may see messages that report the rhosts command failed because of incorrect permissions, or a file was not found, or that a host is not reachable. Your host machine must be configured with permission for the ViewStation to start

In this example, the SetInput resource sets the mouse focus policy. The "select" value (note the lowercase) means that you click in a window to give it focus; the "followmouse" value means that focus follows the mouse position.

If you change the resource file, you must reload the resources (check your currently set resources with the appres program described in Section 3.4), then exit and restart OLVWM for the settings to take effect.

2.5.12 OPEN LOOK Window Manager Menu

The menu for the OPEN LOOK Window Manager is a file called "openwin-menu", which is found in the *./openwin/lib* directory. There are a number of sample and submenu files there. Look at them to see their structure and syntax.

The openwin-menu file contains:

```
#

# @(#)openwin-menu 23.15 91/09/14 openwin-menu

#

# OpenWindows default root menu file - top level menu

#

"Workspace"       TITLE

"Programs"        MENU $OPENWINHOME/lib/openwin-menu-pro-
grams

"Utilities"       MENU $OPENWINHOME/lib/openwin-menu-utili-
ties

"Properties..."   PROPERTIES

SEPARATOR

"Help..."         exec $OPENWINHOME/bin/helpopen handbooks/
top.toc.handbook

"Desktop Intro..." exec $OPENWINHOME/bin/helpopen handbooks/
desktop.intro.handbook

SEPARATOR

"Exit..."         EXIT
```

The menu file consists of a label (which appears on the menu) and the command that it executes. Note the use of submenus (there are other submenus, such as the "demo" entries, in the *./openwin/lib* directory that are not used). Note also the use of "SEPARATOR" entries to provide visual organization for the menu.

Modifying the Menu

If you want to modify this menu, just open the file with a text editor and make the desired changes. You may want to delete the "Desktop Intro..." entry, or add or delete some utility programs to the "Utilities" menu (this menu is in its own file,

2.5.8 Exiting OLVWM

You can exit the OPEN LOOK Virtual Window Manager by selecting the "Exit WM" line in the menu. Clicking on the last line "Exit" closes all of your windows and connections.

2.5.9 Starting Local Processes from the Local OLVWM

When you use the ViewStation's local OLVWM window manager, a submenu for the ViewStation's HDSperx clients is automatically added to your root menu. This submenu lists all the HDSperx clients you have loaded and you can start them simply by clicking on them.

The details of making changes to this menu and setting it up manually are given at the end of this chapter. Refer also to Section 2.1 on using the HDSperx clients.

2.5.10 Starting Remote Processes with Local OLVWM

The ViewStation runs OLVWM locally on the server. The OPEN LOOK Virtual Window manager may be started from a remote host using the **rsh** (BSD Unix) or **rcmd** (System V Unix) command as described in the previous section. Starting processes on other hosts is done in the same way as with a remote window manager, using the appropriate configuration file. The sample OPEN LOOK configuration files for the window manager menus below show the process. This file resides in the user's home directory.

Note that the OPEN LOOK environment variables and paths must be in the user's .cshrc or .profile file, since it is that shell file (not your .login file) that is read when a remote shell process is started. Having these variables in another location, such as the .login file or in a separate shell script, generally means that the remote shell process fails.

2.5.11 Customizing OLVWM

You can use a number of OLWM resources to customize the appearance and operation of OLWM. A complete list of the OLWM resources can be found in the OLWM man pages.

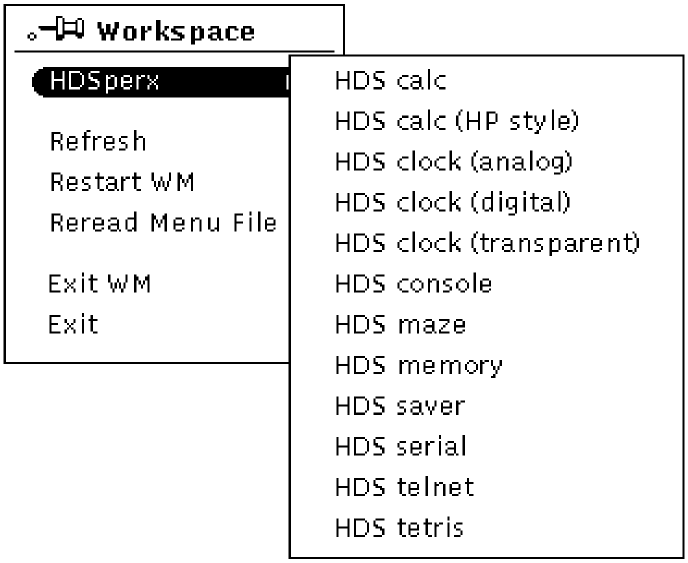
Note that these resources must be loaded into the ViewStation (usually with the **xrdb** utility) before the local OLVWM is started. These examples show some simple OLVWM resource settings:

```
# Mouse focus policy - either followmouse or select

OpenWindows.SetInput:     followmouse

# Select the window colors

OpenWindows.WindowColor:     Pale Turquoise

OpenWindows.WorkspaceColor:     White

# Virtual Desktop size

OpenWindows.VirtualDesktop: 1x1
```

Normal operation of pull-down menus is that they disappear as soon as you release the mouse button that brought them up. You'll notice a little push pin in the top left corner of each menu along with a small hole next to it. If you move the push pin over to the hole, using the mouse pointer, it will pop into the hole and pin the menu onto the screen permanently (or at least until you pull it out of the hole). Having the menu constantly visible is convenient if you use its operations frequently.

2.5.7 Using HDSperx with OLVWM

The HDSperx menu is added to your root menu automatically. This allows you to start HDSperx clients easily. The menu is shown below.



Title Bar - This is the block showing the title you (or the client) gave to this window.

Resize corner - These are the handles to resize the window.

Iconify button - This button reduces the window to an icon.

2.5.5.6 Moving a Window

You can move windows around on the screen to make your work more efficient. This is a primary function of the window manager. To move a window, place the mouse pointer in the title bar or in one of the window borders and press the left button down, then drag the mouse pointer to the new location of the window. An outline of the window will follow your movements. When the position of the window is correct, simply release the mouse button and the window appears at its new location.

2.5.5.7 Resizing a Window

You can change a window's size to make your work easier. Imagine that the window has elastic sides. By grabbing a corner (point the mouse on a corner and press the left button and hold it down), you can stretch the window to a larger size or let it contract to a smaller size. Dragging the lower left corner stretches the left and bottom sides (with the upper right corner remaining fixed). Try resizing a window - it's easier to learn by doing than by reading.

2.5.5.8 Iconifying and Deiconifying a Window

You can gain space on your screen by changing a window into an icon. The window will continue to function if it has a process running, but it is temporarily hidden from view. To iconify a window, simply move the pointer to the Iconify button and click on it. The window changes to an icon and moves to a row of icons, usually at the bottom of the root window. You can move the icons (they are windows), or define a different place for the icon row.

The icon itself has an image and a label so you can distinguish them from each other. To deiconify an icon, move the mouse pointer to it and double click. The window will be restored to its original size and location on the screen.

2.5.6 OLVWM Menu

Pressing the right mouse button brings up the OLWM Menu. It allows you to refresh the screen, to restart the window manager, to reread the Menu file, to exit the window manager, or to exit your entire session. The menu is shown below:

2.5.5.4 Selecting a Window with OLVWM

When you have many windows on the screen, you move from one to another by moving the mouse pointer and clicking (with the left button) on the window to select it. You will notice that the border of the window changes color or shade as an indication that it is the currently selected window. This is called getting window focus; it means that data from the keyboard or mouse enters that window. Giving a window focus also means that it moves to the top of the stack, so if any other windows were covering it, they are moved down.

The OPEN LOOK Virtual Window Manager can give focus to a window in two ways - either by placing the pointer in the window and clicking on it (this is the default choice), or by simply moving the pointer into the window. There is a configuration selection (called a resource) that selects this mode of operation. Setting this resource will be discussed later in this chapter; general descriptions of resource handling are in Section 3.4. If you point and click to give a window focus, that window retains focus regardless of where the mouse pointer is located. Many people prefer this type of operation because they can work in one window and ignore the mouse position. Others prefer to be able to get window focus by moving the mouse only, without the need to click when they are in the window.

This is a significant difference in the way a user controls the screen. The operation of the window manager controls this function and many other significant operations depending on its configuration. These questions of configuration are important and are discussed at the end of this chapter.

2.5.5.5 Window Decoration

The border around the window, called its decoration, has areas marked that permit easy manipulation of the window. These are labeled below, along with brief descriptions of their operations.

```
hds486 > []
```

2.5.5 Using the OLVWM Window Manager

The OPEN LOOK Virtual Window Manager is a graphical user interface that lets you handle your windows by pointing with the mouse and clicking to select various window operations. This simple process applies to all of your windows, regardless of the host running them or the contents of the windows themselves.

The HDS ViewStation has the OPEN LOOK Virtual Window Manager running as a local client. This should not be confused with the OPEN LOOK environment which was designed to use an entire range of functions, including the File Manager program, editing capabilities, and so on. These functions can be used, but they must be added as part of your host environment before you can use them from within your window manager.

2.5.5.1 Pointer Shapes

The mouse cursor, called the Pointer, indicates the kind of operation you are currently using. In OPEN LOOK, the pointer is an arrow. There are other pointer shapes to give you current information, like a watch cursor when a process is busy, or a question mark when a selection isn't appropriate for that situation.

2.5.5.2 Mouse Buttons

You can use either the standard three-button mouse or an optional two-button mouse or optional trackball. The three mouse buttons behave differently; but their operation is usually just a simple click (a single press and release), a double click (two clicks in rapid succession), or click and drag (pressing a button and holding it down while you move the mouse pointer).

Different operations are selected by the left, middle, and right mouse buttons. If you have a two-button mouse, the middle button operations are selected by pressing both buttons at the same time. (The actual configuration of the mouse buttons, such as swapping left and right buttons, can be modified by the **xmodmap** utility, as described in Section 1.8.3.1 of this manual.)

OPEN LOOK has made some default assignments for the mouse buttons. The three button mouse has these assignments for its buttons:

Left - SELECT - this selects objects or controls

Middle - ADJUST - this extends or reduces the number of selected objects

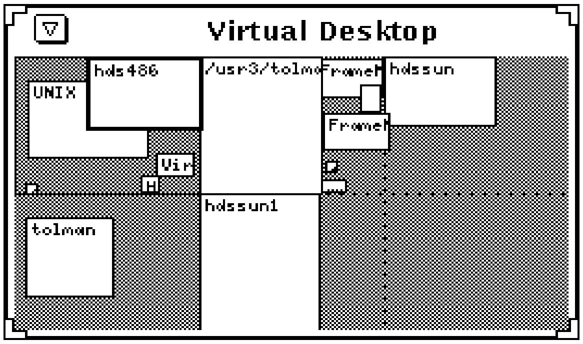
Right - MENU - this displays and chooses menus

2.5.5.3 Mouse Buttons and Modifier Keys

The operation of mouse buttons is affected by any modifier keys, such as Shift, Ctrl, and Alt, you have set. For instance, selecting the header bar of the window by pressing the left mouse button to move it won't work if you have the Ctrl or Alt key depressed. This is especially frustrating (and often overlooked) with other modifier keys, such as Caps_Lock and Num_Lock. If you find unexpected problems with your window manager operation, check the modifier keys. For instance, OPEN LOOK uses the Caps_Lock key to invoke micropositioning for the mouse, so you will see much slower mouse tracking.

2.5.4 OPEN LOOK Virtual Window Manager Operations

The OPEN LOOK Virtual Window Manager is an extension of the basic OPEN LOOK window manager. Its operations are the same. The only difference is that the Virtual window manager works on a desktop, that is, a virtual screen that is larger than the actual screen size. By default, its virtual screen is six screens arranged in a tiled pattern. A small window appears in the upper left corner that shows these six screens with small representations of the windows currently displayed in each screen tile. Where possible, the names of the windows are shown, though the names are often truncated.



You can move a window from one screen tile to another by clicking and dragging it, or by starting it in that window using your window manager’s menus. For instance, you might want to keep your xterm and Setup windows in screen tile 1, your database windows in screen tile 2, and your Framemaker documents in screen tile 3, or whatever arrangement you want. You can move from one screen to another by moving to the Virtual window manager window (which appears in all screen windows) and double-clicking on the screen tile you want.

The Virtual Desktop window can be resized to increase or decrease the number of screen tiles. It is a convenient way to organize your activities and manage a large number of clients and windows. If you want to use OLWM (not using the Virtual Desktop feature), you can use an OPEN LOOK resource to restrict the Virtual Desktop to a 1x1 tile. The OPEN LOOK resource is:

OpenWindows.VirtualDesktop: 1x1 (default is 3x2)

Use this resource in your .Xresource file or in any of your resource files. Note that OPEN LOOK resources may use either:

OpenWindows.resource: value

or

olvwm.resource: value

Both resource names are accepted but the OpenWindows syntax is preferred if the resource applies to Open Windows clients other than the window manager.

The rsh command line can be entered from any location that will start a remote shell, such as your .xsession file or .login file. Configuration options for the window managers, such as applicable resources from .Xresources, or from .openwin-menu, etc. should be read before the remote shell command is given.

When the window manager is started from a remote host, the ViewStation attempts to read any applicable resource files, such as, /usr/lib/X11/app-defaults/olwm, /usr/home/username/.Xdefaults, /usr/home/username/.openwin-menu, and so on with the **rcp** (remote copy) command. If one of those files isn't located, the ViewStation will timeout and search for the next one; using valid entries in these files, or using them with null entries, will reduce the timeout pauses. You can follow this process, and locate all the file names, by monitoring the Console shell messages.

2.5.3 The OPEN LOOK Environment

OPEN LOOK GUI (Graphical User Interface) has been designed as a total environment, including its own application programs, editors, and so on. These things are not automatically included with the local OLWM, since they are not local functions but run remotely on the host computer. These application programs are available on your host machine, typically a SUN machine, but they require that you set a number of paths and environments so they can be located and used.

The following is a simple shell script to set those environments. You might label this script do.olwm and put it in your home directory. If you are using the local window manager to start remote processes, these settings must be in your .cshrc file. When you start a session, you could start OLWM by typing the do.olwm command, or you could have it executed automatically by xdm, the X display manager.

This sample script starts the OLWM remotely and contains some environment settings and the command to start the window manager (if you want to set these things for the local OLWM, you should enter them in your .cshrc file in your home directory):

```
setenv OPENWINHOME  /usr/openwin

setenv LD_LIBRARY_PATH  $OPENWINHOME/lib

setenv MANPATH  $OPENWINHOME/share/man:/usr/man

set path=( $OPENWIN/bin  $OPENWINHOME/bin/xview  $path )

echo "Starting OPEN LOOK Virtual Window Manager"

echo "OLWM DISPLAY = $DISPLAY"

$OPENWINHOME/bin/olvwm &
```

This example uses simple settings and "default" paths and directories; your installation may use different locations and directories. This shell script must be executable and should be in your home directory.

suggestive and you should refer to the OPEN LOOK reference materials for details.

2.5.1 Starting OPEN LOOK Virtual Window Manager

You can start the OPEN LOOK Virtual Window Manager by selecting the OPEN LOOK Window Manager icon from the Main Menu of Setup Mode and clicking on it. You will notice that the window manager puts a border around all the windows currently on your screen. Any new windows will automatically be controlled by the OPEN LOOK Virtual Window Manager.

If you use resource customizations or set OLWM environments, these must be loaded into the ViewStation server before the local OLVWM is started. Setting resources is discussed in Section 3.4 and some OLWM environment settings are discussed in the next section.

2.5.2 Starting Local OLVWM Remotely

Normally, you just click on the OPEN LOOK Virtual Window Manager option on the Setup Menu to start it. Many times it is more convenient to start the local OLVWM from an Xsession file or from a remote host login file. Starting the OLVWM from a remote host copies your customized window manager files, such as .openwin-menu, to the local window manager's operation.

The local ViewStation OPEN LOOK Virtual Window Manager can be started with the rsh (remote shell command) as follows:

rsh -n <*hostname*> <*olvwm*> [arguments] '&'

(you can use *olvwm* or *olwm*; both names are supported)

(your operating system may use different syntax; SCO uses **rcmd** and HP uses **remsh**, for example)

The remote shell command must use the ViewStation's hostname as it is entered in the */etc/hosts* file (the **rsh** command does not accept an IP address). You should use the -n option (null input) so the process exits properly when you close the window manager. The next entry is the window manager you want to start, either mwm or olwm. Only one window manager can be running at one time. If you try to start a window manager when another is running, the remote shell command will fail but there will be no error message displayed except in the ViewStation's Console window. The last argument "'&'" allows the rsh process to exit cleanly when the window manager starts.

Command line arguments for the window manager can be entered on this line. For instance, entering the -display argument would permit you to run the ViewStation's window manager on another display device. The remote shell also accepts command signals, such as Ctrl-C, which could be used to stop the window manager after it has started. If you use resource names on the command line you must take care to quote and escape them properly; for example, the operating system would misinterpret the * (asterisk) in the resource name as a wildcard character.

OLVWM - OPEN LOOK Virtual Window Manager

This chapter describes the operation of the OPEN LOOK Virtual Window Manager. Some examples of customizations are also included.

Section 2.0 - HDS ViewStation Local Clients

2.5 OPEN LOOK Virtual Window Manager

The HDS ViewStation can download the OPEN LOOK Virtual Window Manager to run locally with its server code. Effective with ViewStation server code version 2.1, this OPEN LOOK Window manager is version 3.0 (replacing the version 1.0). OPEN LOOK version 3.0 supports the "virtual window manager" extension, which is described later. You can use either "OLWM" or "OLVWM" on your command lines to start olvwm; the terms are used interchangeably in this manual, though only OLVWM is actually run. If you don't want to use the Virtual window manager, you can resize its window to a single screen tile size.

Since the OPEN LOOK Virtual Window Manager is an authorized version, not a copy or a facsimile, you can use all the standard commands, resources, and reference materials that apply to the OPEN LOOK Virtual Window Manager, but be careful with differences between OLWM and OLVWM versions.

This chapter gives a brief overview of the OPEN LOOK Virtual Window Manager. The OPEN LOOK Virtual Window Manager has the same functions as OLWM, but manages a virtual desktop that is larger than the actual screen. There are some simple exercises to help you learn what symbols perform which operations, and so on, but you should refer to individual OPEN LOOK reference materials for full details. The OLVWM man pages and supporting example files are found on the HDSware tape in */hds-fx/man*.

This chapter also contains sample files for OPEN LOOK resources and some suggestions for customizing the OPEN LOOK Virtual Window Manager for your system environment and your personal preferences. Again, this material is only