# HP Visual User Environment 3.0
# User's Guide

# HP Visual User Environment 3.0

# User's Guide

# Introduction

HP VUE is shorthand for Hewlett-Packard Visual User Environment. Assuming you already know what Hewlett-Packard is, that leaves "Visual User Environment" to be explained.

**HP VUE is an environment and a set of powerful applications**

HP VUE creates an environment for interacting with your workstation. The environment it provides is called a "visual" or "graphical" environment because it provides things you can see on the display that guide you through performing tasks. HP VUE also includes a set of applications that make it easier to interact with your workstation and extend its functionality.

**HP VUE is an alternative to commands**

Computers understand commands, and commands often have complex rules and syntax. In the absence of HP VUE, you must learn the commands and type them in exactly the right way to accomplish a task. When you are running HP VUE, you can issue commands to the computer without having to learn the commands.

**HP VUE provides controls you can maniputate**

In HP VUE, commands are represented by controls drawn on the display that look much like buttons on an instrument panel. You use the mouse or keyboard to "push" the buttons. Since you must be able to find all these controls, the designers also fashioned a number of standard control containers such as menus and dialog boxes.
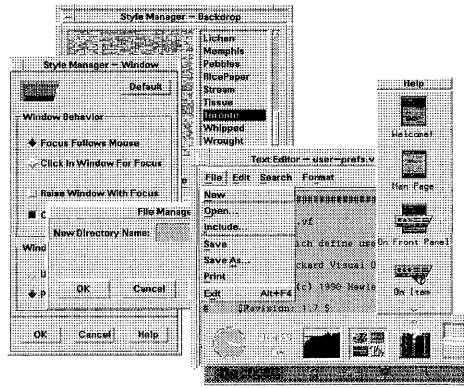
**HP VUE uses many different types of controls.**

**HP VUE extends the functionality of your computer**

In addition to making it easier to get things done, HP VUE extends your computer's capabilities. For example, in an ordinary command-driven environment, you have one command line. With HP VUE's windows, you can have many applications running at once.

HP VUE also extends some physical limits of the computer. For example, most systems have only one physical display. HP VUE makes the most of that display by providing multiple workspaces. Each workspace occupies the entire screen, and you can switch from one to another.

**HP VUE is part of your workstation's open system**

HP VUE is one component in a layered, open system. An open system can be adapted and extended to meet your needs. Since the building blocks of HP VUE use industry standards, you can easily integrate other applications into it, providing a way to start the application in its own window without having to type a command.

**HP VUE includes a set of applications**

In addition to providing you with an environment that provides windows, workspaces, and easy-to-use controls, HP VUE includes a set of applications that help you use and customize your workstation. For example, Text Editor and Icon Editor provide an easy way to edit text files and create your own icons; Style Manager is an application for setting your own personal color scheme, fonts, and other aspects of the appearance and behavior of your system.

**HP VUE is flexible**     There are two forms of HP VUE.

Depending on your system and your needs, you can run:

- HP VUE sessions. These provide all the features of HP VUE.

- HP VUE Lite sessions. These are scaled-down versions of HP VUE that run run faster and require less memory.

You choose which type of session you want at the login screen.

# In This Book

This book explains how to use and configure the HP Visual User Environment, version 3.0.

Before using this book, you should examine the HP VUE Quick Start Guide. It gives you an overview of the product and a brief introduction to its major components.

This book is divided into two parts:

Part 1: Using HP VUE    Chapters 1 through 10 explain how to use HP VUE to perform common tasks such as logging in and out, manipulating and editing files, and customizing your workstation.

Part 2: Configuring HP VUE    Chapters 11 through 29 cover advanced tasks in customizing the appearance and behavior of HP VUE. Many of the tasks involve editing configuration files; some require you to log in as superuser ("root").

**Getting Help**

Most of the information in this book is also available as online help. You can get help on a particular HP VUE application (for example, Style Manager) using its Help menu.

To browse all of the HP VUE online help:

1. Choose the Help control in the Front Panel (the icon with the books and question mark). This opens Help Manager, which lists all of the online help installed on your system.

2. Find the underlined title "HP Visual User Environment, Version 3.0." Choose this title to list all of the HP VUE online help.

3. Each underlined title represents a *help volume* for an HP VUE application or general subject area. Choose any of the underlined titles to display the online help.

For more information about getting online help and using help windows, refer to Chapter 4, "Getting Help."

# Contents

## 5. Managing Files and Using the Desktop

## Part 2. Configuring HP VUE

**21. More About Actions**

## 22. Networking and Distributed Computing

## 28. For Previous HP VUE 2.01 or X11 Users

# Contents

# Part 1

# Using
# HP VUE

# 1

## Introducing Your
## Workstation

# Introducing Your Workstation

Your HP workstation uses the HP-UX operating system and the HP Visual User Environment (HP VUE). HP-UX is a versatile operating system that you can use to run application programs and perform a variety of tasks. HP VUE is a powerful graphical environment for HP-UX that simplifies many of your daily activities.

**Installing your workstation**

If you have not installed your hardware or started your workstation, please refer to:

- Your workstation's *Installation Guide*.

- Your workstation's *Owner's Guide*.

# Your Workstation Environment

Once your workstation is installed and running, it will display one of two ways to log in.

**Workstations running HP VUE**

If your workstation is running HP VUE, you will see the HP VUE login screen:
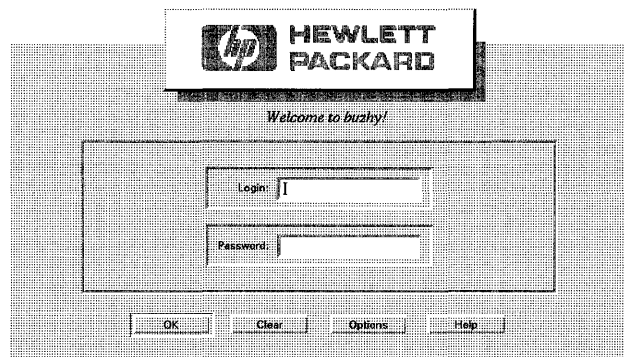


**Workstations not running HP VUE**

If HP VUE is not running, you will see the system console login prompt:

# Choosing an environment

We recommend that you use HP VUE on your workstation whenever possible. The powerful features of HP VUE make it easier to learn to use your workstation and extend its functionality. HP VUE is also very flexible.

**Choosing HP VUE**

You will want to use HP VUE if you:

- Want an easy-to-use interface. You'll still be able to type commands if you choose to do work that way.

- Want to run more than one application at a time. You can run applications that create their own windows and applications that must be run in a terminal.

- Want to run the applications that are part of HP VUE. For example, HP VUE's Text Editor application provides an easy way to edit files. Its Help Manager lets you access extensive online information.

**Choosing a console**

You will want to use a single system console if you:

- Want to run a single application that uses the entire display, such as Starbase Graphics or a CAD application.

- Want to use the X11 Window System without HP VUE.

**If HP VUE does not start automatically**

Under certain conditions, such as after a system update, or after a new workstation is added to a diskless cluster, HP VUE may not be configured to run automatically, and you will see a system console when you boot your system. If HP VUE is not running and you want to turn it on, read Chapter 11 for instructions.

**Turning HP VUE off**

If you need to totally disable HP VUE, see "Stopping HP VUE" in Chapter 11 for instructions.

# Introducing HP VUE

HP VUE is a powerful graphical environment and a set of applications for interacting with your computer.
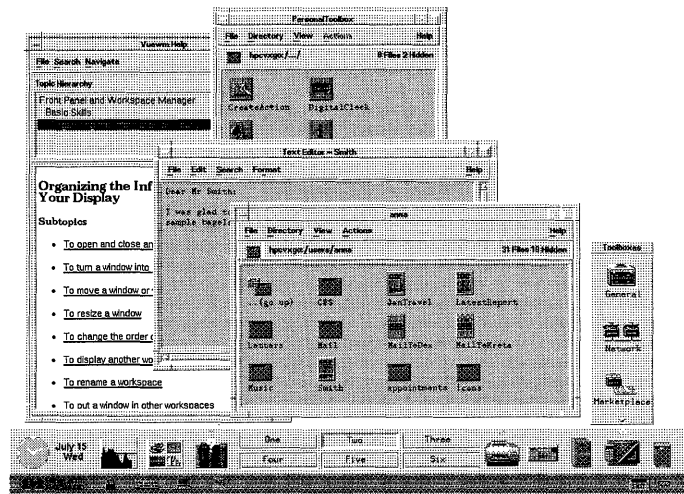
**Features of HP VUE**

HP VUE includes these features:

- Windows and workspaces. Windows are containers on the screen for applications; they let you run more than one application at a time. Workspaces provide a way to make a single display seem like several displays. Each workspace occupies the entire display, and you switch from one workspace to another using a control. It's as though your display had several layers that you can shuffle.

- Icon-based file management. Files are represented by icons that can be selected and moved on the display.

- Front Panel and toolboxes for easy access to applications.

- Extensive online help.

- Session management. HP VUE remembers which applications were running when you logged out and restarts them the next time you log in.

- Easy customization for colors, fonts, window behavior, and other aspects of the appearance and behavior of your workstation.

- Easy-to-use Text Editor and Icon Editor.

- Multi-media application for annotating files.

**There are two forms of HP VUE**

Regular HP VUE sessions provide all the functionality of HP VUE.

Some models and configurations of HP workstations may not give satisfactory performance when running HP VUE. If you wish to exchange some HP VUE features for enhanced performance, you can use HP VUE Lite by selecting it from the Options menu on the HP VUE login screen.

HP VUE Lite is a subset of HP VUE. It features enhanced system performance by omitting full icon-based file management, full session management, and file annotation.

**Entering commands**

While HP VUE provides an easy way to work with files and applications using icons, you will encounter tasks that require you to enter commands manually using a command line. Command lines are provided by HP VUE's Terminal Emulator application. See Chapter 8 to learn how to use a Terminal Emulator. For information about commands, see the *Using HP-UX* manual or the "HP-UX 9.* Operating System Help" in the online HP Help.

**There are additional login options**

You can use the Options menu on the HP VUE login screen to suspend HP VUE in order to run a special program or to perform certain configuration tasks. The Options menu also accesses sessions in other languages.

**Learning about HP VUE**

This manual covers how to use and configure HP VUE.

You can also use HP Help to access online help about HP VUE. To learn more about HP Help, see Chapter 4 in this manual.

# Introducing the console environment

The console provides a command-line interface for running special applications and performing certain system administration functions.

```
$ ls
UTD.cv    OSSD.fc
anna      jodi
dex       jim
ellen     richard
kreta
patti
```

**Learning About System Commands**

The *Using HP-UX* manual explains basic HP-UX commands that you can use in the System Console.

If you use the console to run the X11 Windows System, you can also use HP Help to get information online. The "HP-UX 9.* Operating System Help" contains extensive information about HP-UX commands.

# Summary of Features

| Feature | HP VUE | HP VUE Lite | Single Console |
|---|---|---|---|
| Windows | yes | yes | no |
| Workspaces | yes | yes | no |
| File management using icons | yes | no | no |
| Front Panel containing controls for common tasks | yes | yes | no |
| Toolboxes containing applications | yes | no | no |
| Text editor | HP VUE Text Editor or **vi** | | **vi** |
| Icon editor | yes | yes | no |
| Session management | yes | partial | no |
| Applications for customizing your workstation | yes | yes | no |
| Command line | yes | yes | yes |
| Mailer | yes | yes | yes |

# Information About Your Workstation

**Manuals**

To learn more about your workstation, continue reading this manual or the *Using HP-UX* manual. These other manuals may also be useful:

- If you need help with system hardware installation, see the *Installation Guide* and *Owner's Guide*, for your system.

- If you have not yet installed your HP-UX system, see *Installing and Updating HP-UX* for your version of HP-UX.

- For administration information, see the *System Administration Tasks Manual*.

- For troubleshooting HP-UX, see *Solving HP-UX Problems*.

**Online Help**

If you are using HP VUE or the X Window System, you can use the HP Help System to access information about HP-UX and HP VUE.

The topic "Finding HP-UX information" in "HP-UX 9.* Operating System Help" provides a database of all the documentation published by Hewlett-Packard pertaining to HP-UX workstations.

**Information About Your Workstation**

2

Starting an HP VUE
Session

# Starting an HP VUE Session

An HP VUE session is the time between logging in and logging out.

The login screen, created by the HP VUE Login Manager, is your gateway to HP VUE. It provides a place for you to type your login name and password.

The Options menu on the login screen provides a way to choose what will happen after you log in. In addition to running an HP VUE session, you can choose to run several alternative types of sessions, such as HP VUE Lite, or a failsafe session. You can also select the language for your session.
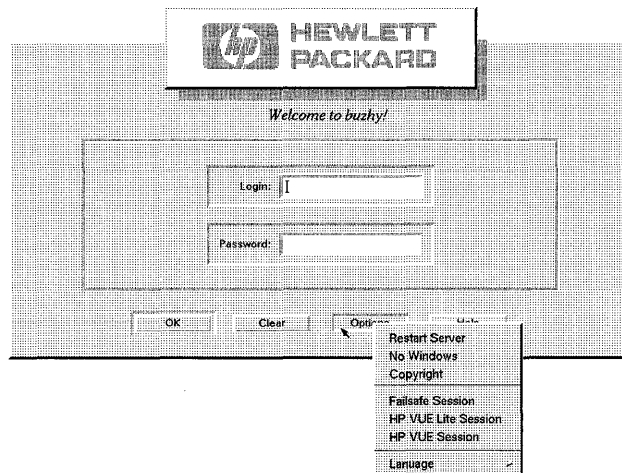


**The Options menu lists your login options.**

# Starting and Ending an HP VUE Session

An HP VUE session starts when you log in. The HP VUE Session Manager takes over after Login Manager recognizes your login and password.

Session Manager provides the ability to "manage" sessions—to remember the state of your most recent session and return you there the next time you log in.

Session Manager saves and restores:

- The appearance and behavior settings—for example, fonts, colors, and mouse settings.

- The window applications that were running—for example, your File Manager and Text Editor windows. Certain types of applications can't be saved and restored by session manager. For example, if you start the vi editor from a comand line in a Terminal Emulator, Session Manager cannot restore your editing session.

- "To run an HP VUE Lite session" covers starting an HP VUE Lite session.

## To log into an HP VUE session

1. Select the Name box and type your login name. Press (Return) or choose OK.

2. Type your password. Press (Return) or choose OK.

If Login Manager does not recognize your name or password, choose Clear and start over.

If your previous session was an HP VUE lite session, then you must select HP VUE Session from the login screen Options menu before logging in.

Once you've logged in, Session Manager starts a session:

- If this is the first time you've logged in, you'll get a new session.

- If you've logged in before, your previous session will be restored.
- If this is the first time you've logged into HP VUE 3.0, but you previously used HP VUE 2.01 on this system, your previous HP VUE 2.01 session will be restored.

**NOTE**

Your system administator may have configured to use only HP VUE Lite sessions. If this the case, logging in will automatically use HP VUE Lite.

# To log out

- Choose the logout control on the Front Panel.
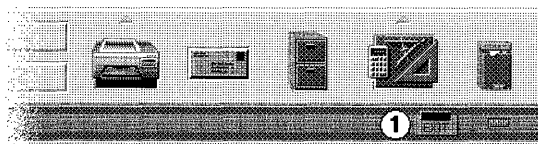


**Use the log out control ① to end the session.**

- *Or,* choose Log out from the workspace menu.

When you log out of a regular HP VUE session, Session Manager saves information about your current session so that it can be restored the next time you log in.

**Cancelling a Session**    Cancelling a sesson ends the session without saving any session information. It also provides a way to log out if, for some reason, the Front Panel and Workspace Menu are not working properly.

To cancel a session, you must stop the X server. In HP-UX systems, use (Shift) (Ctrl) (Reset) ((Shift) (Ctrl) (Pause) for PC-101 keyboards).

# To change your password

1. Display the Personal Toolbox by choosing the Tools control in the Front Panel.

2. Double-click ChangePassword. This displays a window with a prompt for your old password.

3. Supply your current password and press (Return).

4. In response to prompts, enter your new password twice.

# To use a session in a different language

1. Select the language using the menu displayed from Language in the Options menu on the Login Screen

2. Log in.

The default language for your system is set by your system administrator. The Options menu lets you access other languages. Choosing a language in the Options menu sets the LANG environment variable for your session. The default language is restored at the end of the session.

**See Also**    Chapter 27 covers localizing HP VUE sessions.

# Starting Other Types of HP VUE Sessions

In addition to the regular HP VUE session, HP VUE provides these additional types of sessions:

- HP VUE Lite is a scaled-down version of HP VUE. It is useful for limited-memory system, or in situations where you do not need all the functionality of regular HP VUE.

- A failsafe session provides a Terminal Emulator and Window Manager. It is useful for performing configuration tasks before logging into another HP VUE session.

- No Windows mode lets you temporarily leave HP VUE to work in your system console.

## To run an HP VUE Lite session

**To log in**

1. Select HP VUE Lite from the Options menu. (This is unnecessary if your previous session was an HP VUE lite session.)

2. Log in.

Login Manager remembers whether your previous session was a regular session or an HP VUE lite session.

> **NOTE**
>
> Your system may be configured to automatically run an HP VUE lite session. If this is the case, the HP VUE session option in the Options menu is inactive.
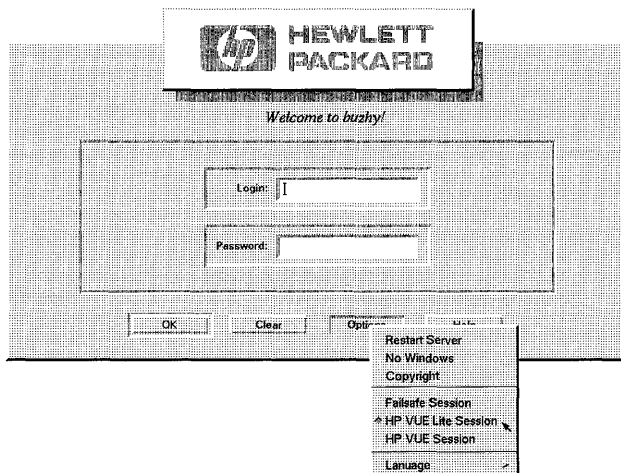
**Use the Options menu to select HP VUE Lite.**

**To log out**
- Choose the logout button on the Front Panel.
- *Or,* choose Log out from the Workspace Menu.
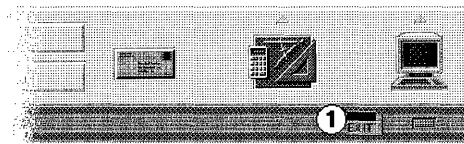


**Use the log out control ① to end the HP VUE Lite session.**

**See Also**
- "What's In the Front Panel (HP VUE Lite Session)" in Chapter 3 describes the Front Panel for HP VUE Lite.
- "Using Tools in HP VUE Lite" in Chapter 6 describes how to run applications in HP VUE Lite.

# To log into and out of a fail-safe session

**To log in**

1. Select Fail-safe from the Options menu.

2. Log in.

**To log out**

• Execute the **exit** command in the Terminal Emulator.

A fail-safe session is a simple session that starts the Workspace Manager and a single terminal window. It is useful when you need access to a single Terminal Emulator window to execute several commands before logging into an HP VUE session.

**See Also**

"To customize the fail-safe session" in Chapter 15.

# To suspend HP VUE using No Windows mode

> **NOTE**
>
> Certain types of configurations (for example, X-terminals) do not provide a No Windows mode option.

**To enter No Windows mode**

1. Choose No Windows from the Options menu. The login screen disappears and is replaced by a console prompt.

2. Supply your login and password as prompted.

**To leave No Window mode**

- Execute `exit` from a prompt.

No Windows mode is not an HP VUE session. When your system is in No Windows mode, HP VUE is suspended. You log in using your operating system mechanism, rather than Login Manager, and there are no windows because the X server is not running.

# Using a Home Session

Ordinarily, HP VUE saves session information when you log out and uses that information to start your next session. If you start or stop applications during your your session, or use Style Manager to change the appearance and behavior of your system, changes you make are reflected in your next session. This type of session is called a **current** session.

HP VUE also provides a **home** session. A home session is a session that you explicitly save. It's like taking a snapshot of your current session at some point in time. Once you've saved a home session, you can specify that logging in always restores that session instead of the current session.

**See Also**

"How Session Manager Works" in Chapter 15 describes how Session Manager manages the current and home session.

## To save a home session

- Choose the Style Manager button in the Front Panel.
- Choose the Startup button in Style Manager to display the Startup dialog.
- Choose Set Home Session in the Startup dialog.
- Choose OK in the confirmation dialog.
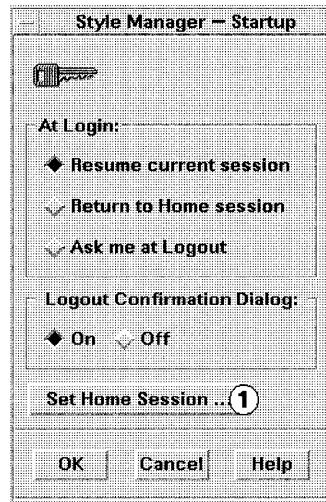
Choose Set Home Session ① to save the current state of your session.

## To automatically start the home session at login

1. Choose the Style Mananger button in the Front Panel.

2. Choose the Startup button in Style Manager to display the Startup dialog.

3. Select Return to Home session.

4. Choose OK.

When you choose Return to Home session, Session Manager will *not* save your session at logout.

# To choose between the current and home session for each session

1. Choose the Style Mananger button in the Front Panel.
2. Choose the Startup button in Style Manager to display the Startup dialog.
3. Select Ask Me at Logout.
4. Choose OK.

You decide what will be your next session—current or home—at logout time. If you choose the return to the home session, the current session information is not saved. (You cannot change your mind the next time you log in.)

# Using the Display Lock and Timeout

The display lock lets you prevent other people from using your workstation when you are away from it without logging out. You can set the appearance of the locked display.

System timeout automatically turns off the display after a certain amount of time elapses in which you haven't used the keyboard or mouse. You can set:

- The amount of time until timeout.

- Whether or not to lock the display at timeout.

## To lock and unlock the display

**To lock the display**
- Choose the lock control in the Front Panel.
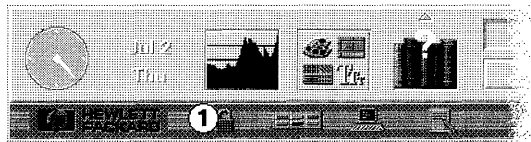


**Choose the lock control ① to lock your display.**

**To unlock the display**
- Type your password.

**See Also**
"To configure who can unlock the display" in Chapter 22 explains how to configure the lock to let other people unlock your display.

# To set the appearance of the locked display

1. Choose the Style Manager control in the Front Panel.
2. Choose the Screen button.
3. Select Full Screen Cover: On or Off.
4. Choose OK.
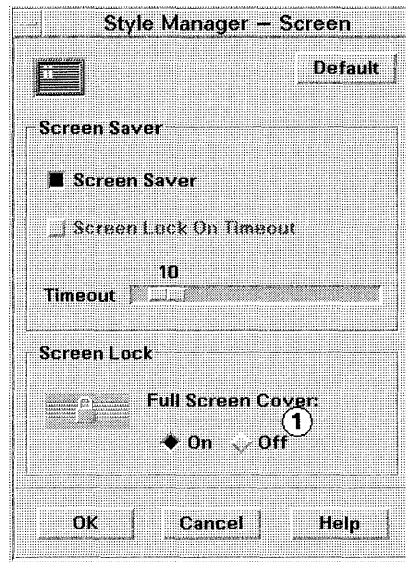


**Set the type of lock cover using buttons ① in the Screen dialog box.**

Ordinarily, the screen is fully covered when the display is locked. You can configure your system to partially cover the locked display.

# To configure system timeout

**To set timeout time**

1. Choose the Style Manager button in the Front Panel.
2. Choose the Style Manager Screen button.
3. Use the slider control to decrease or increase the number of minutes.
4. Choose OK.

**To lock at timeout**

Your system must be running the R5 X server to use this feature.

1. Choose the Style Manager button in the Front Panel.
2. Choose the Style Manager Screen button.
3. Select Screen Lock on Timeout.
4. Choose OK.

# 3

## Basic Skills

# Basic Skills

HP VUE is a graphical user interface that provides workspaces, windows, menus, controls, and a Front Panel to help you organize and manage your software applications.

- A **workspace** is the screen area where you bring the applications needed for your work, arrange them to suit your preferences, and put them away when you're done. HP VUE initially comes with six workspaces (four in an HP VUE Lite session), each giving you a surface on which to put your applications and tools.

- A **window** contains a software application and frames it with controls so you can move it, make it larger or smaller, or put it in additional workspaces.

- The **Front Panel** is a window that contains a collection of frequently-used controls and services all your workspaces.

- **Menus** and **controls** help you manage and operate the software application.

# Organizing the Information on Your Display

You can organize application windows by choosing which applications belong in each workspace. For example, a workspace could contain applications used for correspondence, such as a mailer and Text Editor. Or, you could choose to set up your workspaces according to projects.

A window frames the application with controls that:

- Focus the workstation's attention on the application window. When a window becomes "active," its frame changes color and the application can receive information from you via the keyboard.

- Move the window to a convenient location on the screen or to another workspace.

- Make the window bigger or smaller, or turn it into an icon.

- Remove the window from the workspace.
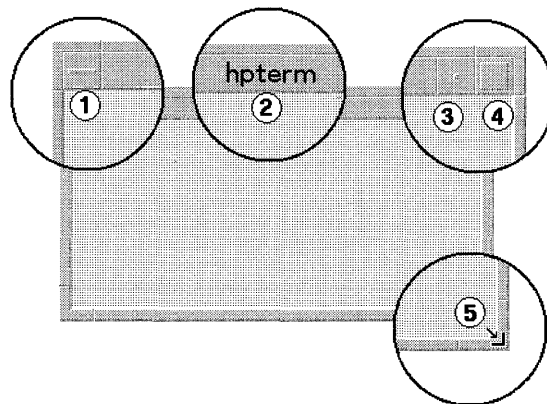


**The window frame lets you perform common window tasks.**

① Window menu button.
② Title bar.
③ Minimize button.
④ Maximize button.
⑤ Resize border.

The application decides which parts of the window frame it needs. For example, Text Editor uses all five, while Front Panel uses none.

# To open and close an application window

**Opening an application window**

- Choose its button in the Front Panel.
- *Or*, choose its icon in a toolbox.
- *Or*, execute the command to start the application.

**Closing an application**

- With the mouse, choose the application's Exit or Close command (usually in its File menu). *Or*, double-click the window menu button.
- With the keyboard, press (Alt)+(F4).

Closing an application removes it from all workspaces.

**CAUTION**

Before closing an application, save your work.

# To turn a window into an icon

As you work, your screen can become cluttered with windows. Changing a few of those windows into **window icons** tidies up the workspace. Programs running in window icons continue to run.

**Mouse**

- Click the window's Minimize button.

**Keyboard**

1. Press (Alt)+space bar to display the window menu.

2. Choose the command:

   - Press the down arrow until you reach the command, then press (Enter).
   - *Or*, press (N), the mnemonic shortcut.

**Restoring a window**

To turn the window icon back into a window:

- Double-click the icon
- *Or*, choose Restore from the window icon menu.

# To move a window or window icon

**Mouse**

1. Position the pointer:

   - For a window, move the pointer over its title bar.
   - For a window icon, move the pointer over the icon.

2. Hold down mouse button 1 as you drag it to its new location.
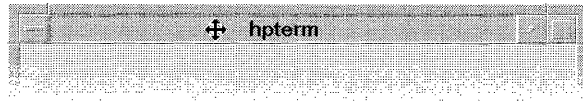


**Move a window by dragging it by its title bar.**

**Keyboard**

1. Press (Alt)+space bar to display the window menu.

2. Press (M), the mnemonic shortcut for the Move command.

3. Use the arrow keys to relocate it. (Using (CTRL) plus an arrow key moves it to its new location faster.)

4. Press (Enter).

To cancel the move operation, press (Esc).

# To resize a window

**Mouse**

1. Place the pointer on a window frame's side or corner.
2. Hold down mouse button 1 as you drag the window outline to the new size.
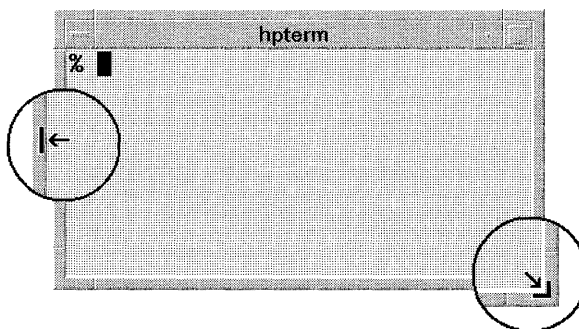


**Stretch or shrink a window by dragging its border to a new size.**

**Keyboard**

1. Press (Alt)+Space Bar to display the window menu.
2. Press (S), the mnemonic shortcut for the Size command.
3. Use the arrow keys to stretch or shrink the window. (Using (CTRL) plus an arrow key stretches or shrinks the window faster.)
4. Press (Enter).

To cancel the resize operation, press (Esc).

# To change the order of overlapping windows

Windows in a workspace can overlap, just like pieces of paper on a desk. And, like papers, you can change the stacking order, bringing the one needing your attention to the top.

**Mouse**

- To bring a window to the top, click a visible part of the window's frame.
- To bring a concealed window to the top, choose Shuffle Up from the Workspace menu.

**Keyboard**

- To cycle through the windows, press (Alt)+(Tab).
- To bring the bottom window to the top, press (Alt)+(ESC).
- To put the top window on the bottom, press (Alt)+(Shift)+(ESC).

## To display another workspace

**Mouse**

- Click the workspace's button in the Front Panel.
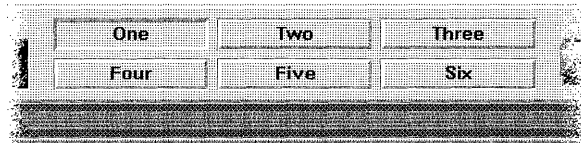


**Display a workspace by choosing its button in the Front Panel.**

**Keyboard**

1. Press (Alt)+(Tab) until you reach the Front Panel.
2. Tab to the top row.
3. Use the arrow keys until you reach the button for the workspace you want to display.
4. Press (Enter).

# To rename a workspace
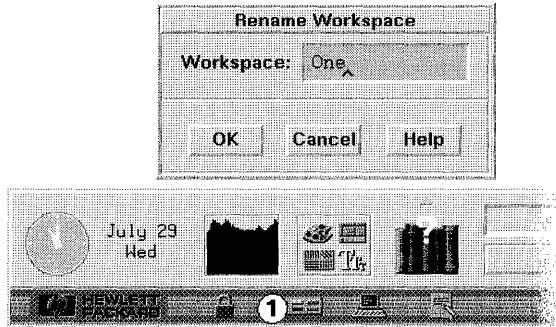


**Use the Rename Workspace button ① to rename the current workspace.**

**Mouse**

1. Display the workspace you want to rename.

2. Choose the Rename Workspace button in the Front Panel to open the Rename Workspace dialog.

3. Edit the Workspace field.

4. Choose OK.

**Keyboard**

1. Display the workspace you want to rename.

2. Use (Tab) and the arrow keys until you reach the Rename Workspace button.

3. Press (Enter) to open the Rename Workspace dialog.

4. Tab to the Workspace field and edit its contents.

5. Press (Enter).

The workspace name can be more than one word.

**See Also**

- "To change the number of workspaces" in Chapter 17 explains how to add or remove workspaces.

# To put a window in other workspaces

A window can occupy one or more workspaces. The workspaces in which a window currently resides are highlighted in the Occupy Workspace dialog.

To put a window in all workspaces, choose Occupy All from the window menu.

**Mouse**

1. Choose Occupy Workspace from the window menu to open a dialog.

2. Click to select or unselect a workspace in which the window is to reside. Selected names are highlighted.

3. Choose OK.
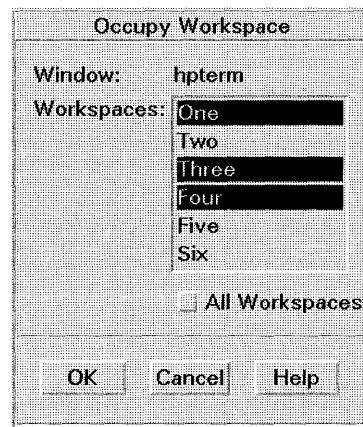


**Select the workspace(s) the window will occupy from the list of workspace names.**

**Keyboard**

1. Press (Alt)+space bar to display the window menu.

2. Press the down arrow key until you reach Occupy Workspace, then press (Enter) to open a dialog.

3. Tab to the Workspaces list.

4. Select the workspace(s) in which the window is to reside:

   • Use the arrow keys to move through the list.

- Press the space bar to select or unselect a workspace. Selected names are highlighted.

5. Tab to OK, and press (Enter).

# Keys used to move around in a workspace

You can use the mouse or the keyboard to move around windows and workspaces. If you use the keyboard, note the following:

- (Alt) is the same as (Extend char) on some keyboards.
- (Enter) is the same as (Return) on some keyboards.

**Within a workspace**

| Key(s) | Moves to ... |
|---|---|
| (Alt)+(Tab) | Next window or window icon. |
| (Shift)+(Alt)+(Tab) | Previous window or window icon. |
| (Alt)+(F6) | Next window belonging to an application or between the Front Panel and a subpanel. |
| (Shift)+(Alt)+(F6) | Previous window belonging to an application or between the Front Panel and a subpanel. |

**Within a window**

| | |
|---|---|
| (Tab) | Next tab group. |
| (Shift)+(Tab) | Previous tab group. |
| (▼) | Next control in a tab group. |
| (▲) | Previous control in a tab group. |

**Within a menu**

| | |
|---|---|
| (▼) | Next menu command. |
| (▲) | Previous menu command. |

# Using the Front Panel

The Front Panel provides a central location for information and applications you use frequently. For information on modifying the Front Panel, see Chapter 17.

HP VUE has two Front Panels— one or regular HP VUE sessions, and another for HP VUE Lite sessions.

Top-row controls with an arrow at the top have subpanels that extend the control's functionality.
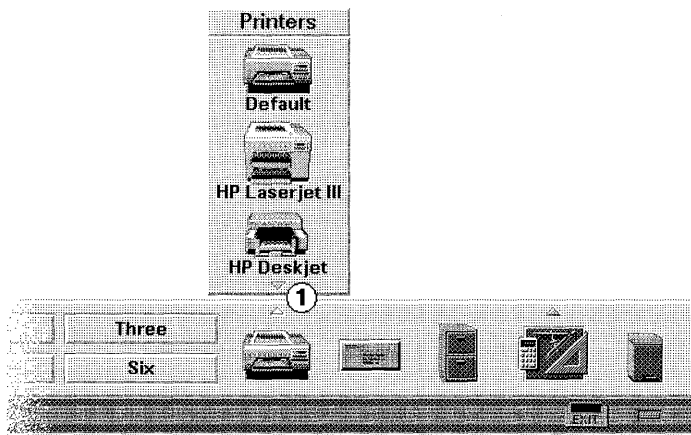


Use the arrow controls ① to display and close the subpanels.

**Getting help on a control**

To get help on a control, choose On Item in the Help subpanel; then, choose the control for which you want help. (Keyboard: give the control focus and press press (F1).)

# What's In the Front Panel (Regular Session)
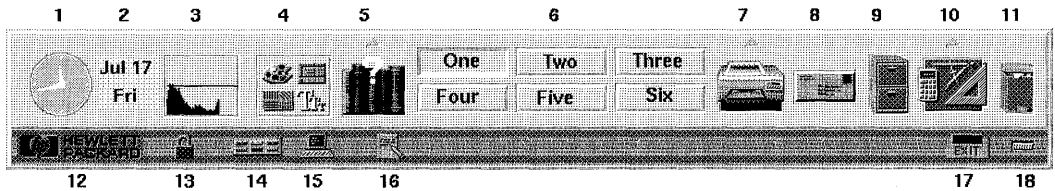


**Top-row controls**

1   Clock displays the current workstation time.

2   Date displays the current workstation date.

3   Load displays workstation activity. This control is actually an application displaying a window in the Front Panel.

4.  **Style Manager** starts Style Manager with which you change display appearance, such as colors, and change system device behavior, such as mouse double-click speed.

5   **Help Manager** starts Help Manager. The Help subpanel provides access to additional online information.

6   **Workspace Switch** displays another workspace.

7   **Printer** displays printer job status on the system default printer. The button is also a drop zone that accepts a file icon. The Printer subpanel can be configured for other printers.

8   **Mailer** starts your electronic mail application. The button is also a drop zone and accepts a file icon.

9   **File Manager** starts a File Manager window showing your home directory.

10  **Toolbox** opens your Personal Toolbox. The Toolbox subpanel opens other Toolboxes that contain actions and utilities.

11  **Trash Can** displays the contents of the trash can. The button is also a drop zone that accepts a file icon.

**Bottom-row controls**    12    Logo gives HP VUE version information.

13    Lock locks your workstation, preventing unauthorized input.

14    **Rename Workspace** displays a dialog in which you can rename a workspace.

15    **Terminal Emulator** starts a terminal emulator window, providing access to a command-line prompt.

16    **Text Editor** starts Text Editor. The button is also a drop zone that accepts a file icon.

17    **Logout** begins the logout process.

18    **Progress Light** blinks to indicate an activity in progress, such as a new window opening.

# What's In the Front Panel (HP VUE Lite Session)
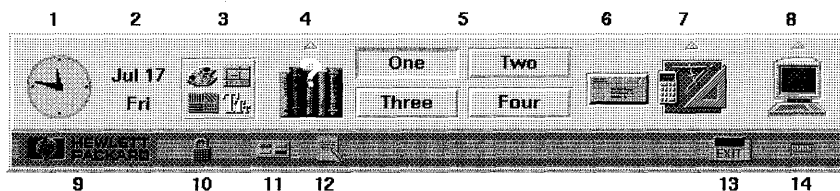


**Top-row controls**    1    Clock displays the current workstation time.

2    Date displays the current workstation date.

3    **Style Manager** starts Style Manager with which you change display appearance, such as colors, and change window behavior.

4    **Help Manager** starts Help Manager. The Help subpanel provides access to additional online information.

5    **Workspace Switch** displays another workspace.

6        **Mailer** starts your electronic mail application. The button is also a drop zone and accepts a file icon.

7        **Toolbox** can be configured to run an application of your choice.

8        **Terminal Emulator** starts a terminal emulator window, providing access to a command-line prompt.

**Bottom-row controls**   9        Logo gives HP VUE version information.

10       **Lock** locks your workstation, preventing unauthorized input.

11       **Rename Workspace** displays a dialog in which you can rename a workspace.

12       **Text Editor** starts Text Editor. The button is also a drop zone that accepts a file icon.

13       Logout begins the logout process.

14       **Progress Light** blinks to indicate an activity in progress, such as a new window opening.

## To choose a Front Panel control

**Mouse**

- Move the pointer over the control and click mouse button 1.

**Keyboard**

1. Press (Alt)+(Tab) until you reach the Front Panel.

2. Move keyboard highlight to the control:
   - (Tab) moves highlight to another row.
   - The arrow keys move highlight to a control within a row.

3. Press (Enter) to choose the control.

# To choose a control from a subpanel

**Mouse**

1. Move the pointer over the control's up arrow and click mouse button 1 to display the subpanel.

2. Move the pointer over the subpanel control and click mouse button 1.

**Keyboard**

1. Move keyboard input focus to the Front Panel control's up arrow.

2. Press (Enter) to display the subpanel.

3. Move keyboard input focus to the subpanel control:

   - (Alt)+(F6) moves focus between a subpanel and the Front Panel.
   - The arrow keys move focus within a subpanel.

4. Press (Enter) to choose the control.
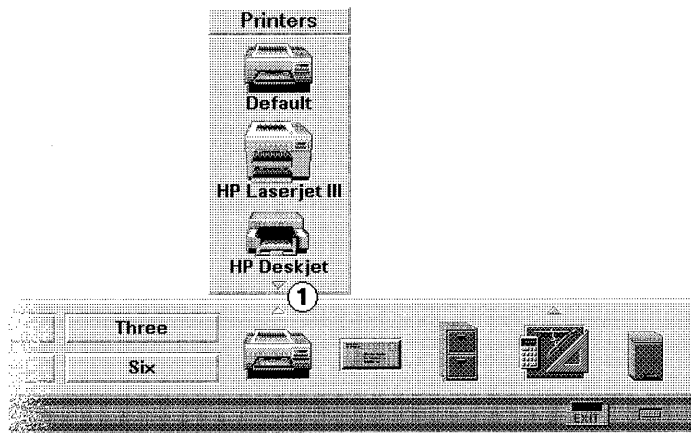


**Use the arrow controls ① to display and close the subpanels.**

**Closing a subpanel**

- Choose the down arrow at the bottom of the subpanel.

**Moving subpanels**

- Move the subpanel as you would move any other window.

# To move the Front Panel

**Mouse**

1. Move the pointer over the Front Panel.
2. Hold down (Alt)+mouse button 1.
3. Drag the Front Panel to its new location.
4. Release (Alt) and mouse button 1.

**Keyboard**

1. Press (Alt)+(Tab) until you reach the Front Panel.
2. Press (Alt)+space bar to display the Front Panel menu.
3. Press (M).
4. Use the arrow keys to relocate the Front Panel.
5. Press (Enter).

# To turn the Front Panel into an icon

**Mouse**

- Choose Minimize/Restore Front Panel from the Workspace menu.

**Keyboard**

1. Press (Alt)+(Tab) until you reach the Front Panel.
2. Press (Alt)+space bar to display the menu.
3. Press (N).

**Restoring the Front Panel**

- Double-click its icon.
- *Or*, choose Minimize/Restore Front Panel from the Workspace menu.
- *Or*, choose Restore from its icon menu.

# Using Controls in Application Windows

To help you manipulate the application's data, windows contain standard sets of controls. You'll come across some of the following standard controls:

- Buttons execute commands, start actions, or specify options and settings. Types of buttons include push buttons, toggle buttons, and radio buttons.

- Text fields provide an area where you can type information.

- Lists display a list of choices from which you can select.

- Sliders provide incremental selection from a range of values.

## To choose a push button

Choosing a push button immediately performs the associated command or action. For example, choosing an OK button applies any changes made to a dialog, then closes it.

**Mouse**

- Move the pointer over the button and click mouse button 1.

**Keyboard**

1. Use (Tab) and the arrow keys until you reach the button.
2. Press (Enter).

## To select a toggle or a radio button

Generally, selecting a toggle button merely specifies the option or setting to be used when a command, such as OK, is performed. You can select more than one toggle button in a group.

Selecting a radio button also specifies an option or setting, but only one radio button in a group can be selected at a time.

**Mouse**

● Move the pointer over the button and click mouse button 1.

To unselect the button, move the pointer over a selected button and click mouse button 1.

**Keyboard**

1. Use ⌷Tab⌷ and the arrow keys until you reach the button.

2. Press the space bar.
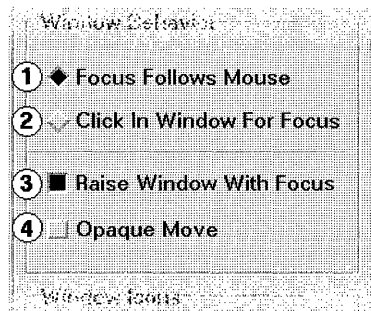


① Selected radio button.
② Unselected radio button.
③ Selected toggle button.
④ Unselected toggle button.

# To scroll the contents of a window

**Mouse**

• Use the scroll bar.
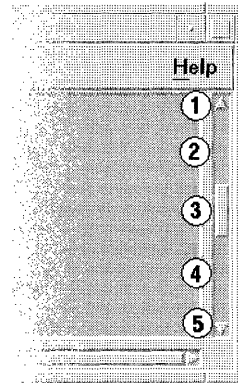


① Click to go up one line.
② Click to go up one page.
③ Drag the slider to scroll incrementally.
④ Click to go down one page.
⑤ Click to go down one line.

**Keyboard**

1. Move to the window area you want to scroll.

2. Use the arrow keys, (Prev) and (Next), or (Pg Up) and (Pg Down).

# To choose a list item

A selected list item highlights and is acted upon when a command, such as OK, is chosen.

**Mouse**

1. Click the list item.

2. Choose a command, such as OK.

In some lists, double-clicking an item selects the item and chooses the default command.

**Keyboard**

1. Tab to the list.

2. Select the list item:

   - In a single-selection list, use the arrow keys to reach your choice.

   - In a multiple-selection list, use the arrow keys to reach your choice, then press the space bar. To unselect the item, press the space bar again.

3. Choose a command, such as OK.
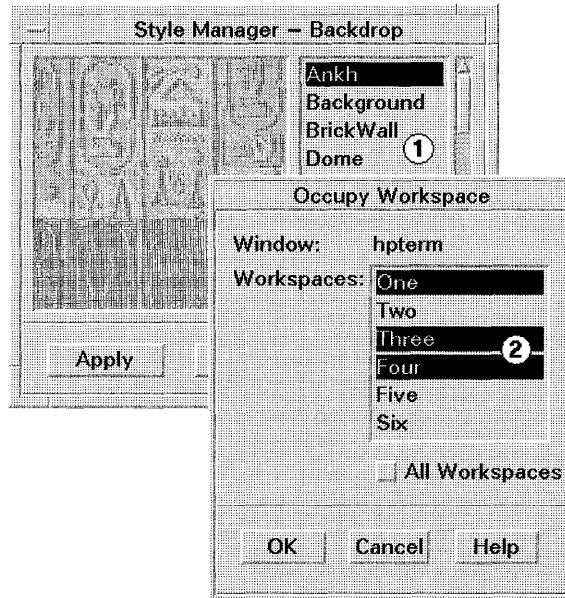


① Single-selection list.
② Multiple-selection list.

# To enter text into an empty field

**Mouse**
- Click the field to get the text insertion cursor.
- Start typing.

**Keyboard**
1. Press (Tab), (Shift)+(Tab), or the arrow keys until you reach the field.
2. Start typing.
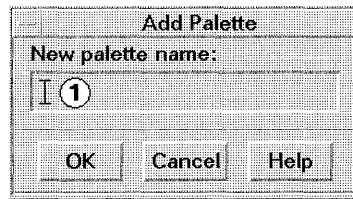


**A text field ① accepts information you type.**

# To edit text in a field

**Mouse**
1. Select the text in the field to edit:
   - **Character(s):** Move the pointer to the first character and drag the pointer to the last character.
   - **Word:** Double-click the word.
   - **Line:** Triple-click the line.
   - **Multi-line field:** Quadruple-click the field.
2. Type the replacement text.

**Keyboard**       1. Press (Tab), (Shift)+(Tab), or the arrow keys until you reach the field.

2. Use the editing keys to edit its contents.

**Editing keys**

| Arrow | Move the text insertion cursor. |
|---|---|
| (Backspace) | Delete the character before the cursor. |
| (Delete char) or (Del) | Delete the character following the cursor. |
| (CTRL)+(Delete char) or (CTRL)+(Del) | Delete all the characters from the cursor to the end of the current line. |

# To cut and paste text into windows

Cutting and pasting eliminates the need to retype text. You can cut from and paste into:

- Text fields.

- Terminal emulator windows.

- Editor windows.

**Cutting text**       1. Move the pointer to the start of the text block.

2. Hold down mouse button 1 and drag the pointer to the end of the text block.

**Pasting text**       1. Put the text insertion cursor at the target location.

2. Click mouse button 2 to paste the text into the field.

To unselect text, click mouse button 1 in an empty area of the window that has the text selected, or press (ESC).

# To use sliders

**Mouse**

- Drag the slider by moving the mouse while holding down mouse button 1.

**Keyboard**

1. Press (Tab) or an arrow key until you reach the slider.
2. Use the arrow keys to move the slider incrementally. Using (CTRL) plus an arrow key moves the slider faster.

# Using Menus

A menu is a list of frequently-used commands that help you manage windows and operate software applications.

Choosing a menu command performs an associated action. For example, choosing Minimize from a window menu turns the window into an icon.

Some of the most common menus are:

- Window menus.

- The Workspace menu.

- Software application menus:

    □ Menus that "pull down" from the application's menu bar.
    □ Menus that "pop up" on an application or on a desktop object.
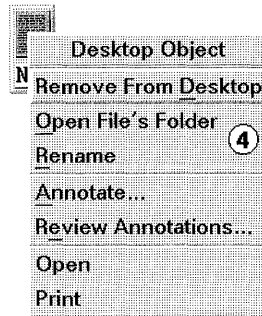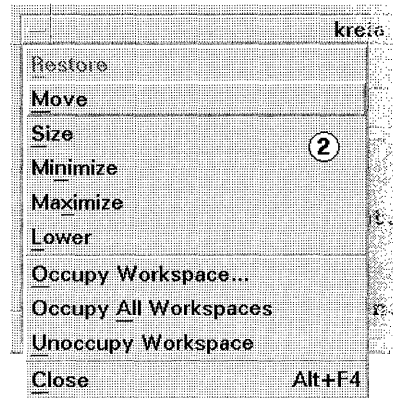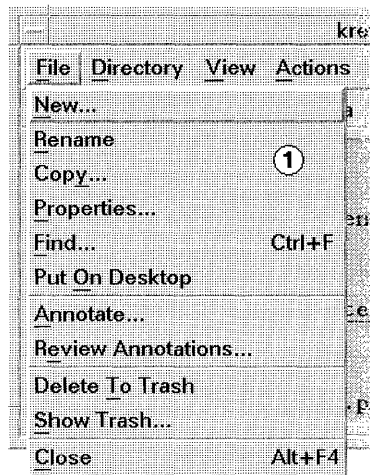
① An application's menu pulled down from a menu bar.
② A window menu.
③ The Workspace menu.
④ A desktop object menu popped-up from the object.

Menu commands that have mottled labels are inactive. These commands are only available under certain conditions. For example, the Restore command is available in a window menu only when the window is an icon.

An underlined character in a command is called a **mnemonic**. A key sequence listed to the right of some commands is called an **accelerator**. Both mnemonics and accelerators provide you with quick keyboard access to menu commands.

# To choose a command from a window menu

**Mouse**

1. Click mouse button 1 on the window menu button to display the menu. *Or*, if the window is an icon, click the icon.

2. Click a menu command.

*Or*, you can use the dragging method:

1. Hold down mouse button 1 as you drag the pointer through the menu.

2. When you reach the command, release the mouse button.

To close the menu without choosing a command, click anywhere off the menu.

**Keyboard**

1. Press (Alt)+(Tab) until you reach the window or window icon.

2. Press (Alt)+space bar to display the window menu.

3. Press the command's *mnemonic*, the underlined character.

To close the menu without choosing a command, press (ESC).

**The window menu**

The window menu provides easy access to services that control the behavior of the window.
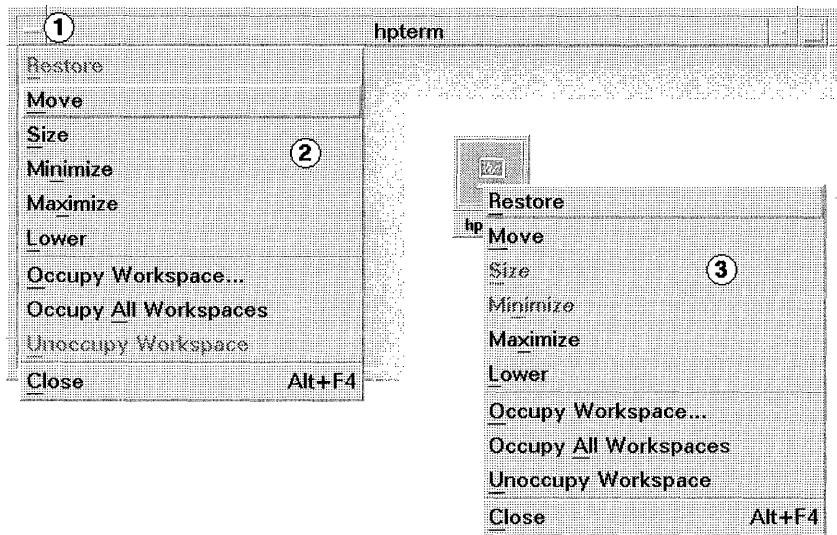
① Window menu button.
② Window menu.
③ Window icon menu.

| | |
|---|---|
| Restore | Turns an icon back into a window. This item is active only when the window is an icon. |
| Size | Interactively changes the size of the window. This item is inactive in the icon's menu. |
| Minimize | Turns the window into an icon. |
| Maximize | Enlarges the window to its greatest allowable size. |
| Lower | Puts the window on the bottom of the stack of overlapping windows in a workspace. |
| Occupy Workspace | Displays a dialog to select the workspaces in which the window should appear. |
| Occupy All Workspaces | Displays the window in all workspaces. |

Unoccupy Workspace          Removes the window from the current
                            workspace. This item is inactive if the window is
                            displayed in only one workspace.

Close                       Closes the window, removing it from the
                            workspace

# To choose a command from the Workspace menu

1. Move the pointer over the workspace backdrop.

2. Hold down mouse button 3 as you drag the pointer through the menu.

3. When you reach your choice, release the mouse button.

The Workspace menu can only be accessed with the mouse.



The Workspace menu contains commands that help manage the workspace.

Shuffle Up                  Puts the bottom window (in a stack of
                            overlapping windows) on the top of the stack.

Shuffle Down                Puts the top window (in a stack of overlapping
                            windows) on the bottom of the stack.

Refresh                     "Repaints" the screen should the display become
                            unreadable.

| Minimize/Restore Front Panel | Turns the Front Panel into an icon. When selected a second time, restores the Front Panel. |
| Restart Workspace Manager | Stops then restarts the **Workspace Manager** after you have customized configuration files. |
| Log out | Begins the logout process, the same as pressing the exit button in the Front Panel. |

# To choose a menu command with the mouse

**Clicking method**

1. Display the menu:

   • For a menu bar menu, click the menu name using mouse button 1.

   • For a popup menu, click inside the application using mouse button 3.

   • For an object's popup menu, click the object with mouse button 3.

2. Click the command.

To close the menu without choosing a command, move the pointer off the menu, and click the mouse button a second time.

**Dragging method**

1. Hold down the mouse button as you drag the pointer through the menu:

   • For a menu bar menu, use mouse button 1.

   • For a popup menu, use mouse button 3.

2. Release the mouse button when you reach the command.

To close the menu without choosing a command, move the pointer off the menu and release the mouse button.

# To choose a menu command with the keyboard

**Menu bar**

1. Press (Alt)+(Tab) until you reach the application window.

2. Press (Alt)+*mnemonic*, the underlined character, to display the menu.

3. Press the command's *mnemonic*.
   *Or*, use the arrow keys to move to the command, and press (Enter).

**Popup menus**

1. Press (Alt)+(Tab) until you reach the application window or desktop object.

2. Press (f4) to display the popup menu.

3. Press the *mnemonic*.
   *Or*, use the arrow keys to move to the command, and press (Enter).

To close the menu without choosing a command, press (ESC).

# 4

# Getting Help

# Getting Help

Hundreds of online help topics are at your fingertips while you're using HP VUE. Here are the ways you can request help:

- Press F1—also known as the "help key"—to get *context-sensitive* help.
- Choose a command from an application's Help menu.
- Open Help Manager to browse all the available help.

**Press F1**
**The "Help" Key**

The quickest and easiest way to get help when you get stuck is to press F1. When you press F1, the application you are using responds by displaying the help topic most closely related to your current activity.

Some computers have a dedicated "Help" on the keyboard. If yours does, it may take the place of the F1 key.

**Use the Help Menu**

Most applications have a Help menu that contains additionl commands for requesting specific kinds of help—such as "Introduction," "Tasks," and "Reference."

To learn more about using help windows, choose Using Help from the Help menu in any HP VUE application. Or, you can press F1 while using a help window.

**Browsing Help with**
**Help Manager**

The HP VUE Help Manager is a special help window that lists all of the online help installed on your system.

To browse the HP VUE help volumes:

1. Choose the Help control in the Front Panel.
2. Choose the "HP Visual User Environment" title to display the list of help volumes available with HP VUE.
3. Browse the list of titles. To open a volume, choose its title.

# Getting Help Within an Application

The HP Help System is built into each of the HP VUE applications (and possibly other applications installed on your system). When you ask for help while using an application, the application displays a help window.

The HP Help System provides two types of help dialogs that applications may use:

- **General help dialogs** have a menu bar, a Topic Hierarchy (that tells you where you are), and a help topic display area. (See ① below.)

- **Quick help dialogs** have just a topic display area and one or more dialog buttons at the bottom. (See ② below.)

---

**①**

**Helpview Help**

File  Search  Navigate                     Help

**Topic Hierarchy**

Welcome to HP VUE

Survival Skills for New Users

**Survival Skills for New Users**

If you are new to HP VUE, here are the essential skills you need to get started:

**To get help within any window:**
Press F1. (Try it now to get help on using help windows.)

**To select and raise a window:** Click the window's frame. Or, press Alt+Tab to select the next window in the stack. Repeat until

**②**

**Help On Help**

**HP Help Command Summary**

- **Backtrack** returns to the previous topic. The Backtrack command (Ctrl+B) is in the Navigate menu and in the pop-up menu in the help display area. (*See also* "Backtracking")

OK          Backtrack          Print

---

## To display context-sensitive help

- Press F1—the "help key."
- *Or*, if you are using a dialog box, choose the dialog's Help button.

When you ask for context-sensitive help within an application, the application determines which help topic is most appropriate for your current situation, or *context*.

## To display other application help

- Choose one of the following commands from the applcation's Help menu:

| | |
|---|---|
| Introduction | Displays the *home topic* for the application. The home topic is the main topic at the top of the application's help topic hierarchy. Hyperlinks in the home topic lead to all other help topics. |
| Tasks | Displays task instructions for most operations performed with the application. |
| Reference | Displays reference summaries for various features, such as windows and dialogs, menus, and resources. |
| On Item | Lets you pick an item in any of the application's windows, then presents a description of the item you've selected. |
| Using Help | Provides help on using the help windows. |
| Version | Displays the version and copyright information for the application. |

Non-HP VUE applications may have different commands in their Help menus.

# Using Hyperlinks

One of the most important skills you need for using help windows is using **hyperlinks**. Any text in a help window that is displayed with an underline is a hyperlink. Choosing a link (with the mouse or the keyboard) immediately jumps to another related topic.

A solid underline ① indicates a standard jump. A dashed underline ② indicates a *definition link*, which displays a popup definition window (actually a quick help dialog).
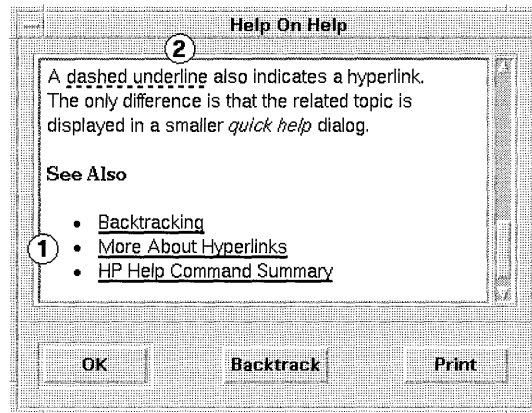


**Some Links Display a New Window** Most hyperlinks display the related topic in the same window. However, if you choose a hyperlink that jumps to a help topic for another application, a new window is displayed. Links that jump from one application's online help into another's, are called *cross-volume* or *cross-application* hyperlinks.

# Knowing Where You Are

Since hyperlinks can quickly take you from one piece of information to another, it is sometimes helpful to know where you are. The Topic Hierarchy list provided in general help dialogs provides this information.
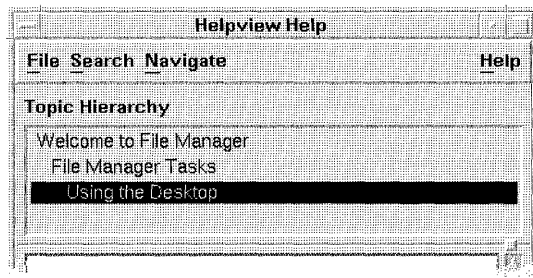


The first item in the list is the *home topic*, or top, of the current help volume. (Usually there's one help volume per application.) Each item below the home topic represents a level of hierarchy down to the last item, which is always the current topic.

As you follow hyperlinks, you'll notice the titles in the Topic Hierarchy list changing, always to reflect your current location.

# Knowing Where You've Been

Each general help dialog keeps track of the 20 most recent topics that you have visited. To display this list, choose History from the Search menu. The History dialog lists the title of each topic (most recent topic on top).
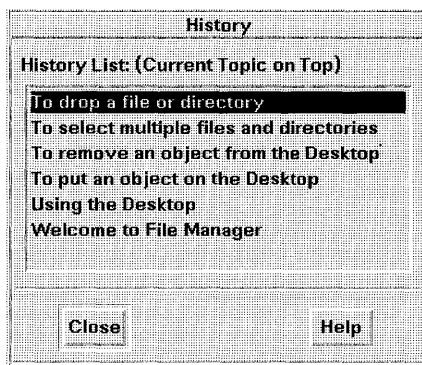
To return to a particular topic, choos its title in the list. When you're done, choose Close.

## To find a topic by exploring

- Choose hyperlinks to jump to related topics.
- Use the Topic Hierarchy list to jump "up" to a higher-level topic.
- Return to the home topic by choosing Home Topic from the Navigate menu.
- Go back to the previous topic by choosing Backtrack from the Navigate menu or from the popup menu in the topic display area.
- Use the Keyword Index to locate topics that cover a certain subject. (Choose Keyword from the Search menu.)

# To find a topic by keyword

1. Choose Keywords from the Search menu to display the Keyword dialog.

2. *Optional:* Type all or part of the keyword you are looking for into the Keyword Filter field. As you type, the keyword list is shortened to show only the keywords that begin with the characters you type in (case is ignored).

3. In the keyword list, select the keyword you are looking for.

4. In the Keyword Result List, select a title of the topic you want to view. The topic you select is displayed in the help dialog.
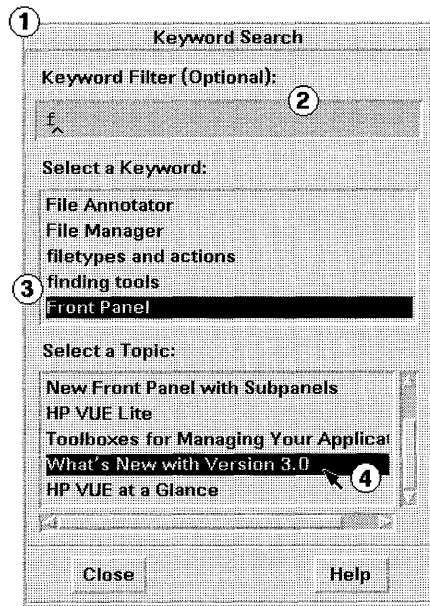
You may view as many topics as you want. When you are done, chose Close.

The scope of the keyword index is limited to the current help volume. That is, you cannot review a global index of keywords for all online help.

# To change the topic display colors

- Set the following resources:

```
*XmDialogShell.XvhHelpDialogWidget*DisplayArea.background:    color
*XmDialogShell*XmDialogShell.XvhHelpDialogWidget*DisplayArea.background: color
*XmDialogShell.XvhHelpDialogWidget*DisplayArea.foreground:    color
*XmDialogShell*XmDialogShell.XvhHelpDialogWidget*DisplayArea.foreground: color
```

Any application started after setting these resources will use the new values for the topic display areas within its help windows.

**Default Values**     The default values for these resources are:

```
*XmDialogShell.XvhHelpDialogWidget*DisplayArea.background:  white
*XmDialogShell*XmDialogShell.XvhHelpDialogWidget*DisplayArea.background:  white
*XmDialogShell.XvhHelpDialogWidget*DisplayArea.foreground:  black
*XmDialogShell*XmDialogShell.XvhHelpDialogWidget*DisplayArea.foreground:  black
```

**See Also**     If you're unsure how to set resources, refer to the following topics for more information:

- "To set resources using the EditResources action" in Chapter 25
- "To create a backup resource file" in Chapter 25
- "To set sys.resources for first session" in Chapter 25
- "To set sys.res.lite for VUE Lite" in Chapter 25

# Browsing Help on Your System

The HP VUE Help Manager lets you browse lists of the online help installed on your system. At the top level, Help Manager lists all of the *product families* that have installed help. If you open a product family, you see a list of help volumes installed for that product. To open a particular help volume, you choose its title.

**Accessing the Previous Version of Help Manager**
Also listed at the top level (at the end of the list) is a special hyperlink for starting the older version of the HP VUE Help Manager (for HP VUE 2.01 or earlier). Access to the previous version of Help Manager is provided so you can access help for applications that have not yet updated their online help for the current 3.0 version of Help Manager.

# To open a help volume using Help Manager

1. Choose the Help control in the Front Panel (the books with a question mark). A general help dialog is displayed, listing all of the product families with online help. This is the top level of Help Manager.
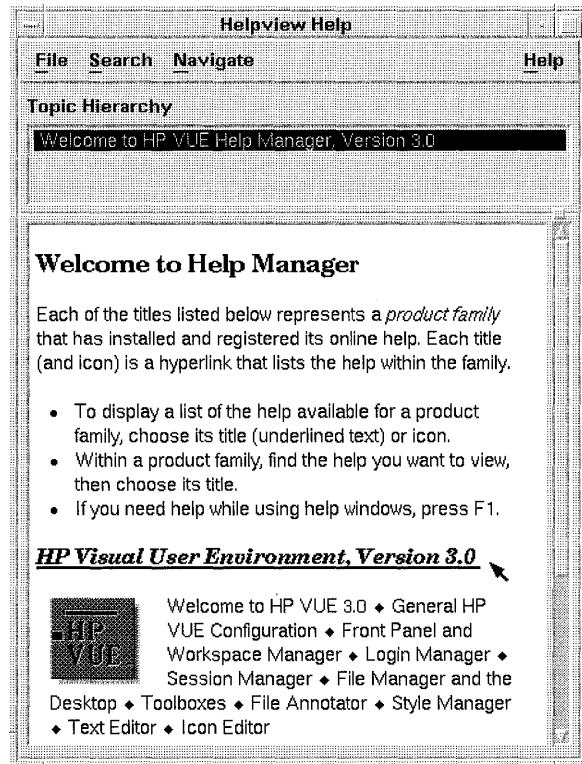


For example, in the figure above, the "HP Visual User Environment, Version 3.0" product family is listed. You can scroll to see additional families. (The order of families in your Help Manager may be different.)

2. Scroll the window to find the product you are interested in. Then choose its title (underlined). This lists the help volumes available for that product.

For example, suppose you want to browse File Manager's online help. After choosing the "HP Visual User Environment, Version 3.0" product family, scroll down until you see "File Manager and the Desktop, (HP VUE 3.0)" . . .
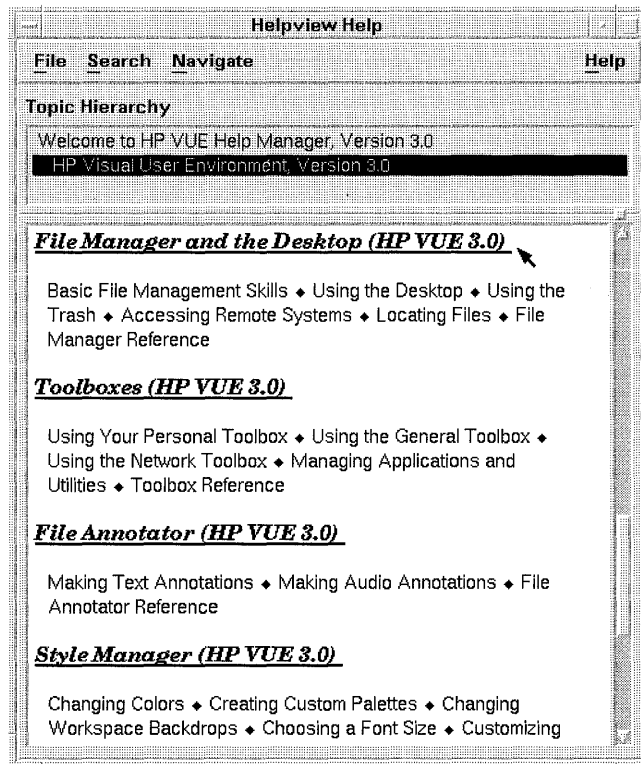
3. To open a particular help volume, choose its title (underlined text). A new window is displayed showing the home topic of the volume you chose.

   For example, if you choose the "File Manager and the Desktop, (HP VUE 3.0)" title, Help Manager displays File Manager's help in a new window . . .
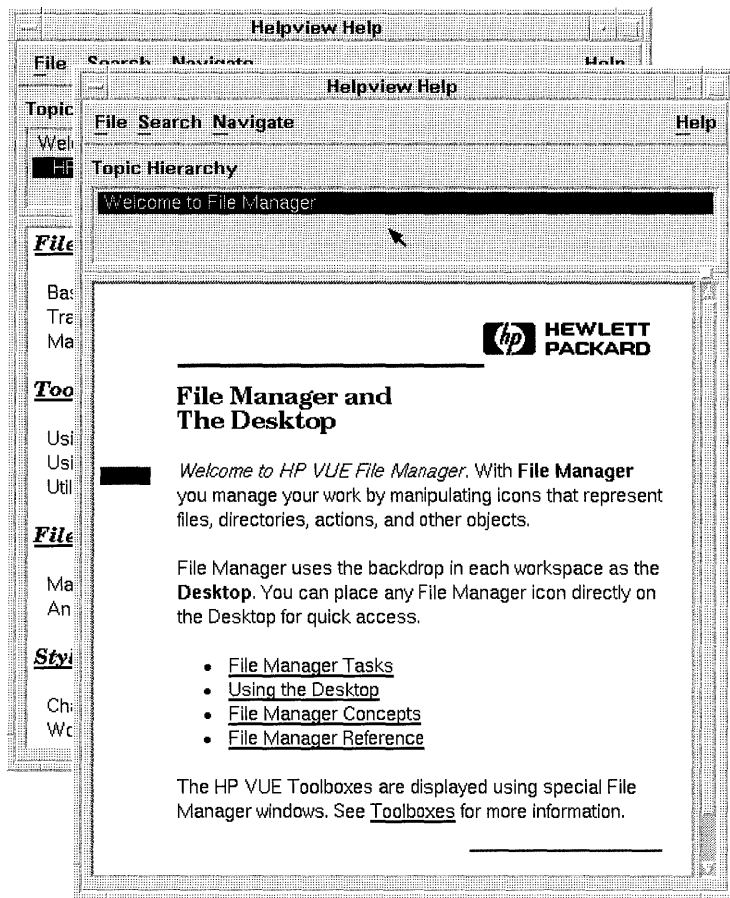


This is the same help topic you would see if you chose Introduction from the Help menu in a File Manager window.

# Displaying a Man Page

Manual reference pages, traditionally known as "man pages," can be displayed using the Man Page control in the Help subpanel.
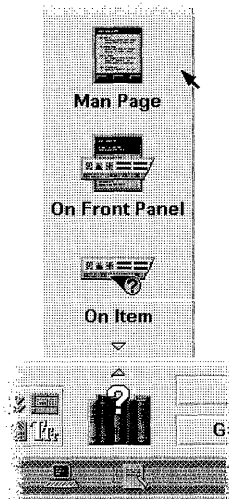


## To display a man page

1. Choose Man Page from the Help subpanel. A dialog is displayed, prompting you to enter a man page.

2. Enter the name of the man page you want to see.

3. Choose Show Man Page.

You may repeat steps 2 and 3 as many times as you want to view additional man pages. When you are done, choose Exit.

**Example**

To display the man page for the **vuewm** command, first choose Man Page from the Help subpanel. A dialog is displayed, prompting you for the man page name. Enter the string "**vuewm**" into the name field ...

```
┌─────────────────────────────────────────┐
│ ─           Man Page           ◄  │║│    │
│                                           │
│  Man Page: │ vuewm│                       │
│                                           │
│                                           │
│   Show Man Page          Exit             │
│                                           │
└─────────────────────────────────────────┘
```
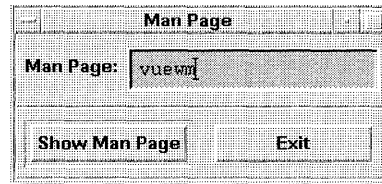
... then choose Show Man Page. The man page is displayed in a quick help dialog.

```
┌─────────────────────────────────────────────────┐
│                      vuewm                 ‡ │ ⌐ │
│ ┌───────────────────────────────────────────┐   │
│ │                                           │   │
│ │ VUEWM(1X)                      VUEWM(1X)  │   │
│ │                                           │   │
│ │                                           │   │
│ │ NAME                                      │   │
│ │     vuewm - The HP VUE Workspace Manager. │   │
│ │                                           │   │
│ │ SYNOPSIS                                  │   │
│ │     vuewm [options]                       │   │
│ │                                           │   │
│ │ DESCRIPTION                               │   │
│ │     Vuewm is an X11 window manager based upon the Motif Window Manager │
│ │     (mwm, version 1.1.3). Vuewm is an integral part of the Hewlett-Packard │
│ │     Visual User Environment (HP VUE). It communicates with and facilitates │
│ │     access to the other components in the environment. │
│ │                                           │   │
│ │     Access to many other VUE components is provided throughout the front │
│ │     panel (also called the workspace manager in HP VUE 2.1). The front │
│ │     panel can be configured to contain controls and indicators for tasks │
│ │     that you perform frequently. These controls include file management, │
│ │     session management, customization, printing, trash, help, or other │
│ │     functions that you define. │
│ │                                           │   │
│ │     Workspace window management represents a significant enhancement of │
│ │     mwm. Workspaces can be thought of as virtual screens.  Multiple │
│ │     workspaces gives you a larger, virtual screen area to place your │
│ └───────────────────────────────────────────┘   │
│   ┌──────┐          ┌──────────┐      ┌───────┐  │
│   │  OK  │          │ Backtrack │     │ Print │  │
│   └──────┘          └──────────┘      └───────┘  │
└─────────────────────────────────────────────────┘
```

When you're done, dismiss the man page by choosing OK. To display another man page, enter its name just as you did for **vuewm**. When you are done viewing man pages, choose Exit.

# Printing Help Topics

Sometimes it is useful to have a printed version of a help topic—or even a whole help volume—for a particular application. The HP Help System lets you print help topics to LaserJet Series II or Series III printers. (The default is Series III.)

When you print a topic, it is formatted just as it is on the display. Color graphics are automatically dithered to black and white.

## To print help topics

1. Choose Print from the File menu to display the Print dialog.

2. Select the topics you want to print:

   - Print All—to print all topics.
   - Print Current Topic—to print only the current topic (the default).
   - Print Current and Down—to print the current topic and its subtopics.

3. Choose OK.

In quick help dialogs, the Print button immediately prints the current topic, without displaying the Print dialog.

By default, printed output from HP Help is directed to your default printer. You can specify another printer by setting the `printer.name` resource. If you are not printing to a LaserJet Series III (or later) printer, you must also set the `printer.`*name*`.model` resource.

## To choose a printer for yourself

1. Copy the **/usr/lib/X11/app-defaults/Helpprint** file into your home directory and make it writable.

2. Edit your **/**_HomeDirectory_**/Helpprint** file, setting the **name** and **model** resources for the printer you want to use:

   **\*printer.name:** _name_
   **\*printer.**_name_**.model:** _model_

   Where _name_ is the device name of the printer and _model_ is either **LJII** or **LJIII**. (The default model is **LJIII**.)

Your system administrator may have already edited the **Helpprint** file to configure the correct models for the printers on your system. If so, you specify just the **printer.name** to select a printer.

**Examples**          Suppose you have a LaserJet II printer named **hp-laser**. To direct all printed help topics to that printer, enter these resources into your **Helpprint** file:

   **\*printer.name:          hp-laser**
   **\*printer.hp-laser.model:   LJII**

_Or_, suppose you want to direct all printed help topics to a LaserJet III printer named **lj3d**. Since the default model is **LJIII**, you need to specify only this single resource:

   **\*printer.name:        lj3d**

If you want to print help two-sided (_duplex_), include this resource, too:

   **\*lpCommand:  lp -oraw -od**

The **-od** option requests duplex printing if your printer supports it.

# To choose a printer for all users

1. Login as superuser.

2. Edit the **/usr/lib/X11/app-defaults/Helpprint** file, adding the necessary **model** resource for each printer as follows:

   **\*printer.**_name_**.model:** _model_

   Where _name_ is the device name of the printer and _model_ is either **LJII** or **LJIII**. The default model is **LJIII**.

3. To specify the default printer, use the printer **name** resource:

   **\*printer.name:** _name_

   Where _name_ is the device name of the printer you are configuring.

> **NOTE**
>
> For non-default languages, be sure to edit the **Helpprint** file in the correct **/usr/lib/X11/app-defaults/**_language_**/** subdirectory.

# Accessing a Help Server

A **help server** is a computer system on a network that other systems can rely on for accessing online help files. You can configure your local system (the "client") to access a remote system (the "server") by mounting the remote system's disks on your system, modifying the help search path, and updating your Help Manager.

## To configure access to a help server

1. On the remote system (the "server"), log in as superuser, then do these tasks:

   □ *Install the online help you want to access.* Usually, online help files are installed when you install application software. It is common for an application server to also be a help server, since the files are usually kept together.
   □ *Export the disk volume where the online help resides.* Usually, this is the **/usr/** volume—if not, also export **/usr/**, since the **/usr/vhelp/** directory should also be available for mounting.

2. On the local system (the "client"), log in as superuser, then do these tasks:

   □ *Mount the disk volume(s) exported from the remote system* (in step 1).
   □ *Edit the* **/usr/vue/config/Xsession***script.* Set the XVHHELPSYSTEMSEARCHPATH environment variable to search the newly mounted directories for online help.
   □ *Log out and back into HP VUE.* This ensures the **Xsession** script is executed, setting the search path variable.
   □ *Update your local Help Manager.* Open a Terminal Emulator window. Verify that the XVHHELPSYSTEMSEARCHPATH is set properly (use the **env** command). Then, execute this command to update the browser volume used by Help Manager to list available help:

      ```
      /usr/vue/bin/helpgen
      ```

**Example**

This example shows how the XVHHELPSYSTEMSEARCHPATH environment variable is used within HP VUE to access help installed on another computer on the network. (These tasks require superuser permission to mount a new file system and edit the **/usr/vue/config/Xsession** script.)

To make the remote files accessible, "mount" the remote system's **/usr/** disk volume on your local system in the directory **/nfs/**_hostname_**/usr/**. (Refer to the operating system documentation or online help to learn how to "mount" a _network file system_.)

Once the remote files are physically available, you must tell the HP Help System where to look for them. You do this in HP VUE by editing the **/usr/vue/config/Xsession** script. Search for the section of the file that defines XVHHELPSYSTEMSEARCHPATH, then uncomment and edit the lines so that they look like this:

```
XVHHELPSYSTEMSEARCHPATH=\
/usr/vhelp/%T/%L/%H:\
/usr/vhelp/%T/%H:\
/usr/vhelp/%T/%L/%H.hv:\
/usr/vhelp/%T/%H.hv:\
/usr/vhelp/%T/C/%H:\
/usr/vhelp/%T/C/%H.hv:\
/nfs/hostname/usr/vhelp/%T/%L/%H:\
/nfs/hostname/usr/vhelp/%T/%H:\
/nfs/hostname/usr/vhelp/%T/%L/%H.hv:\
/nfs/hostname/usr/vhelp/%T/%H.hv:\
/nfs/hostname/usr/vhelp/%T/C/%H:\
/nfs/hostname/usr/vhelp/%T/C/%H.hv
```

Where _hostname_ is the name of the directory you created for the remote system.

When you're done editing, save the file, then log out and back in. To update Help Manager on the local system, run this command:

```
/usr/vue/bin/helpgen
```

Any help installed on the remote system (_hostname_) should now be available in Help Manager. (If you already have a Help Manager window open, close it and open a new one to see the changes.)

# 5

## Managing Files and Using the Desktop

# Managing Files and Using the Desktop

HP VUE provides the following to help organize and maintain the files and directories you use:

- **File Manager** is an application for managing files and directories.

- The **Desktop** allows you to place any file or directory icon directly on the workspace backdrop for easy access.

- **Toolboxes** are special File Manager views for managing applications and other software available to you.

**Introducing File Manager**

The HP VUE File Manager displays files and directories as icons. Using a mouse, you can manipulate the icons directly by dragging and dropping them into place. For example, to print a file, simply drop it on the printer in the Front Panel.

File Manager associates a meaningful icon with each file and directory based on its **filetype**. More importantly, **actions** associated with each filetype define the desired behavior for the file or directory.

For instance, you might have a filetype that identifies document files. The default action—performed by double-clicking the file's icon—would open the file for editing. That is, the action saves you the trouble of remembering and typing the command to start your word processor. The action automatically starts your word processor and loads the file. Additional actions may provide shortcuts to other frequent tasks, such as printing the document.

---

**NOTE**

File Manager is *not* available in HP VUE Lite. If you are using HP VUE Lite, you must use shell commands in Terminal Emulator windows to manage and manipulate files and directories.

---

**The Desktop for fingertip access**

Any file, directory, or action that you access frequently can be placed directly on the Desktop for quick access. The Desktop occupies the entire backdrop in each workspace. In fact, since HP VUE has six workspace (by default), you effectively have six separate Desktops.

When you put an object on the Desktop, the original object is not altered. Each Desktop object is really just a reference to the actual object that remains in its original location.

You can leave objects on the Desktop as long as you want. Their locations are remembered when you log out and restored when you log in again.

**Popup Menus**

To display a popup menu, you point to the object's icon, then press and hold mouse button 3 (the right button on a two-button mouse). The popup menu contains a few standard commands for each object, but it also includes all of the available actions for the object.

When you no longer need an object on the Desktop, choose Remove From Desktop from the object's popup menu.

**See Also**

- Chapter 6 explains how to use the toolboxes.
- Chapter 19 is an introduction to creating filetypes and actions.
- The *Beginner's Guide to HP-UX* manual also includes basic file management skills.

# Introducing the Desktop

The HP VUE Desktop occupies each workspace. Its a handy place to put the files, directories, actions, and applications that you use most frequently.

To place an object on the Desktop, you drag its icon from a File Manager or Toolbox window then drop it on the workspace backdrop. A Desktop icon appears right where you drop the object.

You use Desktop icons exactly like the icons in the File Manager or Toolbox. To execute an object's default action, double-click its icon on the Desktop.

Each Desktop object also has a popup menu containing additional commands and actions for the object. To display the popup menu for a Desktop object using the mouse, point to the icon, then press and hold the right mouse button. To display the menu with the keyboard, use (Alt)+(Tab) until the icon is highlighted, then press (F4).

> **NOTE**
>
> The Desktop is not available in HP VUE Lite.

# The Hierarchical File System

If you are new to computers, the idea of a "hierarchical file system" may be new to you. Here are the basic concepts . . .

**What's a file?**

A file is a named container for information. Most of the files you use contain data of some particular format—a document, a spreadsheet, a chart.

Most applications understand a limited number of file formats. For example, a document editor may not be able to read a spreadsheet file. HP VUE helps you recognize different types of files using a filetype database. A filetype identifies the files of a particular format and associates them with the appropriate applications. These associations mean you don't have to remember commands to accomplish frequent tasks.

On some computers, file names cannot be longer than 14 characters. If you are not sure if your computer can support longer file names, check with your system administrator.

**What's a directory?**

A directory is a container for files, similar to a folder in a file cabinet. Since a directory can contain other directories—sometimes called *subdirectories*—you can create multiple layers of organization that forms a hierarchy.

Within any single directory, each file name must be unique. However, files in different directories may have the same name.

As you navigate from directory to directory, your current location is referred to as the **current working directory**.

**What's a path?**

The location of a file is often specified using the directories and subdirectories that lead to the file—this is called a **path**. A path is an *absolute* path if it begins at the **root directory**. The root directory is the single common directory where the hierarchy begins. If a path begins with a slash (**/**), it is an absolute path specified from the root directory. For example, this is an absolute path to the file **sys.vuewmrc**:

> /usr/vue/config/sys.vuewmrc

A path is *relative* if it describes the location of a file or directory as it relates to the current working directory. If a path does not begin with a slash, it is a

relative path. For example, if the current working directory is **/usr/vue/**, the relative path to the **sys.vuewmrc** file is:

    config/sys.vuewmrc

If your current working directory is different, then the relative path to the same file is different.

Two special directory names are useful specifying relative paths. The **./** directory (sometimes called "dot") represents the current working directory. The **../** directory (sometimes called "dot-dot") represents the *parent* directory—the directory one level up in the directory hierarchy. For example, if you current working directory is **/usr/vue/config/panels/**, then the relative path to the **sys.vuewmrc** file is:

    ../sys.vuewmrc

That's because the file is in the **/usr/vue/config/** directory, one level above the current directory.

**See Also**

- "To change to another directory"
- "To specify a remote directory or file"
- "To find a file by name"
- "To find a file by contents"

If you still want to learn more about your computer's file system, refer to the online help or documentation for your operating system. There are also many commercial books available that cover the basics of file systems and file management.

# File Ownership and Security

Three groups of users can access files: *owner*, *group*, and *other*. File access is divided into three functions: *read* permission, *write* permission, and *execute* permission.

**Who Has Access?**    The three basic classes of users are:

- Owner—Usually the person who created the file.

- Group—Several users that have been grouped together by the system administrator. For example, the members of a department might belong to the same *group*.

- Other—All other users on the system.

**What Kind of Access?**    The access permissions on a files specify how that file can be accessed by the owner, group, and other.

- Read Permission—Allows access to retrieve or view the contents of the file or directory.

- Write Permission—Allows access to change the contents of the file or directory.

- Execute Permission—For a file, allows access to *run* the file (for executable files, scripts, and actions). For a directory, allows access to run commands, scripts, and actions within that directory.

With File Manager, you can view and change the access permissions for any file or directory. See "To change the owner of a file or directory" and "To change the access permissions on a file or directory".

**Examples**

To make a directory private:

- Change the directory's properties, giving yourself (the "owner") read, write, and execute permission, but give no permissions for "group" and "other." This means that only you and the superuser can view the contents of the directory.

To make an action that you've created available for everyone to use, but protected so it isn't inadvertently overwritten:

- Change the file's properties, giving read and execute permission to "owner," "group," and "other." Don't give anyone write permission.

**Default Permissions**

The default permissions used when you create a new file or directory may be altered by your system administrator. To determine what your current defaults are, create a new file or directory, then open the Properties dialog for that file or directory.

Check with your system administrator regarding how to change your default permissions.

# Basic File Management Skills

To get started using File Manager, you need to learn a few basic skills:

- *Selecting*—Many commands operate on the "selected" file or directory.

- *Dragging and Dropping*—If you use a mouse, the easiest way to perform many operations is to manipulate the icons with the mouse.

- *Using Popup Menus*—Each icon displayed by File Manager has its own popup menu, including objects on the Desktop. Each popup menu provides shortcuts to many common operations.

## To select a single file or directory

**Mouse**

- Click the object's icon.

**Keyboard**

1. Use the (Tab) and direction keys to move the highlight to the object you want to select.

2. Press Enter or Spacebar.
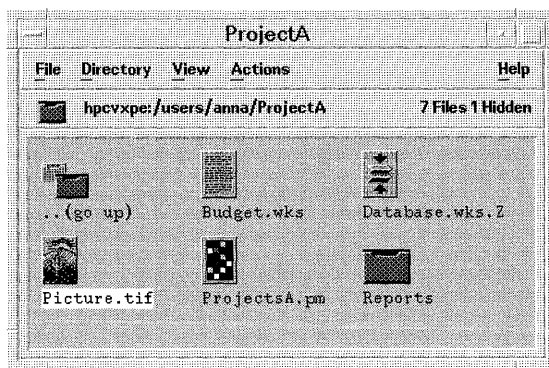
**When an object is selected, its name and icon are highlighted.**

When you select an object, its name is highlighted. Many commands in File Manager's menus apply to the *selected* object. In particular, the Actions menu always contains the actions that can be performed on the selected file or directory. If no icons are selected, the entire Actions menu (and some commands in other menus) is inactive.

To select multiple objects, hold down (Ctrl) while selecting.

To deselect an object, select another one or click an empty area within the File Manager window.

# To select multiple files and directories

**Mouse**
- Drag a box around the objects you want to select. (Press mouse button 1 in a blank area of the view, drag to draw a box, then release to select the icons included in the box.)

- *Or,* click to select the first object, then (Ctrl)+click to select additional objects.

**Keyboard**

1. Select the first object.

2. For each additional object you want to select, move the highlight to its icon, then press $\boxed{\text{Ctrl}}$+$\boxed{\text{Enter}}$.

When multiple icons are selected, dragging any one of the selected icons drags the whole group.

Also, when multiple icons are selected, the Actions menu and some commands in other menus are inactive.

**Reasons for Selecting Multiple Objects**

- Deleting several files at once.

- Moving a group of files to a new directory.

- Putting several objects on the Desktop.

# To rename a file or directory

**Mouse**

- Select the object's name, type the new name, then press Enter.

**Keyboard or Mouse**

- Choose Rename from the File menu, type the new name, then press Enter.

To cancel a rename operation, press $\boxed{\text{Esc}}$.

# To get help on a file or directory

**Mouse**

- Select the object, then press F1.

- *Or,* choose On Item from the Help menu, then click the object's icon.

**Keyboard**

- Move the keyboard highlight to the object's icon, then press (F1).

The description displayed when you ask for help on an icon describes the filetype associated with the file or directory. The description is specified in the filetype definition.

**See Also**

- Chapter 4 explains how to use HP Help.
- Chapter 19 is an introduction to creating filetypes and actions.

## To open a file or directory

- Double-click the object's icon.

- *Or,* select the object's icon, then choose a command from the Actions menu that opens the file or directory. (The Actions menu contains different commands depending on the filetype of the selected object.)

- *Or,* choose a command from the object's popup menu.

Double-clicking an icon executes the object's *default action*, which is always the first command in the Actions menu. For most data files, the default action opens the file by starting the appropriate application and loading the file.

Directories have actions defined to open them (either in place, using the current File Manager window, or in a new window).

## To drop a file or directory

1. Point to the object's icon.

2. Press *and hold* mouse button 2, known as the "drag" button. (On a two-button mouse, press both buttons simultaneously.)

3. Drag the icon to the location where you want to drop it, then release the mouse button.

So, the motion for dropping an object is *press ... drag ... release.*
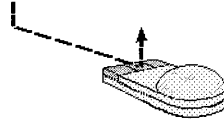
To cancel a drag in progress, press (Esc) before releasing the mouse button.

If more than one icon is selected, you drag the entire group by dragging any of the selected icons.

---

**NOTE**

You cannot drag and drop an object without a mouse or other pointing device.

---

**What Dropping a File Means**

- If you drop an object into an open File Manager window or into a directory icon, the file is moved to that directory.

- If you drop an object onto the workspace backdrop, it's placed on the Desktop in the current workspace. (Actually, a Desktop object is really just a *reference* to the real object, which remains in the File Manager view, unchanged.)

- If you drop a file on the Printer control in the Front Panel (or one of the printers in the Printers subpanel), the file is printed.

- If you drop a file on the Personal Toolbox control in the Front Panel, the object is copied into your Personal Toolbox.

- If you drop a file on the Trash control in the Front Panel, the object is moved into the Trash Can.

- If you drop a file on an action icon, the action is invoked with the file as a parameter.

If you attempt to drop the icon in a location that does not support dropped objects, the icon snaps back to its original location in File Manager or on the Desktop.

## To display a popup menu

**Mouse**

1. Point to the icon whose popup menu you want to display.

2. Press mouse button 3—the "popup" button. (On a two-button mouse, press the right mouse button.)

3. To choose a command from the menu, drag to the command, then release. Or, click the command.

**Keyboard**

1. Using the (Tab) and direction keys, move the highlight to the icon whose menu you want to display.

2. Press (F4).

3. To choose a command from the menu, use the direction keys to highlight the command, then press Enter.

To cancel the menu without choosing a command, press (Esc).

# Manipulating Files and Directories

The major identifying features of a file or directory are:

- Its name and location in the file hierarchy.

- Its owner permissions (properties).

- Whether or not it is linked to another file. A linked file is a representation of another file. Linked files provide a way for you to have one file that appears to exist in two or more directories.

## To create a new file or directory

1. Choose New ...

   - From the File menu to create a new file.
   - *Or,* from the Directory menu to create a new directory.

   A dialog box prompts you for the name of the new file or directory.

2. Enter the new name.

3. Choose OK.

**Other Ways to Create Objects**

- Copying an existing object, then rename it.

- Create a new file within an application. For example, if you use Text Editor to write a new text file, when you save it for the first time, a new file is created.

# To move a file or directory

**Mouse**

1. Be sure the destination directory is visible (either as an icon or as an open File Manager view).

2. Drop the icon into its new directory.

**Keyboard**

- Copy the file, then delete the original file.

**Notes About Moving**

- If you move a directory, the entire directory and its contents are moved.
- If you attempt to move a file or subdirectory from a directory where you do not have permission to make changes, File Manager may create a copy of the object, and display a "File Manipulation Error" indicating that you cannot delete the original file.

**For HP VUE Lite**

Since File Manager is not available in HP VUE Lite, you must use the **mv** command (move), to move and rename files. Open a Terminal Window then execute the **mv** command like this:

    **mv** *oldfile newfile*

Or, to move one or more files into an existing directory:

    **mv** *files directory*

# To copy a file

**Mouse**

1. Be sure the destination directory is visible (either as an icon or as an open File Manager view).

2. Press *and hold* (Ctrl) while dragging the file's icon, then drop the icon into the directory where you want the copy.

**Keyboard**

1. Select the file's icon.

2. Choose Copy from the File menu. A dialog displays, prompting you for the name of the new copy.

3. Enter a name for the new copy of the file.

4. Choose OK.

> **NOTE**
>
> The Copy command cannot be used to copy directories.

**For HP VUE Lite**

Since File Manager is not available in HP VUE Lite, you must use the **cp** command (copy), to copy files and directories. To copy a file, open a Terminal Emulator then execute the **cp** command like this:

    **cp** *oldfile newfile*

Or, to copy a directory and its entire contents:

    **cp -r** *directory newdirectory*

File and directory names may include a path, if needed.

**See Also**

- "To re-sort files and directories (clean up)"

## To create a symbolic link

**Mouse**

- Press *and hold* (Shift) while dragging the object's icon. When you drop the icon into a new directory, a symbolic link is created in that directory that points to the original object.

**Keyboard**

1. Choose Terminal from the Directory menu. This opens a Terminal Emulator window with the same current working directory.

2. Use the `ln` command to create the symbolic link:

   `ln -s` *filename linkname*

   Where *filename* is a complete path to the file, if it isn't in the current directory, and *linkname* is the name of the symbolic link, which may also be a complete path.

3. Close the Terminal Emulator window by choosing Close from the window menu or by executing the `exit` command.

**For HP VUE Lite**

Since File Manager is not available in HP VUE Lite, you must use the `ln` command (link) to create a symbolic link. Open a Terminal Emulator, the execute the `ln` command as shown above under "Keyboard."

## To change the owner of a file or directory

1. Select the object's icon.

2. Choose Properties from the File menu or from the object's pop-up menu. File Manager displays the File Properties dialog.

3. Make the desired changes:

   - To change the file's owner, type the new owner's name into the Owner Name field.
   - To change the file's group, type the new group name into the Group Name field.

4. Choose OK.

---

**Notes**

- If you give ownership of the file to another user, you may not be able to change the permissions again unless that user returns ownership to you.
- If you do not have permission to change the properties, some of the controls in the File Properties dialog are inactive.

---

# To change the access permissions on a file or directory

1. Select the object's icon.

2. Choose Properties from the File menu. File Manager displays the File Properties dialog.

3. In the Permissions box, select the permissions for the file or directory's owner, group, and other:

   - Select Read for the users that should be allowed to read the file or directory.
   - Select Write for the users that should be allowed to alter the file or directory.
   - Select Execute for the users that should be allowed to execute the file.

4. Choose OK.

The object's owner is identified in the Owner Name field. The object's group is identified in the Group Name field.

Read, write, and execute permission is selected in the Permissions box. The permissions selected in the Group row indicate the access privileges for any user belonging to the named group. The permissions selected for the Other row apply to all other users.

**Notes**

- If you give ownership of the file to another user, you may not be able to change the permissions again unless that user returns ownership to you.
- If you do not have permission to change the properties, some of the controls in the File Properties dialog are inactive.

# Navigating to Directories and Subdirectories

File Manager is like a vehicle that you drive to tour the filesystem on your computer and other computers on the network. Each File Manager window—also called a "view"—shows the contents of a single directory.

If you are not familiar with hierarchical file systems, see "The Hierarchical File System".

## To change to another directory

- Double-click a directory icon.

- *Or,* select a directory icon, then choose Open In Place from the Actions menu or from the directory's popup menu. (Open New View opens a new File Manager window for the directory.)

- *Or,* double-click a segment of the current path. For example, if the current directory is **/users/aaron/.vue/types**, you can change to the **/users/aaron/** directory by double-clicking the word **aaron**.

- *Or,* select the path above the icon area, type the directory you want to view, then press Enter.

## To open a terminal window in the current directory

- Choose Terminal from the Directory menu.

This opens a Terminal Emulator window with the same current working directory as the File Manager window.

To close a Terminal window, choose Close from the window menu, or execute the **exit** command.

**See Also**

- "Starting and Stopping a Terminal Emulator" in Chapter 8 describes how to open and use terminal emulators.
- "To change the default Terminal Emulator" in Chapter 18 lists the steps to change your default Terminal Emulator settings.

## To execute an action for a file or directory

- Using the Actions menu:

  1. Select the object's icon.
  2. Choose the action you want to execute from the Actions menu.

- *Or,* choose the action you want from the object's popup menu.

- *Or,* double-click the object's icon to invoke its "default" action (which is the first action listed in the Actions menu).

**See Also**

- "To select a single file or directory"
- "To open a file or directory"

# Using the Trash

The Trash Can collects all of the files and directories that you delete. They are not actually removed from the file system until the trash is "emptied." You can empty the trash manually, or it is emptied automatically when you log out of HP VUE.

If you change your mind, and want to restore a file you've put in the trash, you can *restore* it if the Trash hasn't been emptied.

The trash control in the Front Panel lets you drop objects to delete them. Choosing the trash control displays the Trash Can window, which lists the files and directories currently in the trash.

> **NOTE**
>
> The Trash Can is *not* available in HP VUE Lite.

## To open the Trash Can

- Choose the trash control in the Front Panel.

- *Or,* choose Show Trash from File Manager's File menu.

There is only one Trash Can window, shared by all File Manager windows.

## To remove a file or directory (to the trash)

- Drop the object's icon on the trash can in the Front Panel.

- *Or,* select the object's icon, then choose Delete To Trash from the File menu.

- *Or,* choose Delete To Trash from the object's popup menu.

**For HP VUE Lite**    Since there is no Trash Can in HP VUE Lite, you must use the **rm** command (remove) to remove files. Open a Terminal Emulator window, then execute the **rm** command:

   **rm** *files*

Where *files* is one or more file names, separated by spaces.

**CAUTION**    Files removed with the **rm** command are not placed in the Trash Can, and *cannot* be restored.

## To restore a file or directory from the trash

1. Open the Trash Can.

2. Select the object(s) you want to restore in the Trash Can's list.

3. Choose Restore from the Trash Can's File menu.

4. Close the Trash Can by choosing Close from the File menu.

Restoring an object from the trash returns it to its original location and removes it from the Trash Can list.

**NOTE**

If you delete a file on a remote system, and then remove access to that system, you cannot restore the file until access to the system is reestablished.

# To empty the trash

1. Open the Trash Can.

2. Select the object(s) you want to empty from the trash:

   • Select individual objects in the list.
   • *Or,* choose Select All from the File menu.

3. Choose Remove from the File menu.

4. Close the Trash Can by choosing Close from the File menu.

If there are no objects in the trash, some menu commands are inactive.

**CAUTION**

When you remove a file from the trash, it *cannot* be recovered.

# Accessing Files on Remote Systems

If your system is part of a network, you may need to access files on other systems. You can use File Manager to browse directories on remote systems and to open remote files.

## To open a directory on a remote system

- Using the Remote Systems command:
  1. Choose Remote Systems from the Directory menu. File Manager displays a new window. Each directory in the new window represents a remote system you can access.
  2. Open the directory representing the system you want—you are now working on the remote computer's file system. Work your way to the directory you want to view.

- *Or,* by typing a new path:
  1. Select the path (just below the menu bar) to make it active for editing. (If you are not using a mouse, choose Fast Change To from the Directory menu.)
  2. Replace the current path with the path to the remote system, using a colon to separate the host name from the path.

     *hostname:/path*
  3. Press Enter.

**NOTE**

Your system administrator must configure you system to provide access to other systems on your network.

**Example**

To change to the directory **/users/sonja/** on the machine **host28**, click the path area or choose Fast Change To from the Directory menu.
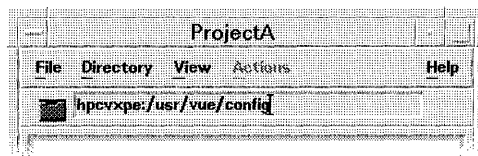


Double-click the current path to select it (the selected text is overwritten as soon as you begin to type). You can also use Backspace to erase the current path.

Now, type the new path: **host28:/users/sonja**. When you press Enter, the new directory is displayed.

If the machine **host28** has not been configured to provide network access, File Manager displays an error, telling you that the directory is invalid.

## To transfer a file to a remote system

- Drag the file's icon to the new location:
  1. Open two File Manager views, one showing the file you want to transfer, the other showing the destination on the remote system.
  2. Drag the file's icon to the new location. If you are making a copy, hold down (Ctrl) while dragging. If you are making a symbolic link, hold down (Shift) while dragging.

- Use the Copy command:

  1. Select the file's icon.
  2. Choose Copy from the File menu. File Manager displays the Copy File dialog, prompting you for the new name.
  3. Enter the host, path, and file name, using the following format:

     *hostname*: */path/filename*

     For example: `host5:/users/aaron/myfile`
  4. Choose OK.

---

# To add access to a remote system

1. On the remote system, *export* the disk volumes that you want available by editing the `/etc/exports` file.

2. On the local system, *mount* the remote disk volumes by editing the `/etc/checklist` file, creating directories as mount points, then executing the `mount` command.

**Notes**

- This task requires you to have superuser permission on the local and remote systems.
- File Manager recognizes only systems mounted as `/nfs/`*hostname*. File systems mounted in other locations are not recognized as "remote systems," however, you can navigate to them just as you can any other directory.
- If you are not familiar with the procedures to "export" and "mount" network file systems, refer to the system administration documentation or online help for your computer. You may also find useful information in commercial books on the Network File System (NFS).

# To specify a remote directory or file

- Use the syntax:

  *hostname*:*/path/filename*

  This syntax for specifying remote files and directories is used in action and filetype definitions and in any File Manager dialog where a file name is required.

**Example**

To access the file **/tmp/data1** on remote host **scribe**, use this name:

```
scribe:/tmp/data1
```

**See Also**

- Refer to the networking documentation for your operating system for more information about domain naming.

# Using the Desktop

The Desktop lets you put any file or directory icon directly on the backdrop of the current workspace for quick access. Any icon you drop on the Desktop stays where you put it.

> **NOTE**
>
> The Desktop is not available in HP VUE Lite.

**A Desktop in Each Workspace**

Since each workspace has its own Desktop, you can set up your working environment by putting the files and directories in the workspaces where you use them most. Objects can be placed on the Desktop in more than one workspace, if needed.

Placing an object on the Desktop does not alter the original file or directory. In fact, the icon that appears on the Desktop is really just a shortcut for accessing the real file or directory. Any operation you perform on the Desktop object, is actually performed on the file or directory it represents.

**Popup Menus**

Each Desktop object has its own popup menu, displayed with the right mouse button or by pressing F4 when the icon has the highlight. This menu contains commands for manipulating the object, including all of the actions that appear in the File Manager's Actions menu when the object is selected.

The Destkop popup menu is similar to the popup menu available within File Manager windows, but contains a few different commands.

## To put an object on the Desktop

1. Switch to the workspace where you want the object.

2. Then:

   - Drop the object's icon on the workspace backdrop.
   - *Or,* select the object, then choose Put On Desktop from the File menu or from the object's pop-up menu.

   You can repeat these steps to put the object on the Desktop in as many workspaces as you want.

## To display the pop-up menu
## for a file or directory

**Mouse**

1. Point to the object's icon.

2. Press *and hold* mouse button 3.

3. Choose a command from the menu by dragging to the command then releasing the mouse button.

**Keyboard**

1. Move the highlight to the object's icon:

   - For an object in a File Manager window, use (Alt)+(Tab) to highlight the File Manager window, then use the (Tab) and direction keys to move the highlight to the correct icon.
   - For a Desktop object, use (Alt)+(Tab) to move the highlight.

2. Press (F4).

3. Choose a command from the menu by using the direction keys to highlight the command, then press Enter.

   To cancel the menu without choosing a command, press (Esc).

File Manager
Properties...
Put On Desktop
Annotate...
Review Annotations...
Delete To Trash
Execute
Edit

**Each icon in a File Manager window has a popup menu.**

Desktop Object
Remove From Desktop
Open File's Folder
Rename
Annotate...
Review Annotations...
Open
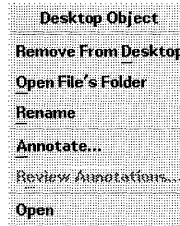
**Each icon on the Desktop also has popup menu.**

# To remove an object from the Desktop

1. Switch to the workspace where the object resides.

2. Locate the object's icon on the Desktop

3. Choose Remove From Desktop from the object's popup menu.

Removing an object from the Desktop does not alter the original object. That is, the object still exists.

# Locating Files

Keeping track of many files can quickly become too much to remember. File Manager helps by providing the ability to search for a file or directory by name. You can also search for files based on the contents of the file.

## To find a file by name

1. Choose Find from the File menu. File Manager displays the Find dialog.

2. Enter the name of the file or directory you want to find into the File Name field.

3. If you want the search to include symbolic links, choose On from the Follow Links options menu. Otherwise, symbolic links are ignored.

4. Enter the directory where you want the search to begin into the Search Directory field. (By default, this field contains the current working directory of the File Manager view.)

5. Choose Start.

File Manager begins searching the Search Directory and its subdirectories for files that match the name you've given. As matches are found, they are listed in the Files Found list. You may stop the search at any time by choosing Stop.

**Matching Patterns**    When you specify a file or directory name, you can include wildcard characters, such as asterisk (*) and question mark (?). The * matches any string of characters (including no characters), and ? matches any single character.

Examples:

| | |
|---|---|
| vue* | Matches all names that begin with the string vue. |
| *.vf | Matches all names that end with the .vf extension. |
| *.??? | Matches all names that have a three-character dot extension. |

**Tip for Advanced Users**

The file name and contents can be specified using the same *regular expression* syntax allowed by the `find` command. (Refer to the `find` man page for more information.)

# To find a file by contents

1. Choose Find from the File menu. File Manager displays the Find dialog.

2. Enter the string you want to search for into the File Contents field. Case is ignored for this string—that is, upper- and lower-case letters are equivalent.

3. If you want the search to include symbolic links, choose On from the Follow Links options menu. Otherwise, symbolic links are ignored.

4. Enter the directory where you want the search to begin into the Search Directory field. (By default, this field contains the current working directory of the File Manager view.)

5. Choose Start.

File Manager begins searching the Search Directory and its subdirectories for files that contain the File Contents string. As matches are found, they are listed in the Files Found list. You may stop the search at any time by choosing Stop.

If you leave the File or Directory Name field empty, File Manager searches every file within the Search Directory. If you know the name (or a partial name) of the files that contains the search string, you may be able to speed up the search by also providing a file name (or pattern) in the File or Directory Name field. File Manager examines only the files that match the File or Directory Name field.

**Tip for Advanced Users**

The file name and contents can be specified using the same *regular expression* syntax allowed by the **find** command. (Refer to the **find** man page for more information.)

# Altering File Manager Views

As you become more familiar with File Manager, you may want to change some settings to better suit your needs or personal preferences. The Preferences dialog lets you make these changes.

You may also want to alter your File Manager views by specifying which filetypes you are not interested in seeing. You can hide certain files and directories using the Filter List.

## To change the order icons are sorted

1. Choose Set Preference from the View menu.

2. Select the sort order you prefer from the Order box.

   - Alphabetically sorts all icons alphabetically by name, A-Z (ascending) or Z-A (descending). Upper-case letters are sorted first.
   - By File Type groups icons according to file type. Within each group of similar types, the icons are sorted alphabetically.
   - By Date sorts icons according to the last time the files were modified, oldest to youngest (ascending) or youngest to oldest (descending).
   - By Size sorts icons by file size, smallest to largest (ascending) or largest to smallest (descending).

3. Select the sort direction, from the Direction box: Ascending or Descending.

4. Choose OK.

If you have Placement set to As Placed, icons are sorted only when you choose Clean Up from the View menu or when you choose Apply in the Preferences dialog. If Placement is set to Sorted Grid, the icons are sorted each time there's a change to the directory's contents or when you choose Reread from the Directory menu.

## To re-sort files and directories (clean up)

- Choose Clean Up from the View menu.

The Clean Up command sorts the objects in the current view (according to the settings in the Set Preferences dialog) and lines them up in a grid pattern. The settings in other File Manager windows are not affected.

The Clean Up command is inactive if you have the Placement preference set to Sorted Grid.

## To change the style of a File Manager view (preferences)

1. Choose Set Preference from the View menu.

2. Select your preferences (Placement, Show, View, Order, and Direction).

3. Choose OK.

If you want to view the results before closing the Set Preferences dialog, choose Apply.

# To save the current preferences as your default

1. Set the preferences that you want (using Set Preferences and Modify Filter List in the View menu).

2. Choose Save Settings from the View menu. File Manager prompts to verify that you want the current settings saved.

3. Choose OK.

After saving the current preferences, all new File Manager views you open uses the new preferences and filter list.

# To show hidden files and directories

- Choose Show Hidden Files from the View menu.

The Show Hidden Files command is a toggle, so to hide the files, choose Show Hidden Files again.

Hidden files and directories are those whose filetypes are selected in the *filter list*. By default, the hidden filetypes are DOT_FILE, DOT_DIR, and CURRENT_DIRECTORY.

# To hide certain types of files and directories (filter list)

1. Choose Modify Filter List from the View menu. File Manager displays the Modify Filter List dialog

2. In the horizontal list of icons, select the icons for the filetypes you want to be hidden.

3. Optional: Enter a name pattern into the Filter String field specifying additional file and directory names to be hidden.

4. Choose OK.

If you want to view the results before closing the dialog, choose Apply.

To reset to the default filter list, choose Reset. This resets the default values but does not apply them until you choose Apply or OK.

---

**NOTE**

The filter list specifies which files are *not* to be displayed. If you select all filetypes or enter * into the Filter String field, *no files are displayed.*

---

Advanced users: The Filter String may be any *regular expression*. Refer to the `regexp` man page (section 5) for more information.

# 6

# Using the Toolboxes to Run Applications

# Using the Toolboxes to Run Applications

The toolboxes are containers for applications and utilities.
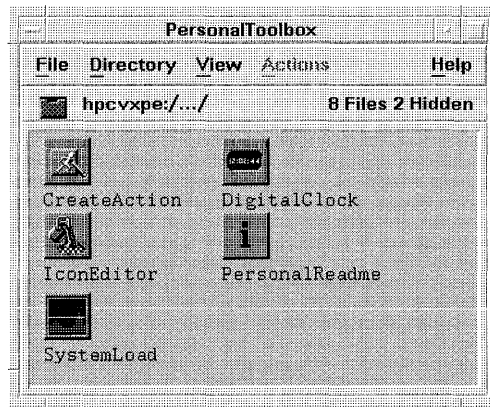


**The toolboxes contain icons representing applications and utilities.**

There are three toolboxes:

| | |
|---|---|
| Personal Toolbox | Your personally-configurable toolbox. You can create actions yourself or copy them from other toolboxes. |
| General Toolbox | Contains applications and utilities built into HP VUE or provided by your system administrator. |
| Network Toolbox | Lets you access actions on other systems. |

**The toolboxes are based on File Manager**    You can think of the toolboxes as File Manager views of special directories set aside for applications and utilities. However, within a toolbox, you cannot go up past the top level of the toolbox, which is represented in the status line by **/ ... /**.

**The toolboxes contain action icons**    Each application or utility is represented by an icon called an **action icon**. Actions and their icons are HP VUE's way of creating a visual representation for applications, utilities, and other commands.

**See Also**

- "Basic File Management Skills" in Chapter 5 covers manipulating files (and actions icons).

- "Using Tools in HP VUE Lite" covers using the HP VUE Lite Toolbox.

- "General Action and Filetype Concepts" in Chapter 19 explains what actions are and how they integrate applications into HP VUE.

# The Personal Toolbox

The Personal Toolbox is a container for applications and utilties you use frequently or create yourself.

Initially, the personal toolbox contains:

- Actions you personally place there.
- Several actions from the General Toolbox that you may want to use frequently.
- Other actions provided by your system administator.

You have control over the contents of your Personal Toolbox. You can add and remove tools, and organize the contents by creating subdirectories.

**Previous HP VUE 2.01 users**    Actions you created in HP VUE 2.01 are preserved in HP VUE 3.0.

**See Also**

- "To display HP VUE 2.01 tools from the HP VUE 3.0 toolboxes" in Chapter 28 explains how to access your HP VUE 2.01 actions using the toolboxes.

# To open the Personal Toolbox

- Choose the Tools control in the Front Panel.
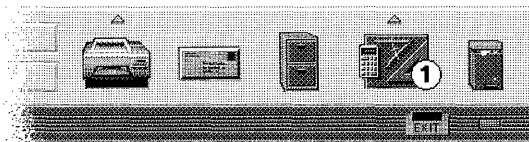
**Choose the Tools control ① to display the Personal Toolbox.**

- *Or*, choose Personal in the Toolboxes subpanel.

---

## To start an application or utility

- Double-click an action icon. If the application requires file data, a prompt is displayed.

- *Or*, drop a data file onto the action icon (if the application or utility accepts file data).

**To get help on an action icon**
- Select the action and press (F1).

---

## To add a subdirectory to the Personal Toolbox

1. Choose New from the Directory menu.

2. In the Personal Toolbox - New Directory dialog box, enter a name into the New Directory Name Field.

3. Choose OK or press (Return).

---

## To make a personal action more readily available

- Put the action icon on your Desktop.

- *Or*, add the action to your Front Panel.

- *Or*, copy ((Ctrl)+drag) the action icon to a directory where you'll be using it frequently.

It is often useful to copy action icons to directories containing application data.

- If the action takes a data file, you can drop the data file on the action icon using only one File Manager view.

- If the action prompts for data when double-clicked, you can supply a filename (without the entire path). The action assumes the data file is located in the same directory as the action icon.

**See Also**
- "To put an object on the Desktop" in Chapter 5.
- Chapter 17 explains how to add controls that start actions to the Front Panel.

# To copy a general or network action to the personal toolbox

- Drop the action icon on the Tools control in the Front Panel.
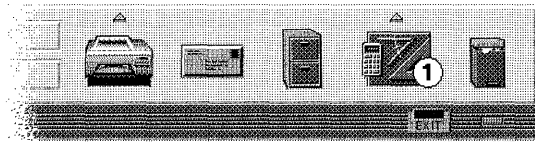


**Dropping an action icon on the Tools control ① copies the action to your Personal Toolbox.**

- *Or*, open both the Personal toolbox and the toolbox containing the icon. Copy ([Ctrl]+drag) the action icon to the Personal Toolbox.

# To add an application to the Personal Toolbox

1. If you have not already done so, install the application on your system and find out The command line you would execute to start the application from a Terminal Emulator.

2. Double-click CreateAction in the Utilities subdirectory of the System Toolbox to display the Create Action dialog box.
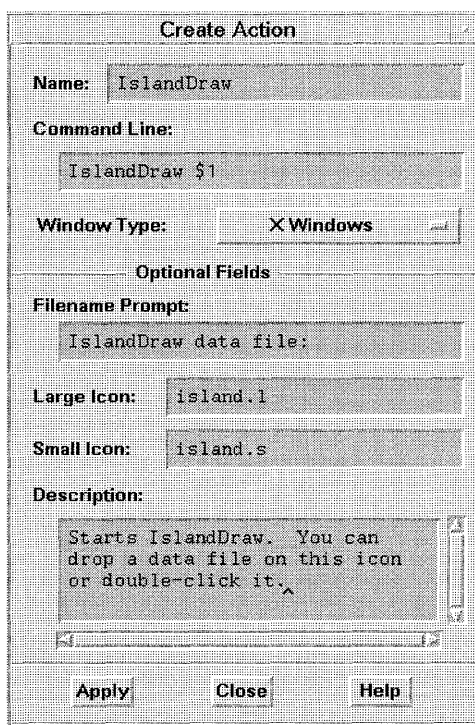


**Use Create Action to add an application to your Personal Toolbox.**

3. In the Name field, type a unique name for the action.

4. In the Command Line field, type the command that would start the application. If the the command takes a data file, substitute $\$n$ for each argument, where $n$ is 1 for the first data file, 2 for the second, etc. (Example: `Wingz $1`.)

5. Use the Window Type options menu to select how the action's output will be displayed. Select X Windows if the application creates it's own window; select Terminal if the application runs in a Terminal Emulator window.

6. Choose Apply.

   • An icon for the action will appear in your Personal Toolbox.

   • The action definition will be put in a new configuration file. A dialog box will tell you the location of the new configuration file.

7. Test the action by doubl-clicking the action icon. If it doesn't work properly, edit the fields in the Create Action dialog and Apply your changes.

8. When you are satisfied that the action is working properly, choose Close.

If the application is located on another system, you can use `remsh` in the Command Line. However, HP VUE actions provide a better way to start applications on other systems.

**See Also**

   • "Using Actions to Run Remote Applications" in Chapter 22 covers creating actions for applications located on other systems.
   • "Creating a Simple Action" in Chapter 19 explains additional features of Create Action.
   • "Creating Actions Manually" in Chapter 21 covers creating actions by editing a configuration file.
   • These are step-by-step examples for creating actions:
     □ "Example: Creating an action for an application" in Chapter 19
     □ "Example: Creating a personal action and filetype" in Chapter 19

# The General Toolbox

The General Toolbox is a container for:

- Actions built into HP VUE.

- Actions placed there by your system administrator for system-wide use.

You can move about in the General Toolbox and invoke actions from it. However, you are not allowed to write to it (unless you are logged in as superuser). Therefore, you cannot add actions to it or reorganize it.

Some of the actions in the General Toolbox require you to be logged in as superuser. If you try to start one of these actions logged in as yourself, a dialog box will prompt you for the root password.

**See Also**

- "To start an application or utility" explains how to start actions.

## To open the General Toolbox

1. Display the Toolboxes subpanel by choosing the up arrow above the Tools control.

2. Choose General in the Toolboxes subpanel.
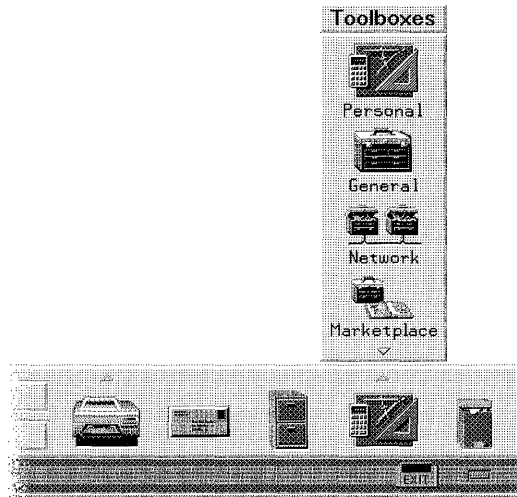
**Choose the "General" control to display the General Toolbox.**

## To make a general action more readily available

- Put the action on your Desktop.
- *Or*, copy the action to your Personal Toolbox.
- *Or*, add the action to your Front Panel.

**See Also**

- "To put an object on the Desktop" in Chapter 5 covers how to put an action on the Desktop.
- "To copy a file" in Chapter 5 explains how to copy actions.
- Chapter 17 explains how to add controls that start actions to the Front Panel.

# The Network Toolbox

The Network Toolbox lets you use actions that are defined on other systems.

The top level of the Network Toolbox contains a subdirectory for each remote system whose actions you can access. If your system has not been set up to access remote actions, no subdirectories will be present.
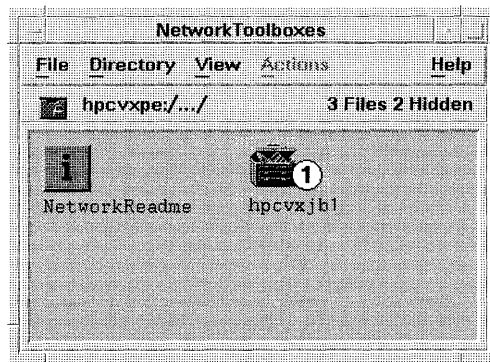


**The top level of the Network Toolbox contains a subdirectory ① for each system whose actions you can access.**

**See Also**
- "To start an application or utility" explains how to start actions.
- "Importing and Exporting Actions" in Chapter 22 covers how to configure the Network Toolbox.

# To open the Network Toolbox

1. Display the Toolboxes subpanel by choosing the up arrow above the Tools control.

2. Choose Network in the Toolboxes subpanel.

3. Double-click the icon for the system that contains the action.
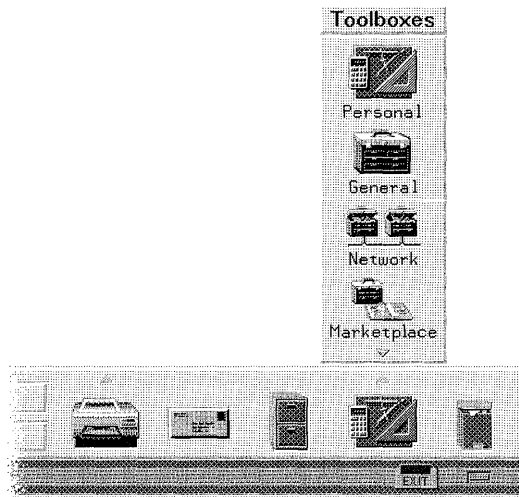


**Choose the "Network" control to display the Network Toolbox.**

# To make a network action more readily available

- Put the action on your Desktop.
- *Or*, copy the action to your Personal Toolbox.
- *Or*, add the action to your Front Panel.

**See Also**

- "To put an object on the Desktop" in Chapter 5 covers how to put an action on the Desktop.
- "To copy a file" in Chapter 5 explains how to copy actions.
- Chapter 17 explains how to add controls that start actions to the Front Panel.

# Using Tools in HP VUE Lite

HP VUE Lite provides a Front Panel control and subpanel for accessing tools.
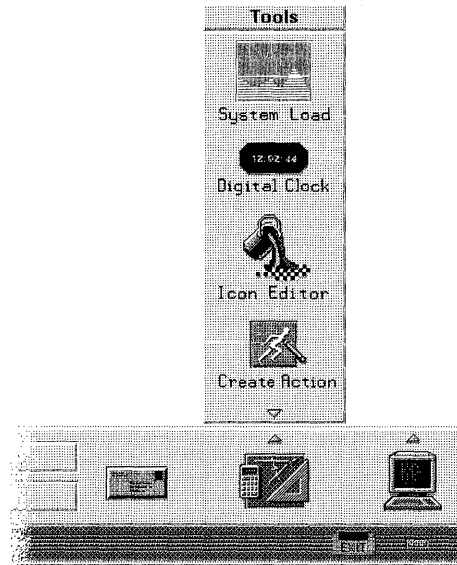


**Use the HP VUE Lite Tools control and subpanel to run applications.**

# To start an application or utility in HP VUE Lite

- Choose the Tools control on the top row of the Front Panel, or one of the controls in the Tools subpanel.

The Tools control is initially *not* attached to an application; you or your system administrator can configure it to run the application of your choice. You can also change the contents of the Tools subpanel so that it contains applications you use often (see "To add an application to HP VUE Lite").

**See Also**
- "To run an action from the command line" in Chapter 23 explains how to run actions from a Terminal Emulator.

# To add an application to HP VUE Lite

Adding an application to the Tools involves configuring the Front Panel to include the new tool.

**See Also**
For information on adding applications to HP VUE Lite:

- "To assign an application to the HP VUE Lite Tools control" in Chapter 23
- "To add an application to the Tools subpanel" in Chapter 23

For information on configuring the Front Panel:

- "To add or remove a control from an existing subpanel" in Chapter 17.
- "Defining Front Panel Controls" in Chapter 17
- "To create a push button for an application" in Chapter 17

For information on creating actions:

- "To use Create Action in HP VUE Lite" in Chapter 23
- "Example: Creating an action for an application" in Chapter 19 is a step-by-step example.

# 7

Using Text Editor

# Using Text Editor

Text Editor is appropriate for small editing tasks such as writing memos and editing resource files.

The Text Editor window contains:

- Title bar, showing the name of the current document.
- Menu bar, containing five menus: File, Edit, Search, Format, and Help.
- Tablet, the area where you write and edit text.
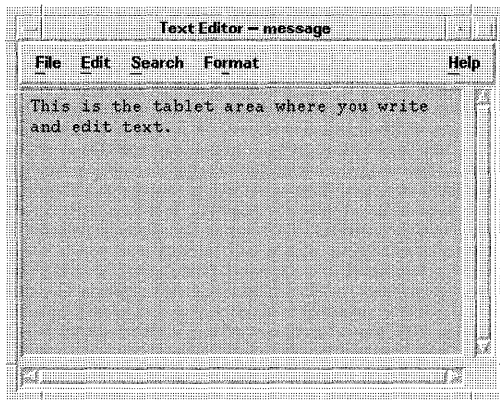
# Starting and Stopping Text Editor

Text Editor is readily accessible from many locations in HP VUE:

- The Front Panel.
- File Manager.
- Terminal emulator.

Once Text Editor starts, the tablet area is ready to receive text entry or edits.

## To start Text Editor

**From the Front Panel**
- Choose the Text Editor button in the Front Panel.



**Start Text Editor by choosing its button in the Front Panel.**

**From File Manager**
- Double-click a data file icon.
- *Or*, drag a file icon and drop it on the Text Editor button in the Front Panel.

7-3

# Starting and Stopping Text Editor

Text Editor is readily accessible from many locations in HP VUE:

- The Front Panel.
- File Manager.
- Terminal emulator.

Once Text Editor starts, the tablet area is ready to receive text entry or edits.

## To start Text Editor

**From the Front Panel**   • Choose the Text Editor button in the Front Panel.
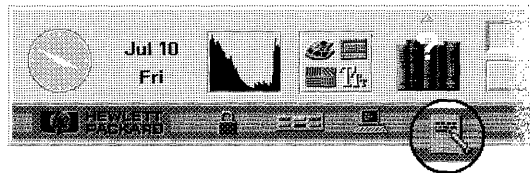


**Start Text Editor by choosing its button in the Front Panel.**

**From File Manager**   • Double-click a data file icon.

• *Or*, drag a file icon and drop it on the Text Editor button in the Front Panel.

**From a terminal emulator**
- Type the following after a command-line prompt:

  vuepad *filename* &    *The filename is optional.*

---

## To close Text Editor

1. Save the open document.
2. Choose Exit from the File menu.

**See Also**
- "To save a document".

# Opening and Saving Documents

Opening a document displays the contents of a selected file in the tablet area. You can open new or existing documents.

Saving a document stores the edited contents into a file. You can save a document to its original file or to a new file.

## To open a new document

- Choose New from Text Editor's File menu.

New documents are titled **Text Editor - (unnamed)**.

## To open an existing document

**From Text Editor**

1. Choose Open from the File menu to display a dialog.

2. Enter or select the document name.

3. Choose OK.

The Open File dialog lists files and directories on your system. You can browse the lists to locate a file.

Text Editor – Open a File

File Filter:

/users/anna/*

Directories                      Files

/sd3/users/anna/Mail            FebCashFlod
/sd3/users/anna/bitmaps         FileToOpen
/sd3/users/anna/reports         JanTravel
                                MailToDex
                                appointments
                                calendar

File to Open:

/users/anna/FileToOpen

OK        Filter        Cancel        Help

**From File Manager**     • Display the document's file icon in a File Manager window.

• Do one of the following:

☐ Double-click the document's file icon.

☐ *Or*, select the document, then choose Edit from File Manager's Actions menu.

☐ *Or*, drag the document to Text Editor's tablet.

☐ *Or*, drag the document to Text Editor's button in the Front Panel.

# To save a document

**To a new file**

1. Choose Save As from the File menu to display the Save As Dialog dialog box.

2. If you used word wrap, choose whether or not to add newline characters to the end of word-wrapped lines.

3. Supply the complete file name in the Save File As field.

4. Choose OK.
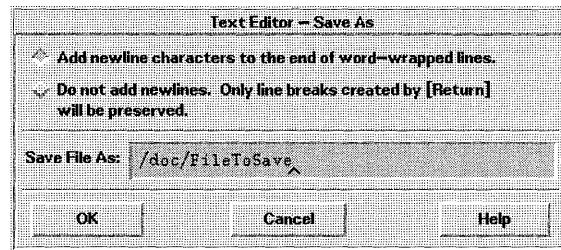


**Use the Save As Dialog Box to Save a New File.**

**To the original file**

- Choose Save from the File menu.

Your edited document is automatically saved to its original file. If you used word wrap, you'll be asked whether or not to add newline characters to the end of word-wrapped lines.

# Entering and Editing Text

As you create a document, you enter and edit text. A text cursor (I) shows your current position, and a marker (^) shows where you last edited, inserted, or selected text.

## To enter new text

- Start typing. Text is inserted at the current cursor position.
- To start a new line, press (Enter) or use word wrap.
- To start a paragraph, press (Enter) twice.

## To turn on/off word wrap

- Choose Word Wrap from the Edit menu.

When word wrap is on, typed words continue onto the next line when the cursor nears the right edge of the tablet.

Using word wrap affects how you save your document. When saving, you can:

- Add a **newline character** to the end of each word-wrapped line, or
- Preserve only line breaks that were created by pressing (Enter).

# To move around in a document

**Moving the cursor**

- Point to the new location in the tablet and click mouse button 1.
- *Or*, use the cursor movement keys:

| Key ... | Cursor Movement ... |
| --- | --- |
| ⬆ | Up one line. |
| ⬇ | Down one line. |
| ◀ | Left one character. |
| ▶ | Right one character. |
| CTRL + ▶ | Right one word. |
| CTRL + ◀ | Left one word. |
| CTRL + ⬇ | Beginning of the next paragraph. |
| CTRL + ⬆ | Beginning of the previous paragraph. |
| ⬇ or Home | Beginning of the current line. |
| f7 | End of the current line. |
| CTRL + ⬇ or CTRL + Home | Beginning of the document. |
| CTRL + f7 | End of the document. |

**Scrolling a document**

- Use the scroll bars.
- *Or*, use the scrolling keys:

| Key ... | Scroll Movement ... |
| --- | --- |
| Prev or Pg Up | Up one page. |
| Next or Pg Dn | Down one page. |
| CTRL Prev or CTRL Pg Up | Left one page. |
| CTRL Next or CTRL Pg Dn | Right one page. |

# To edit text

**Editing characters**
- Use the editing keys for simple editing.

| Key | Action |
|---|---|
| (Back space) | Deletes the character before the cursor. |
| (Delete char) or (Del) | Deletes the character following the insertion cursor. |
| (CTRL)+(Delete char) | Deletes all the characters from the cursor to the end of the current line. |
| (Delete line) | Deletes the line containing the cursor. |

**Selecting text**

Text must be selected before using move (cut and paste), copy, delete, or clear Edit menu commands.

**Selecting with the mouse**
- Text block:
  1. Move the cursor to the beginning of the text block.
  2. Hold down mouse button 1 and drag the cursor to the end of the text block.

     To change the amount of selected text, hold down (Shift) and mouse button 1 as you drag the cursor to the new ending point for the selection.

     To unselect text, click any empty area in the window.
- Word: Double-click the word.
- Line: Triple-click the line.
- All text: Quadruple-click the tablet.

**Selecting with the keyboard**
- Text block:
  1. Move the cursor to the beginning of the text block.
  2. Hold down (Shift) plus an arrow key to move the cursor to the end of the text block.
- All text: Press (Shift)+(CTRL)+(/).

**Replacing text**

1. Select the text.

2. Type the replacement text.

**Moving text (Cutting and Pasting)**

1. Select the text.

2. Choose Cut from the Edit menu. The text is erased from the tablet and stored temporarily.

3. Move the cursor to the new location, then choose Paste from the Edit menu.

**Copying text**

1. Select the text to be copied.

2. Choose Copy from the Edit menu. A copy of the text is stored temporarily.

3. Move the cursor to the new location.

4. Choose Paste from the Edit menu.

**Deleting text**

1. Select the text.

2. Choose Delete from the Edit menu.

Deleted text cannot be pasted, but it can be recovered with the Undo command.

**Clearing text**

1. Select the text to be cleared.

2. Choose Clear from the Edit menu.
   *Or*, press (Clear display).

Use Clear to replace a text block with spaces or blank lines.

**Undoing an edit**

• Choose Undo from the Edit menu.

Undo reverses the last cut, paste, clear, delete, replace, include, or format operation.

# To include a file

1. Position the cursor where you want to insert the document.

2. Choose Include from the File menu to open a dialog.

3. Enter or select the document name.

4. Choose OK.

The Include File dialog lists files and directories on your system. You can browse the lists to locate a file.
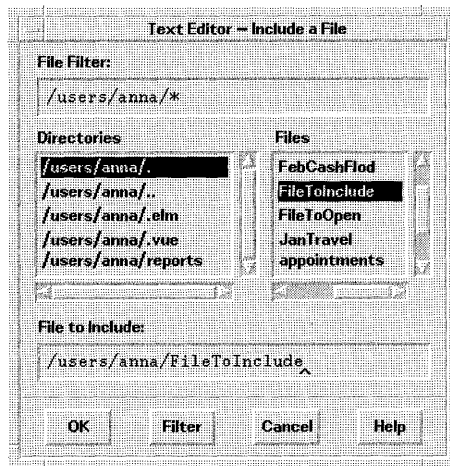


**Dragging a file from File Manager**

1. Open a File Manager displaying the file you want to include.

2. Position the text insertion cursor in the document where you want the file inserted.

3. Hold down (CTRL) and mouse button 1 as you drag the file from File Manager to Text Editor's tablet.

4. Release the mouse button, then release (CTRL).

# To find text

1. Choose Find from the Search menu to open the Find dialog box.

2. Type the text to find in the Find field. The search is not case-sensitive, so you don't need to worry about upper- or lowercase letters.

3. Choose Find to locate the first occurrence of the text.

4. To find additional occurrences:

   - Choose Find again.
   - *Or*, choose Next from the Search menu.

Find starts at the current cursor position and continues through the entire document.
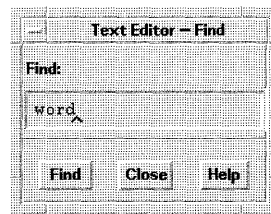


# To replace text

1. Choose Replace from the Search menu.

2. Type the text to locate in the Find field. The search is not case-sensitive, so you don't need to worry about upper- or lowercase letters.

3. Type the replacement text in the Replace With field.

4. Choose Find to locate an occurrence of the text.

5. Choose Replace.

Find and Replace starts at the current cursor position and continues through the entire document.
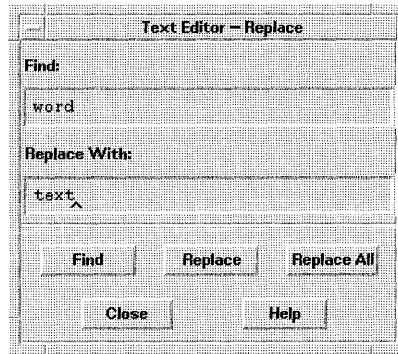


## To correct misspelled words

1. Choose Spell from the Search menu to open the Spell dialog listing misspelled or unrecognized words.

2. Select the word in the Misspelled Word list. The selected word is automatically pasted into the Find field.

3.. Choose Find to locate and highlight the word in the document.

4. Type the correct word into the Replace With field.

5. Choose Replace.

6. Repeat steps 2 - 5 to correct other words.
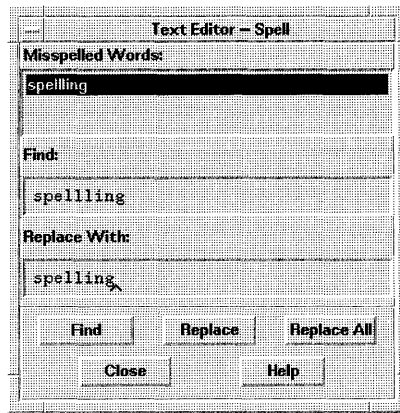
**Correct misspelled words with the Spell dialog.**

---

**NOTE**

The Spell dialog is only available for English.

---

# To use special characters in your text

1. Position the cursor in the text where you want the special character.

2. Type CTRL+Q, then the special character.

There are two kinds of characters you may want to include in your document:

- Special instructional characters, such as one that forces a form feed when printing.
- Special language characters, such as the umlaut used in German text.

**Example**

To embed a form feed character into your document:

1. Position the cursor.

2. Type (CTRL)+(Q), then (CTRL)+(L).

---

**NOTE**

CTRL+Q has no effect on key sequences used in Text Editor as menu **mnemonics** or **accelerators**.

The following English key sequences are unchangeable.

- (Alt)+(E), (Alt)+(F), (Alt)+(H), (Alt)+(R), and (Alt)+(S).
- (CTRL)+(F).

---

# Formatting and Printing Documents

Once your editing is complete, format your document by choosing margins and justifications.

## To format a document

1. Choose Settings from the Format menu.
2. Enter margins and select justifications.
3. Do one of the following:

   - To format a paragraph, place the cursor in the paragraph, then choose Paragraph.
   - To format the entire document, choose All.

To use the settings after the dialog box is closed, choose Paragraph or All from the Format menu.
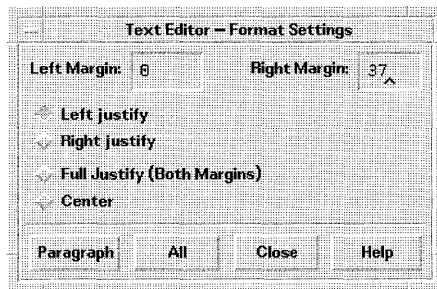


**Choose format settings in the Settings dialog.**

> **NOTE**
>
> The Center option is unavailable for some languages.

## To print a document

1. Open the document in Text Editor.
2. Choose Print from the File menu.

The document prints on your system's printer.

# 8

## Using Terminal Emulators

# Using Terminal Emulators

A **terminal emulator** is an application that displays a window which allows you to use HP-UX commands and cut and paste text within or between terminal emulation windows.

The command line prompt is a special character that is displayed in your terminal emulator after you press ⌈Enter⌋. It can be a **%**, **<**, **$**, or other special character. A small box, called a cursor, shows where characters will appear in the window when you type something on the keyboard.
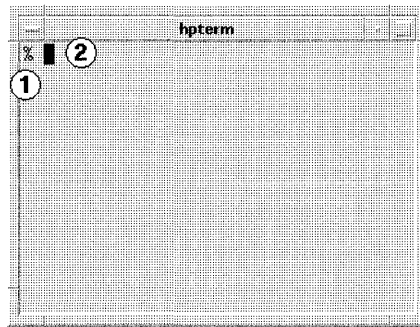


The command line prompt ① and cursor ② appear in terminal windows.

HP VUE provides two terminal emulators:

hpterm           **hpterm** is the default terminal emulator for HP VUE. It displays a window in which command-line prompts and soft keys act like an HP Term0 terminal.

xterm            **xterm** provides a terminal window that emulates DEC and Tektronix terminals.

# Starting and Stopping a Terminal Emulator

You can start a HP VUE terminal emulator either with a Front Panel control, from the Toolbox, or by typing a command. The method of stopping the terminal emulator is the same, regardless of how it was started.

- "Using Terminal Emulators with HP VUE Lite" explains how to start a terminal emulator in HP VUE Lite.

## To start terminal emulator with terminal control

- Click the Terminal control in the Front Panel.
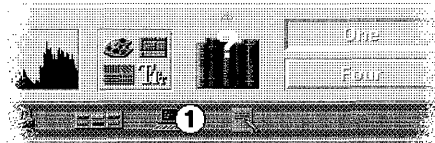


**Use the terminal button ① to start a terminal emulator.**

A terminal window containing a command line prompt and cursor appears. The terminal emulator that appears is the default terminal emulator.

## To open a terminal from a File Manager view

- Choose Terminal from the Directory menu.

This opens a Terminal Emulator with the same current working directory as the File Manager view where it is started.

# To start a terminal emulator from the command line

1. At the command line prompt, type the name of the terminal emulator and any options you want, using:

   hpterm   [*options*] **&**

   Or:

   xterm   [*options*] **&**

   Where:

   *options*          Represents optional items to customize the terminal emulator.

   **&**                  Specifies that the terminal emulator runs in the background, that is, you can continue working in your original window while the terminal emulator is also running.

   The terminal emulator starts in the current workspace unless directed otherwise by options.

**Examples**

The following command starts an **hpterm** window with a black background and a white foreground. (This window will not participate in dynamic color changes through Style Manager since colors are specified directly.)

    hpterm -bg Black -fg White &

The following command starts an **xterm** window in the workspace named "Project Notes":

    xterm -xrm '*workspaceList:  "Project Notes"' &

The following command starts an **hpterm** window on a screen on the system named "LGMCD":

    hpterm -display LGMCD:0.1 &

**See Also**

- Refer to the **xterm** or **hpterm** man pages for details about the options available for each.

## To start a terminal emulator from the Toolbox

• Execute the **hpterm** or **xterm** action in the Utilities subdirectory of the General Toolbox directory.

## To close a terminal emulator window

**Menu button**

• Double-click on the window menu button in the window frame.
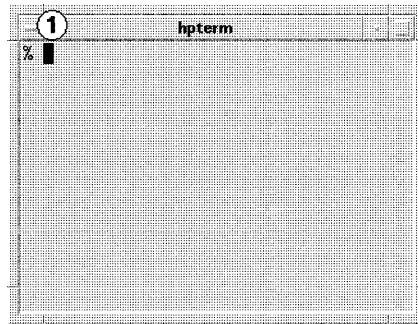


**Close a terminal emulator with the window menu button ①.**

**Command line**

• Type:

    exit (Return)

# Using the Terminal Emulator

The terminal emulator window provides a means to enter HP-UX commands and the ability to cut and paste text between and within windows.

## To cut and paste text

You can select text within one window and copy it to another location within that window, or to another terminal emulator window.

The original text is left in place. If you want to delete it, you must do so manually.

**cutting text**

1. Move the pointer over the first character you want to cut.
2. Press *and hold down* mouse button 1.
3. Drag the pointer over the text you want to move. The text appears highlighted.
4. Release mouse button 1 after all the text you want to cut is hightlighted.

The highlighted text is copied to an internal "clipboard". The text is not removed from your original source.

**pasting text**

1. Position the cursor where you want to insert the text.
2. Click mouse button 2.

A copy of the contents of the clipboard is pasted at the location you indicated. You can make additional copies by repeating the above steps.

## To resize the window contents

- Type the following at the command-line prompt:

  ```
  eval 'resize' (Enter)
  ```

You can change the size of the window using the frame components. However, applications running within the window may not know about the resizing and act as if the window were the original size.

## To start applications in a terminal window

- Enter the command to start the application at the command line prompt.

The general syntax for starting an application is:

  *application* [*options*] **&**

*application*    The application name.

*options*    A list of optional information to be passed to the application.

**&**    Indicates that the application runs in the background; that is, you can continue to use the terminal emulator window while the application is also running.

**Example**    The following example starts a digital clock from the command line:

  ```
  xclock -digital &
  ```

**See Also**    - Refer to the man page or other documentation for each application to find the command and options to use for that application.

# Customizing Your Terminal Emulator

There are several ways in which you can customize your terminal emulator:

- Run the terminal emulator with special options, such as scrollbars.

- Use a different terminal emulator.

- Run the terminal emulator on another system.

- Run the terminal emulator using non-English characters.

- The `hpterm` or `xterm` man page lists all the options available.
- "Using a Non-English Terminal Emulator" in Chapter 27 explains how to run a terminal emulator with non-English characters.
- "Changing the Default Terminal, Editor, and Mailer" in Chapter 18 explains how to change the terminal emulator that appears by default when you press the Terminal control.

## To specify scrollbars

1. Use the EditResources action to specify a `scrollBar` resource for the terminal emulator.

2. Log out, then log back in for the current session. (For home session, set home session, log out, then log back in.)

If the value of `scrollBar` is True, the terminal emulator windows will have scrollbars. If the value is False, they will not have scrollbars.

**Example**    The following line sets scrollbars in all `hpterm` windows:

    hpterm*scrollBar:      True

The following line sets scrollbars only for `hpterm` windows named "localTerminal":

    hpterm*localTerminal*scrollbar:      True

# To display a terminal emulator window on another system

**Using -display option**

- Run **hpterm** or **xterm** with the **-display** option.

The **-display** option has the format:

    -display *host:display.screen*

where:

*host*             is the name of a valid system on the network.

*display*          is the number of the display on the host.

*screen*           is the screen within the display. The default is 0.

You can find these values by typing **env**, and examining the DISPLAY line.

The terminal emulator is running on your system, but the window shows on another system.

The following starts an **hpterm** window on the host computer named **anna**:

    hpterm -display anna:0.1 &

**Using rlogin**

- Run **rlogin** from a terminal window on your system.

You can use an existing terminal emulator window to log into a remote host. Once the window is acting as a terminal to the remote host, you can run applications there, redirecting the display back to your system if you desire.

For example, the following command logs onto a system named **there**, runs the client **xload**, and redirects the display back to your original system. Assume your system is named **here**.

    rlogin there
    xload -display here:0.0

**Using remsh**

- Run the **remsh** command from a terminal window on your system.

The **remsh** command starts a shell on a remote host, performs some client (often starting a terminal emulator on that host), and redirects the display back to your original system if desired. It has the syntax:

**remsh** *remote* **-n** *client* **-display** *system:display.screen*

where:

| | |
|---|---|
| *remote* | The remote host name. |
| *client* | The program you want to run on the remote host. |
| *local:display.screen* | The host and display the results are to be displayed on. |

For example, the following command runs **xload** on the remote host named **there**, and directs output back to your system, **here**.

```
remsh there -n /usr/bin/X11/xload -display here:0.0
```

The **remsh** command is often used when customizing a menu to access other hosts.

- Chapter 22 contains more information about networks and running applications on other computers.

# To set terminal control characters

1. Use the EditResources action to update the **ttyModes** resource.

2. Choose Restart Workspace Manager from the workspace menu.

Because the **hpterm** and **xterm** only *emulate* a terminal, your control characters might not be what you are used to on a physical terminal. The **ttyModes** resource allows you to set control characters for your terminal emulator.

By default, Login Manager sets the following control characters.

Control name    Character

erase             ^H (backspace erases characters)

intr              ^C (interrupt - cancel the current operation and redisplay the command line prompt.)

kill              ^U (stop an operation or application)

start            ^Q (accept keyboard input - used to "continue" an application that has been paused.)

stop             ^S (do not accept keyboard input - used to "pause" an application.)

swtch          ^@ (switch between layers in a shell.)

The "^" character stands for the ⌨Ctrl key. So to interrupt an operation in progress, you press ⌨Ctrl ⌨C.

The syntax for this resource is:

```
ttyModes:  ^C  name
```

where *name* is the control, and *C* is the character. For example, the default value of **ttyModes** describing the above list is:

```
ttyModes: erase ^H intr ^C kill ^U start ^Q stop ^S swtch ^@
```

Additional controls are listed in the **xterm** or **hpterm** man pages.

**See Also**

- Chapter 25 for information about how and where to set resources.

# Using Terminal Emulators with HP VUE Lite

**Terminal control**

- Click the Terminal control on the Front Panel.

  The default terminal emulator will display a window containing a command line prompt and a cursor.
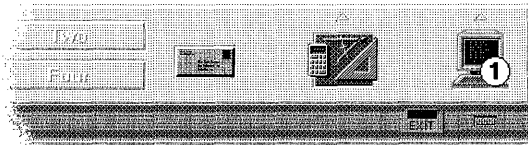


Use the HP VUE Lite terminal control ① to start a terminal emulator.

**Menu**

1. Click on the arrow above the Terminal front panel control.

2. Select one of the choices from the slide up menu:



Make a selection from the Terminals subpanel.

You can start an `hpterm` or an `xterm` terminal emulator.

3. If you chose a remote terminal emulator, a dialog box appears. Enter the name of the system you want to log on to.
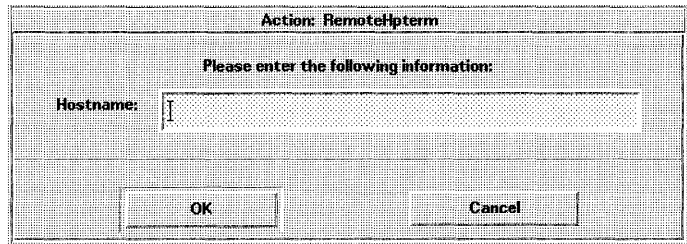


**Enter the host name in the Host Dialog.**

**See Also**

• "To display a terminal emulator window on another system" explains how the terminal window can run on another computer.

**Using Terminal Emulators with HP VUE Lite**

9

**Customizing Appearance
and Behavior**

# Customizing Appearance and Behavior
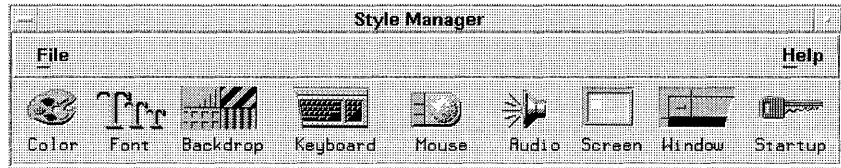


| | |
|---|---|
| Color | Workspace colors and palettes. |
| Font | Application font sizes. |
| Backdrop | Workspace backdrop patterns. |
| Keyboard | Key click volume or character repeat capability. |
| Mouse | Mouse button click settings, double-click speed, pointer acceleration, or pointer movement threshold. |
| Audio | Beeper volume, tone, or duration. |
| Screen | Number of minutes before your screen times out or whether or not your screen is covered and locked at time out. |
| Window | How a window acquires focus, if the window raises when it receives focus, or where window icons are placed. |
| Startup | How your session begins and ends. |

If you're working in an HP VUE Lite session, Style Manager contains four buttons—Color, Backdrop, Font, and Window.

**Starting Style Manager**
- Choose the Style Manager button in the Front Panel.
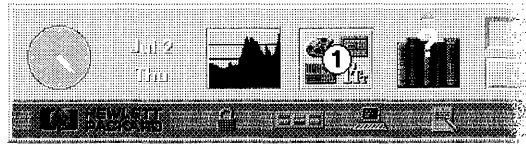
**To start Style Manager, choose its control ① in the Front Panel.**

**See Also**
There are other behaviors you may want to change in your HP VUE working environment, such as changing what a button does in the front panel, or changing your default editor from Text Editor to either `vi` or `emacs`.

- Chapter 17 explains how to customize Front Panel behavior.
- Chapter 18 explains how to change your default Terminal Emulator, Mailer, and Editor.
- "Workspace Manager Menus" in Chapter 24 explains how to add or delete commands from menus.
- "Creating a Simple Filetype" in Chapter 19 explains how to define a file type. When you define a file type, you can specify the actions associated with the file.
- Chapter 14 explains how to change variables affecting your windowing environment.

# Customizing Screen Appearance

The visual elements of your screen can be easily changed with Style Manager:

• Workspace colors.

• Workspace backdrop patterns.

• Font sizes used to label windows and enter text.

## To select a palette

1. Choose Style Manager's Color button.

2. Select a palette from the Palettes list.

3. Choose OK.

Workspace colors come from predefined color palettes. Palettes that come with your system are listed in the Color dialog.
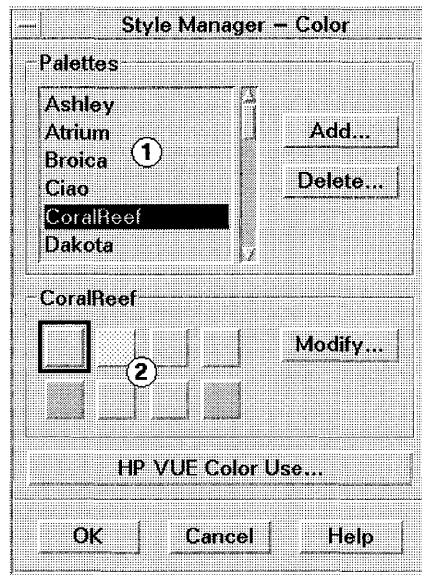
① List of palettes.
② Color buttons, representing the selected palette's colors.

## To modify a palette

1. Select a palette in the Color dialog.

2. Double-click a color button to open the Modify dialog.
   *Or*, click the button, then click Modify.

3. Adjust the settings.

4. Choose OK in the Modify dialog.

5. Repeat steps 2-4 to modify another color button.

6. Choose OK in the Color dialog.
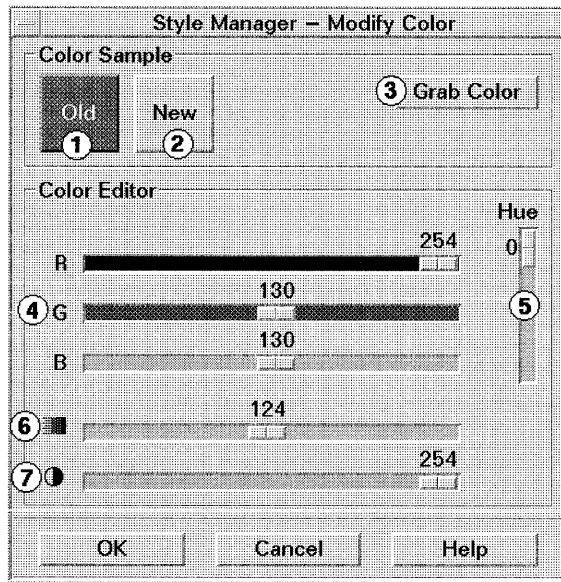
① Shows the current color.
② Shows the new color as you make changes.
③ Displays a pointer for you to choose a color in your workspace.
④ Changes red, green, and blue values.
⑤ Changes red, green, and blue proportions.
⑥ Changes the color's intensity.
⑦ Changes the color's brightness.

**Grabbing a color from the workspace**

1. Choose Grab Color to get the grab pointer.

2. Click the desired color to make it the New color.

## To create a custom palette
## (by copying and modifying an existing palette)

1. Choose Add in the Color dialog.

2. Type the new palette name in the New Palette Name field.

3. Choose OK in the Add Palette dialog.

4. Modify the palette.

5. Choose OK in the Color dialog.

Adding a palette makes a copy of the currently selected palette and gives the copy a unique name. You then create a palette by modifying the copy, thus keeping the original palette intact.

## To delete a palette

1. Select the palette from the Palettes list.

2. Choose Delete.

3. Choose OK in the Delete Palette dialog.

**Restoring a deleted palette**

1. Open a File Manager showing *HomeDirectory*/.vue/palettes.

2. Rename the deleted palette from *˜PaletteName*.**vp** to *PaletteName*.**vp**.

3. Restart Style Manager (close, then reopen).

When you delete a palette, its name is prefixed with ˜ and placed in .**vue/palettes** in your home directory.

# To change the number of colors used by HP VUE

1. Choose HP VUE Color Use in the Color dialog.

2. Select the color use.

3. Choose OK.

The change takes effect the next time you log in.
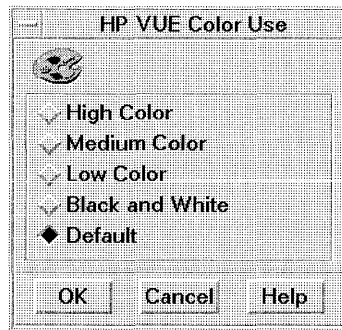


Your display type and the HP VUE Color Use selection determine the number of colors buttons that make up a palette.

While you can't use more colors than your display allows, you can limit the amount of colors used by HP VUE by specifying a lower-valued display type.

For example, if you have a high color display and want to run a color-rich application such as a CAD program, you could choose Low Color to lower the amount of colors that HP VUE uses. The color difference is available for the CAD program.

# To select a font size

1. Choose Style Manager's Font button.

2. Select a font size.

3. Choose OK.

Window labels and text show the new font size as applications are started.
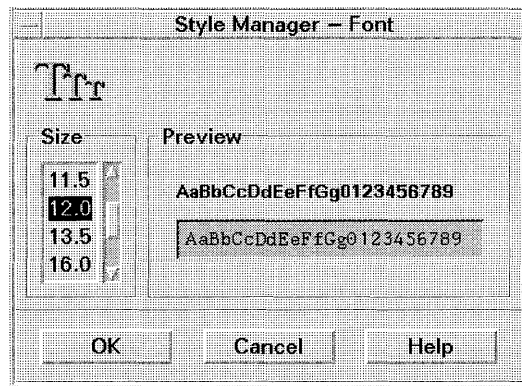
# To select a backdrop

1. Switch to the workspace in which you want the backdrop.
2. Choose Style Manager's Backdrop button.
3. Select a backdrop.
4. Choose Apply.
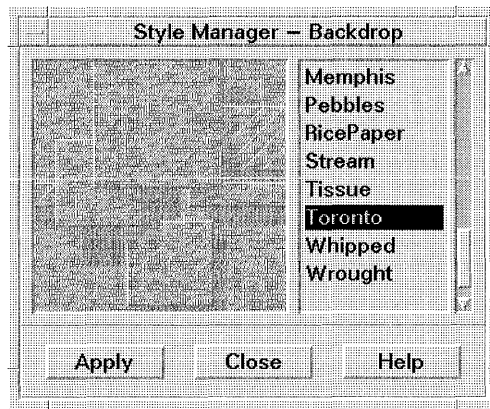


A backdrop is a pattern that covers the screen area underneath your windows. A unique backdrop for each workspace adds variety and helps you to quickly identify the workspace.

# Customizing System Behavior

Behavior settings for system devices can be easily adjusted to fit your preferences and needs:

- Key click volume and character repeat.

- Mouse button click settings, double-click speed, pointer acceleration, or pointer movement threshold.

- Audio beeper volume, tone, or duration.

- Screen time outs and cover when not in use.

- Window focus, movement, and icon behavior.

## To change keyboard behavior

1. Choose Style Manager's Keyboard button.

2. Select settings.
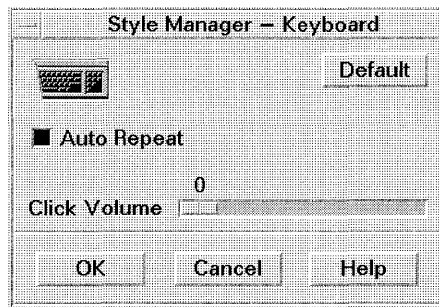
3. Choose OK.



Auto Repeat:    Have a character repeat as you hold down its key.

Click Volume:   Turn off key click (when at 0%) or change volume (up to 100%).

---

# To change mouse behavior

1. Choose Style Manager's Mouse button.

2. Select settings.

3. Choose OK.

A new double-click speed takes effect the next time you log in.



Right-Handed:   Use standard settings for mouse button clicks.

Left-Handed:    Reverse standard settings for mouse button clicks.

Double-Click:   Choose double-click time from .1 to 1 second. To test, double-click the mouse image.

Acceleration:   Choose how fast the pointer moves across the display.

Threshold:   Choose the distance (in pixels) the pointer moves at a slow speed in a short amount of time before the pointer moves at the "accelerated" rate.

## To change audio behavior

1. Choose Style Manager's Audio button.

2. Select settings.

3. Choose OK.
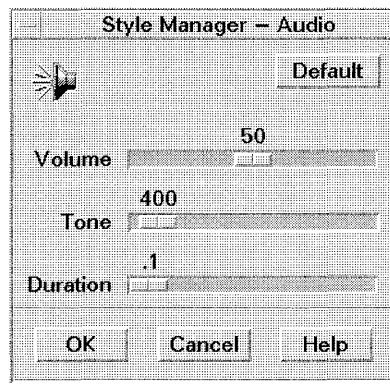


Volume:   Turn off beeper sound (at 0%) or change volume (up to 100%).

Tone:   Choose a beeper pitch from 82 to 9000 Hertz.

Duration:   Choose how long the beeper sounds from .1 to 2.5 seconds.

# To change screen settings

1. Choose Style Manager's Screen button.

2. Select settings.

3. Choose OK.
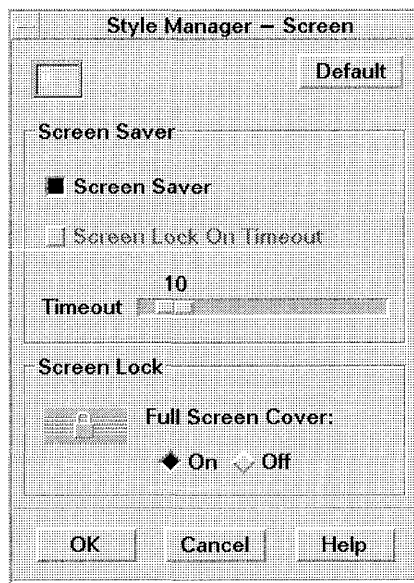


| | |
|---|---|
| Screen Saver: | Turns screen saver on or off. |
| Screen Lock On Timeout: | If screen saver is on and your server supports this feature, also locks your screen. |
| Timeout: | Specifies the time interval (from 1 to 120 minutes) from the last pointer move or keystroke to when the screen times out. |
| Full Screen Cover: | Choose whether or not to cover your screen when Lock is used from the Front Panel. |

If you have a color display, bright colors on the screen can "burn" into the picture tube. A **time out** - when the screen goes blank or is covered - prevents this from happening.

# To change window behavior

1. Choose Style Manager's Window button.
2. Select settings.
3. Choose OK.
4. Choose OK when asked to Restart the Workspace Manager.
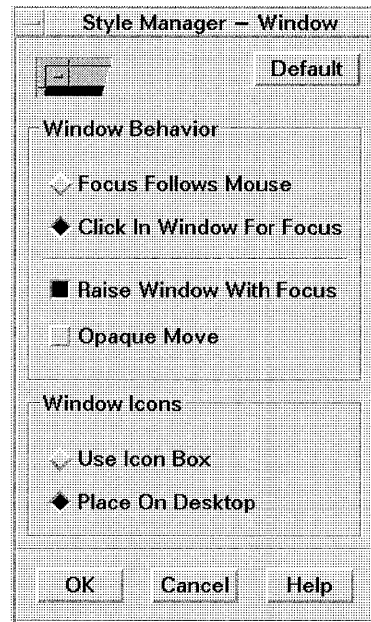
Focus follows Mouse:        Makes a window active when the mouse pointer
                            enters it.

Click In Window For Focus:  Makes the window active when the mouse
                            pointer enters the window and you click mouse
                            button 1.

Raise Window With Focus:    Brings a partially concealed window to the top
                            when it becomes active.

Opaque Move:                Moves the window during a window move
                            (instead of moving its outline first).

Use Icon Box:               Displays an icon box for window icons.

Place On Desktop:           Places window icons on the desktop.

# To change how your next session starts

1. Choose Style Manager's Startup button.

2. Select settings.

3. Choose OK.

Whenever you are logged into HP VUE, you are working in a current session.
By default, when you log out, HP VUE saves your current session and
restores it the next time you log in.

You can also save a home session, a session you choose to be the one to
which you can always return, regardless of what you do during the current
session.

```
┌──────────────────────────────────┐
│    Style Manager — Startup        │
├──────────────────────────────────┤
│ ▣▭▭                              │
│ At Login:                         │
│   ◆ Resume current session        │
│   ◇ Return to Home session        │
│   ◇ Ask me at Logout              │
│                                   │
│ Logout Confirmation Dialog:       │
│   ◆ On  ◇ Off                     │
│                                   │
│   Set Home Session ...            │
│                                   │
│    OK     Cancel     Help         │
└──────────────────────────────────┘
```
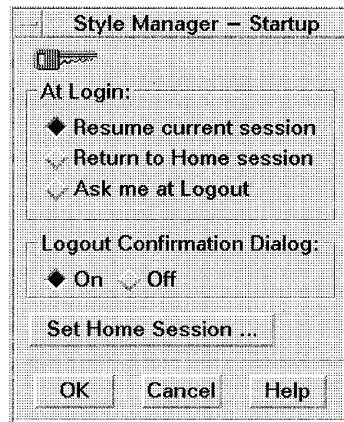
Resume current session:   Starts your next session the way you left your last one, including any setting, client, or resource modifications.

Return to Home session:   Starts the session that you've designated as your Home Session (see Set Home Session). If you have never set a home session, the system default session starts.

Ask me at Logout:   Asks you at logout whether to restart the current session or to restore the home session.

Logout Confirmation Dialog:   When On, asks you to confirm that you want to log out.

Set Home Session:   Defines a new Home Session based on your current state.

**NOTE**

When setting a home session, be sure to first select and save your desired startup settings:

1. Select the desired At Login settings in the Startup dialog.

2. Choose OK.

3. Reopen the Startup dialog, then choose Set Home Session.

# 10

Annotating Files

# Annotating Files

An annotated file is a file that has additional data attached to it. The file can be of any type, and the data can be text or audio.

File annotations are created and accessed using File Annotator, which provides the ability to:

- Review, create, edit, and delete annotations.

- Turn a file into an annotation or an annotation into a file.

To annotate a file, you must have write permission for the directory containing the file. An annotated file is underlined in File Manager.

Your system must be configured for audio before you can record or hear audio annotations.

**See Also**

- "Configuring Audio Devices (HP-UX only)" in Chapter 13 gives audio configuring information if you have audio capability.

# Reviewing Annotations Attached to a File

The File Annotator window lists all annotations attached to a file and shows information about each annotation:

- The type of annotation: text or audio.
- The person who created the annotation.
- The date and time the annotation was created or last modified.

You can review a single annotation or cue up all annotations to be read or played back.

## To see a list of annotations attached to a file

1. In a File Manager window, select the annotated file.
2. Choose Annotate from the File menu to open File Annotator.
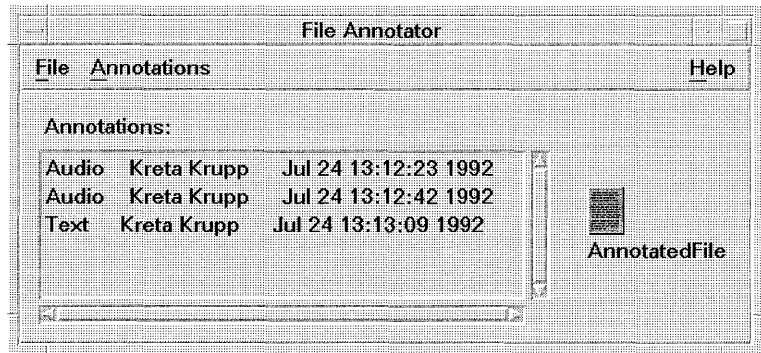
# To review a single annotation (text and audio)

**Text**

1. Open File Manager and select the annotated file.

2. Choose Annotate from the File menu.

3. Double-click the annotation in the list to open a dialog displaying the message.
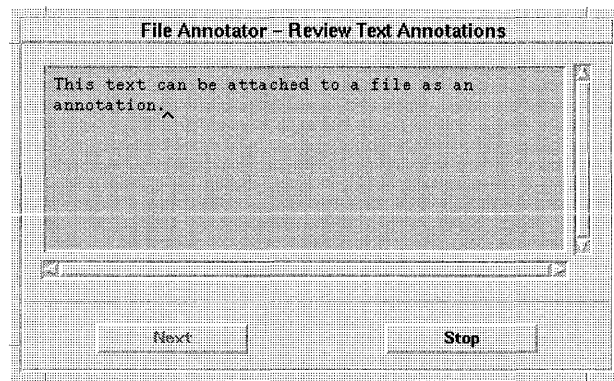


**Audio**

1. Open File Manager and select the annotated file.

2. Choose Annotate from the File menu.

3. Double-click the annotation in the list.
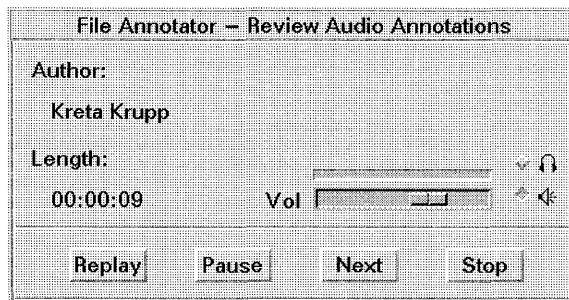
The message plays back as soon as the dialog opens. Use the audio buttons much like you use a tape recorder's buttons.

- To raise or lower the volume, adjust the Volume control.

- To temporarily suspend playback, choose Pause. Choose Pause again to continue playback.

- To repeat the message, choose Replay.

- To choose a new listening device, select headphones or speakers. The new device takes effect the next time an annotation is played.

- To stop playback and close the dialog, choose Stop.

## To review all annotations attached to a file

1. Open File Manager and select the annotated file.

2. Choose Review Annotations from the File menu. A Review Annotations dialog appears for each type (text or audio) of annotation attached to the file.

3. Choose Next to review the next annotation.

When all annotations have been reviewed, the focus moves to Stop.

During an audio review, choosing Next skips the rest of the current message and moves on to the next.
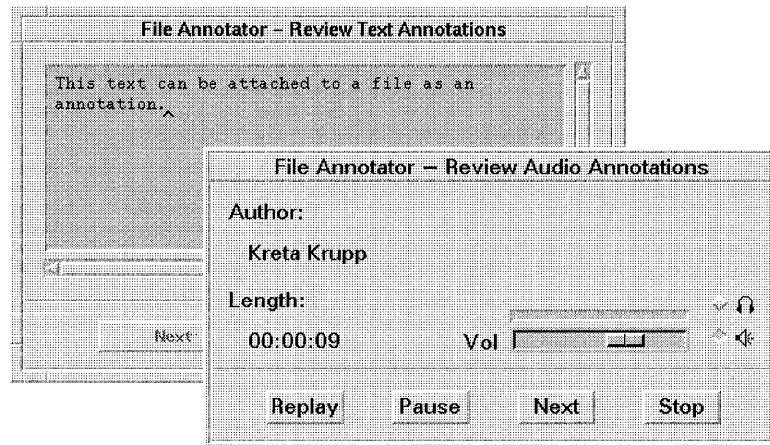
## To delete annotations

**Deleting a single annotation**

1. Open File Manager and select the annotated file.

2. Choose Annotate from the File menu.

3. Select the annotation to be deleted.

4. Choose Delete from the Annotations menu.

To delete an annotation, you must have write permission for the directory containing the annotated file.

**Deleting all annotations**

1. Select the file and choose Annotate from the File menu.

2. Choose Delete All from the Annotations menu.

The Delete menu choices are inactive if you do not have Write permission for the directory in which the annotated file resides.

# Working with Text Annotations

Text annotations, like documents, can be created, edited, and printed.

## To create a text annotation

1. Open File Manager and select the file to be annotated.
2. Choose Annotate from the File menu to open File Annotator.
3. Choose Add Text from the Annotations menu to open the Add Text Annotation dialog.
4. Type the annotation.
5. Choose OK.
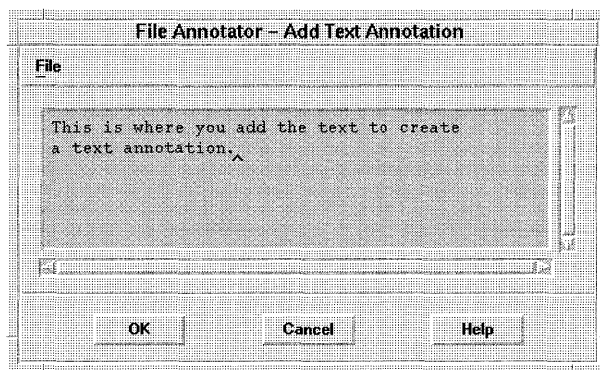
# To edit an existing text annotation

1. Open File Manager and select the file.

2. Choose Annotate from the File menu to open File Annotator.

3. Select the annotation to be edited.

4. Choose Edit from the Annotations menu to open the Edit Text Annotation dialog.

5. Edit the text.

6. Choose OK.

You must be the annotation's author to edit it.

**Including an existing file into an annotation**

1. While editing an annotation, place the cursor where you want to insert the existing file's contents.

2. Choose Include from the File menu to open a dialog.

3. Type the name of the file to be included. Use the space bar for file name completion.

4. Choose OK in the Include Text File dialog. The file's contents are inserted into the Add Text Annotation dialog.

5. Choose OK.

# To print a text annotation

1. Save the text annotation into a separate file.

2. Open the file in Text Editor.

3. Choose Print from Text Editor's File menu.

**See Also**

- "To save an annotation as a file" explains how to save an annotation as a separate file.

# Working with Audio Annotations

An audio annotation is a recorded message that is represented visually by an audio waveform. The waveform is automatically created as you record your message. Editing the waveform is similar to editing a text file; you can insert more information or append the message, delete a selected segment, and move or copy segments.

## To record a new audio annotation

1. Open File Manager and select the file.

2. Choose Annotate from the File menu to open the File Annotator.

3. Choose Add Audio from the Annotations menu to open the Add Audio Annotation dialog.

4. Choose Record and speak into the microphone.

5. Choose Stop.

6. Choose OK.

① Record.      ⑥ Zoom In.
② Stop.        ⑦ Zoom Out.
③ Play.        ⑧ Headphones.
④ Pause.       ⑨ Speaker.
⑤ Time Counter.

**Operating audio buttons**  Use the audio buttons much like you use a tape recorder's buttons.

- To raise or lower the volume, adjust the Volume control.

- To temporarily suspend recording, choose Pause. Choose Pause again to continue recording, or choose Stop to end the recording.

- To hear the complete message, choose Play.

- To append the message, choose Record.

- To erase the message you're currently recording, choose Erase from the Edit menu.

- To discard the message and close the dialog, choose Cancel.

# To select a portion of the recording to edit

**Selecting a single point**   • Point to a place on the waveform and click mouse button 1.
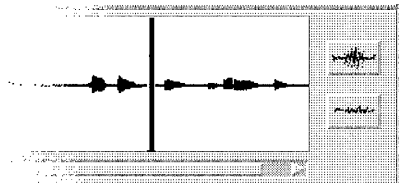


The audio pointer is a vertical line that shows your current position in the audio waveform.

Erase, Play, Paste, Record, Insert, and Save operations start at the selected point.

Selecting a point unselects any previously selected point or segment.

**Selecting a segment**   1. Point to the beginning of the segment.

2. Hold down mouse button 1 as you drag the pointer to the end of the segment.
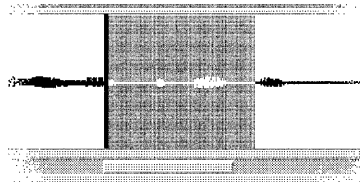


Erase, Play, Paste, Record, Cut, Copy, Insert, and Save operations affect the selected segment.

Selecting a segment unselects any previously selected point or segment. Only one segment can be selected at a time.

To lengthen or shorten the segment, drag the segment end marker using (Shift) + mouse button 1.

**Clearing the markers**

- Choose Clear Markers from the Edit menu.

Clear Markers clears both the segment and point markers.

**Zooming in**

- Choose Zoom In to see a more detailed view of the waveform.

Each time you choose Zoom In, you get a closer view of the waveform, which allows for more precise segment selection.

Choose Zoom Out to see a less detailed view of the waveform.

---

# To edit a recording

**Appending the message**

1. Choose Clear Markers from the Edit menu.
2. Choose Record and speak into the microphone.
3. Choose Stop.

**Hearing a segment**

1. Select a segment or a point from which to start playback.
2. Choose Play.

**Recording over a segment**

1. Select the segment.
2. Choose Record and speak into the microphone.
3. Choose Stop.

The selected area is replaced by the newly recorded audio data.

**Inserting a segment**
1. Select the point where the new information is to be inserted.
2. Choose Record and speak into the microphone.
3. Choose Stop.

**Erasing part of a message**
1. Select the segment to be erased.
2. Choose Erase from the Edit menu.
Only the marked segment is erased.

**Erasing to the end**
1. Select the point at which to start erasing.
2. Choose Erase from the Edit menu.

**Erasing the entire message**
1. Clear the markers.
2. Choose Erase from the Edit menu.

**Copying a segment**
1. Select the segment to be copied.
2. Choose Copy from the Edit menu.
3. Select the point where the new information is to be inserted.
   *Or*, select the segment that will be replaced.
4. Choose Paste from the Edit menu.

**Moving a segment**
1. Select the segment to be moved.
2. Choose Cut from the Edit menu.
3. Select the point where the new information is to be moved.
4. Choose Paste from the Edit menu.

**Including an existing file**
1. Select the point where the existing file will be inserted, or select the segment to be replaced.
2. Choose Include.
3. Type the name and path of the file to be included. Use the space bar for file name completion.

If no point or segment is specified, the external file is appended to the end of the current annotation.

# Using Files with Annotations

Annotations are attached to the file they annotate. There may be times when you want to:

- Save an annotation as a separate file.
- Create an annotation from the contents of a separate file.

## To save an annotation as a file

1. Open File Manager and select the file.
2. Choose Annotate from the File menu.
3. Select the annotation to be saved as a file.
4. Choose Show from the Annotations menu to open the Show Annotation dialog.
5. Choose Save from the File menu to open the Save dialog.
6. Type the name and path of the file into which the annotation will be saved. Use the space bar for file name completion.
7. Press (Enter) or choose OK.

You can save both text and audio annotations as files.

To save only a portion of an audio annotation, select the audio segment before choosing Save.

# To create an annotation from a file

**Text**

1. Open File Manager and select the file to be annotated.

2. Choose Annotate from the File menu.

3. Choose Add Text from the Annotations menu to open the Add Text Annotation dialog.

4. Choose Include from the File menu to open the Include Text File dialog.

5. Type the name and path of the file to be included and press (Enter) or choose OK. Use the space bar for file name completion. The file's contents display in the dialog.

6. Choose OK in the Add Text Annotation dialog.

**Audio**

1. Open File Manager and select the file to be annotated.

2. Choose Annotate from the File menu.

3. Choose Add Audio from the Annotations menu to open the Add Audio Annotation dialog.

4. Choose Include from the File menu to open the Include Audio File dialog.

5. Type the name and path of the file to be included and press (Enter) or choose OK. Use the space bar for file name completion.

6. Choose OK in the Add Audio Annotation dialog.

# Part 2

# Configuring
# HP VUE

# 11

Enabling and Disabling HP VUE

# Enabling and Disabling HP VUE

This chapter covers how to start and halt the HP VUE Login Manager. Login Manager is the portion of HP VUE responsible for starting the X server, validating the user's login and password, and starting sessions. Once Login Manager is running, the user has access to HP VUE sessions and other login options.

> **NOTE**
>
> This chapter applies to HP-UX systems only.
>
> You must have root permission to configure a system to start or stop HP VUE.

Some systems are set up to automatically start Login Manager during the system `init` process. These systems automatically display the HP VUE Login Screen when they are rebooted.

Other systems may be initially configured to run a console. There are two ways to start HP VUE:

- You can configure the system to automatically start HP VUE when the system is rebooted. This is done by changing the default run level for the system.

- You can manually change the system run level from the console. This has no effect on what happens when you reboot your system.

Once a system is running Login Manager, HP VUE can be halted by:

- Changing the default run level.

- Manually changing the run level from the console.

# Before You Start HP VUE

If your system has one of these special configurations, you may need to edit certain Login Manager files before starting HP VUE:

- If the system console is a non-bitmap device, see "Configuring Character-Display Consoles" in Chapter 13.

- If the system is part of a diskess cluster, see "Configuring Diskless Clusters" in Chapter 13.

- If the system is an X terminal or a host for X terminals, see "Configuring X-Terminals" in Chapter 13.

- If the system has more than one display, see "Configuring Multiple-Displays" in Chapter 13.

# Starting HP VUE automatically

When HP VUE is started automatically, the HP VUE Login Manager is run when the system is booted.

To start HP VUE automatically, the system default run level must match a run level assigned to HP VUE. The default run level and run level assignments are made in **/etc/inittab**.

There are two ways to change the default system run level in **/etc/inittab**:

- If the IGNITION fileset is installed on your system, you may be able to use the script **/etc/newconfig/Ignition/configure.sh**.

- You can manually edit **/etc/inittab** in a text editor.

# To use 'configure.sh' to edit 'inittab'

**Requirements for using 'configure.sh'**

You can use **configure.sh** to edit **/etc/inittab** if your system meets these requirements:

- The console is a bitmap display.

- The **/etc/inittab** file has not been extensively customized.

If your system is not suitable for running the **configure.sh** script, you must edit **/etc/inittab** using a text editor.

**Procedure for running 'configure.sh'**

1. Log in as root.

2. Run the script by executing:

   **/etc/newconfig/Ignition/configure.sh**

   For a diskless cluster, you must run the script on every cnode that will be running HP VUE.

3. Respond appropriately to the prompts.

**What 'configure.sh' does**

The script changes the system default run level to 4.

# To edit '/etc/inittab' in a text editor

- Log in as root.

- Make a backup copy of **/etc/inittab**.

- Edit **/etc/inittab** (in a diskless cluster, **/etc/inittab** is a context-dependent file). The default run level (**initdefault**) must match a run level specified for respawning HP VUE (**vue**). The **/etc/inittab** file shipped with HP-UX 9.0 systems respawns HP VUE at run levels 3 and 4; you should avoid changing this line.

**Example**

The following portion of **/etc/inittab** illustrates how to automatically start HP VUE when the system is booted. The default run level is set to 4—a run level assigned to HP VUE.

```
init:4:initdefault:
    .
    .
    .
cons:012456:respawn:/etc/getty -h console console
vue :34:respawn:/etc/vuerc
```

# Starting HP VUE manually

If Login Manager is not started automatically when the system is booted, it can be started later from the console.

1. Log into your system console as root.

2. Examine (for example, with the **page** command) the contents of `/etc/inittab`. Look for the line that respawns HP VUE, and make note of the run level(s) used. For example, the following lines specify that HP VUE runs at run levels 3 and 4.

        vue :34:respawn:/etc/vuerc

3. Execute:

        /etc/telinit $n$; exit

    where $n$ is a run level assigned to HP VUE.

If `/etc/inittab` contains these lines:

        init:2:initdefault:
            .
            .
            .
        cons:012456:respawn:/etc/getty -h console console
        vue :34:respawn:/etc/vuerc

then HP VUE does not start when the system is booted. However, you can start HP VUE by executing:

        /etc/telinit 3; exit        *or*        /etc/telinit 4; exit

# Stopping HP VUE

There are two basic ways to stop HP VUE:

- Change the system default run level so that HP VUE does not start when the system is rebooted. This is done by changing the default run level.

- Change system run levels manually.

## To configure the system so that HP VUE does not run at boot time

The following procedure prevents HP VUE from starting automatically when the system is booted. However, HP VUE can be started later by changing run levels.

- If you are in an HP VUE session, log out.

- Use the Options menu on the login screen to enter No Windows mode.

- Log in as root.

- Make a backup copy of **/etc/inittab**.

- Edit **/etc/inittab**. Change the default run level (**initdefault**) to a run level that runs a console but does not run HP VUE.

**Example**

The following lines show a portion of **/etc/inittab** in which the default run level is set to 2, which runs a console. Since the file specifies that HP VUE runs only at run levels 3 and 4, HP VUE does will not start automatically when the system is booted.

```
init:2:initdefault:
         .
         .
         .
cons:012456:respawn:/etc/getty -h console console
vue :34:respawn:/etc/vuerc
```

# To stop HP VUE manually

1. If you are in an HP VUE session, log out.

2. Use the Options menu on the login screen to enter No Windows Mode. Alternatively, you can log into a fail-safe session.

3. Log in as root.

4. Switch to a run level that does not run HP VUE by executing:

   `/etc/telinit` $n$`; exit`

   where $n$ is a run level for which `/etc/inittab` specifies a terminal or console and does not specify HP VUE.

**Example**

Suppose the contents of `/etc/inittab` is:

```
init:4:initdefault:
    .
    .
    .
cons:012456:respawn:/etc/getty -h console console
vue :34:respawn:/etc/vuerc
```

The following command would stop HP VUE:

```
/etc/telinit 2; exit
```

# Customizing Login Manager

# Customizing Login Manager

Login manager is responsible for:

- Reading initial configuration files. These files configure HP VUE for various types of systems, such as character-display consoles, X-terminals, and systems with multiple displays

- Starting the X server.

- Displaying the login screen and validating the login and password supplied by the user.

- Invoking the HP VUE Session Manager.

     - For information on special configurations.
  - □ "Configuring Character-Display Consoles" in Chapter 13
  - □ "Configuring X-Terminals" in Chapter 13.
  - □ "Configuring Diskless Clusters" in Chapter 13.
  - □ "Configuring Multiple-Displays" in Chapter 13.
  - □ "Configuring Multiple Screens" in Chapter 13.
  - □ "Running Starbase Applications" in Chapter 13.
  - □ "Partial HP VUE Environments" in Chapter 13.
- For information on environment variables:
  - □ "Default Environment Variables" in Chapter 14
  - □ "Setting System-Wide Environment Variables" in Chapter 14
  - □ "Setting Personal Environment Variables" in Chapter 14

# Customizing the Login Screen

You can customize:

- Appearance—the logo, greeting, color, and fonts.

- Entries in the language menu.

- The transition screen—the screen displayed after you log in, before your workspace backdrops appear.

- The vuelogin(1x) manpage contains additional information about Login Manager resources.

## To change the login screen appearance

- Set the appropriate resource in **/usr/vue/config/Xresources**.

    Vuelogin*_resource_:    _value_

- Reread the Login Manager configuration files (see "To reread Login Manager configuration files").

**Logo image**

    Vuelogin*logo*bitmapFile: _path_

**Logo position**
- Percent of the logo above the login matte:

    Vuelogin*logo*verticalOffset: _percent_

- x,y position:

    Vuelogin*logo*x: _pixels_
    Vuelogin*logo*y: _pixels_

**General logo
appearance**

Vuelogin*logo*_resource_:     _value_

**Fonts**

Vuelogin*textFont:   _font_
Vuelogin*labelFont:   _font_

**Content and appearance
of greeting:**

Vuelogin*greeting.labelString:     _string_
Vuelogin*greeting.fontList:     _font_

To include the host name in the greeting, use the string %LocalHost%.

**Example**

These lines in **Xresources** set colors and fonts for the login screen.

```
Vuelogin*background:      lightgray
Vuelogin*highlightColor: #ef506f
Vuelogin*textFont:        -*-prestige-medium-r--normal-*-128-72-*
Vuelogin*labelFont:       -*-swiss 742-medium-r-normal-*-140-*-p-110-*
```

**Example**

These lines in **Xresources** specifies a bitmap and background color for the
logo:

```
Vuelogin*logo*bitmapFile:  /usr/local/lib/X11/vue/bitmaps/MYlogo.bm
Vuelogin*logo*background:  #ef506f
```

**Example**

This line in **Xresources** specifies the login screen greeting:

```
Vuelogin*greeting.labelString: This is %LocalHost%
```

**See Also**

- "To reread Login Manager configuration files" covers how to reread
  **Xresources**.
- "To use different Login Manager resources for each display" in Chapter 13
  covers using different resources for different displays.
- "To provide different resources for each node" in Chapter 13 covers setting
  Login Manager resources in a diskless cluster.
- The comments in **/usr/vue/config/Xresources** contain more
  information about login screen resources.

# To customize the language menu

1. Set the `languageList` resource in **/usr/vue/config/Xconfig**:

   **Vuelogin*languageList:** *language [language ... ]*

   where *language* is a valid value for the LANG environment variable.

2. Reread the Login Manager configuration files.

**Example**

The following line in **Xconfig** specifies the contents of the language menu:

    Vuelogin*languageList: french italian german japanese

**See Also**

- "To reread Login Manager configuration files" explains how to reread **Xconfig**.

# Customizing Login Manager Behavior

This section covers how to:

- Display a message of the day.
- Run system-wide scripts at login and logout.
- Use a different X server.

## To display a message of the day

1. Create a file containing the message of the day.

2. Log in as root.

3. In **/usr/vue/config/Xsession**, find the line that defines the variable VUEHELLO:

    VUEHELLO=$VUEDIR/vuehello

4. Modify the line to include the **-file** option, which takes as its parameter the name of the file containing the message of the day:

    VUEHELLO=$VUEDIR/vuehello -file *path*

The **vuehello** client displays the transition window after the user login and password are validated, before the session manager is run.

By default, **vuehello** displays **Starting the HP Visual User Environment** and the copyright message (**/etc/copyright**) in the transition window.

The -file option specifies files containing additional messages. Up to five files can be added; each uses a separate -file option.

**Example**

The following line adds the general message of the day and a user-specified message.

    VUEHELLO=$VUEDIR/vuehello -file /etc/motd -file $HOME/mymessage

**See Also**
- "To reread Login Manager configuration files" explains how to reread `Xsession`.

## To run system-wide scripts at login and logout

**At login**
1. Log in as root.
2. If **/usr/vue/config/Xstartup** does not exist, create it and make the file executable.
3. Edit the file to contain commands you want run.

**At logout**
1. If **/usr/vue/config/Xreset** does not exist, create it and make the file executable.
2. Edit the file to contain commands you want run.

**Xstartup** is run as root after the user has been authenticated, before the session manager is started. Usually, **Xstartup** will be a shell script.

You can use **Xreset** to undo tasks done by **Xstartup**. Usually, **Xreset** will be a shell script.

**Environment variables for 'Xstartup' and 'Xreset'**

These environment variables are set while **Xstartup** and **Xreset** are running:

| Variable | Value |
|---|---|
| DISPLAY | Display name. |
| USER | Name of user logging in or out. |
| HOME | Home directory of the user. |
| PATH | Value of the **systemPath** resource in **Xconfig**. |
| SHELL | Value of the **systemShell** resource in **Xconfig**. |
| XAUTHORITY | May be set to an authority file. |
| TZ | Set to the current time zone. |

# To run a different X server

• Edit the line in **/usr/vue/config/Xservers** that starts the server.

The change takes effect the next time the user logs in. **Xservers** contains a line for each server. The syntax for specifying a server is:

*DisplayName DisplayClass DisplayType Command*

| | |
|---|---|
| *DisplayName* | $DISPLAY. The display name that is passed to X programs by the **-display** option. The default is *hostname*:0. |
| *DisplayClass* | Used in setting display-specific resources for a particular class of displays. |
| *DisplayType* | Set to **local**. |
| *Command* | The command line to start the server (for local servers only). |

**Example**

The following entry in **Xservers** starts the server **/usr/bin/X11/X** on display **hpcvaaa:0**.

    hpcvaaa:0 Local local /usr/bin/X11/X :0

The *hostname* can be a simple host name (for example, **hpcvaaa**) or a fully qualified domain name (for example, **hpcvaaa.cv.hp.com**).

**See Also**

• "Configuring X-Terminals" in Chapter 13 covers connecting to the X server on an X terminal.

• "To reread Login Manager configuration files" explains how to reread **Xconfig**.

# General Login Manager Database Administration

Since Login Manager starts when the system is booted before a user has logged in Login Manager must log errors separately from errors written during an HP VUE session.

Login Manager configuration files must be reread in order for changes made to Login Manager confuration files to take effect. Ordinarily, this is done by logging out and back in, but Login Manager files can be reread within a session.

## To display Login Manager errors

- Read `/usr/vue/config/Xerrors`.

`Xerrors` records errors produced by:

- `vuelogin`.

- Any standard error output by `Xstartup`, `Xsession`, or `Xreset`.

In a diskless cluster, `Xerrors` is a context-dependent file.

# To reread Login Manager configuration files

- Log out. The configuration files are reread and take effect at the next login screen.
- Reread the files from within a session by executing (as root):

    `kill -HUP` *pid*

  where *pid* is the process ID of the parent vuelogin process.

**'pid' of the parent vuelogin**

- Obtain the process ID from the file **/usr/vue/config/Xpid**. In a diskless cluster, **Xpid** is a context-dependent file.
- *Or*, execute:

    `ps -ef | grep vuelogin`

  There will be two or more **vuelogin** processes. The parent **vuelogin** process is parented (PPID column) by init (1).

# Login Manager Configuration Files and Concepts

This section covers:

- Default configuration

- Login Manager files.

- How (the order in which) Login Manager accomplishes its functionality.

## The default Login Manager configuration

Login Manager configuration files are set up to run HP VUE on these configurations:

- System with a single bitmap display that is also the system console.

- Diskless clusters consisting of cnodes, each with a single bitmap display that is the system console.

**See Also**
- "Configuring Character-Display Consoles" in Chapter 13
- "Configuring X-Terminals" in Chapter 13.
- "Configuring Diskless Clusters" in Chapter 13.
- "Configuring Multiple-Displays" in Chapter 13.
- "Configuring Multiple Screens" in Chapter 13.
- "Running Starbase Applications" in Chapter 13.
- "Partial HP VUE Environments" in Chapter 13.

---

# Login Manager Files

**Executables**

The functionality of Login Manager is provided by three HP VUE clients in /usr/vue/bin.

| Client | Description |
|---|---|
| vuelogin | Performs configuration tasks and spawns another **vuelogin** process for each display in the system. Thus, there will be $n+1$ **vuelogin** clients running, where $n$ is the number of displays managed by the system. |
| vuegreet | Provides the functionality of the login screen. |
| vuehello | Provides a transition effect between successful login and the beginning of the HP VUE session. By default, this is a welcome message and copyright notice. |

**System-wide configuration files**

These files are located in **/usr/vue/config**.

| File Name | Description |
|---|---|
| Xconfig | Contains the resources for the behavior of **vuelogin**. It can also define new locations for the other **vuelogin** configuration files. |
| Xservers | Contains a list of servers to be run by **vuelogin**. |
| Xresources | Contains resources for the appearance of the login screen. |
| Xstartup | Program (usually a shell script) that is run as root after the user login and password is validated. |
| Xsession | Shell script that sets up the user environment variables, runs **vuehello**, and invokes the session manager. |
| Xreset | Shell script that is run as root upon termination of an HP VUE session. It can be used to return the system to its pre-**Xstartup** configuration. |

| | |
|---|---|
| **User-specific configuration files** | These files are located in the user's home directory. |

File Name | Description
---|---
`.Xauthority` | Contains authorization information needed by clients that require an authorization mechanism to connect to the server.
`.vueprofile` | Contains environment variables.

**See Also**
- The `vuelogin` man page.

---

## How Login Manager starts an HP VUE session

1. The Login Manager executable (`vuelogin`) is started by the init process during the system boot sequence.

2. `vuelogin` reads `/usr/vue/config/Xconfig`. `Xconfig` contains resources for various login manager actions.

3. `vuelogin` reads the following files in `/usr/vue/config`:

   - `Xservers`, or the file identified by the resource `Vuelogin.servers` in the `Xconfig` file.

   - `Xresources`, or the file identified by the `Vuelogin*resources` resource in the `Xconfig` file.

4. The parent `vuelogin` starts the X server for each local display.

5. The parent `vuelogin` starts another `vuelogin` process for each managed display.

The following steps happen for each `vuelogin` started by the parent `vuelogin`:

1. `vuelogin` invokes `vuegreet`, which displays the login screen and handles the user's interaction with the login screen.

2. Once the user login and password have been validated, `vuelogin` runs `/usr/vue/config/Xstartup` if it is present.

3. `vuelogin` sets certain environment variables to default values.

4. **vuelogin** then runs **/usr/vue/config/Xsession**, which performs several operations:

   - It runs **vuehello**, which displays the message of the day or some other transition effect.

   - It reads **$HOME/.vueprofile**, which may contain additional environment variables or variables that override those set automatically by **vuelogin**.

5. **vuelogin** invokes the session manager, **vuesession**.

When the user terminates a session, Login Manager runs **Xreset** if it is present.
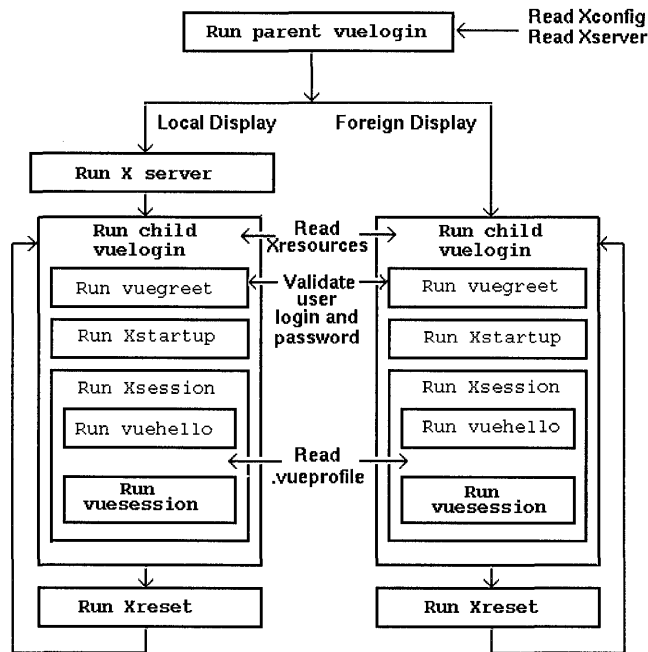


**Login Manager reads its configuration files and starts Session Manager.**

- "How Session Manager Works" in Chapter 15 covers how Session Manager starts a session.
- "Default Environment Variables" in Chapter 14

# 13

## Special Configurations

# Special Configurations

If your system includes any of these special configurations, you must customize Login Manager to accommodate them:

- Character-terminal consoles.

- X terminals.

- Diskless clusters.

- Multiple displays.

- Multiple screens.

- Starbase applications.

Other special configurations include:

- Starting an HP VUE session from an X Window System startup script.

- Running the 'vuewm' Window Manager without HP VUE.

- Moving from Softbench to HP VUE.

- Using an audio server.

# Configuring Character-Display Consoles

A character-display console is a configuration in which the console is *not* a bitmap device.

## To configure a character-display console if no bitmap display is present

1. Comment out the line in **/usr/vue/config/Xservers** that starts the X server.

   ```
   # * Local local@console /usr/bin/X11/X :0
   ```

2. Reread the Login Manager configuration files.

**See Also**

- "To reread Login Manager configuration files" in Chapter 12 explains how to reread **Xconfig**.

## To configure a character-display console if a bitmap display exists

1. Edit the line in **/usr/vue/config/Xservers** that starts the X server to read:

   ```
   * Local local@none /usr/bin/X11/X :0
   ```

2. Reread the Login Manager configuration files.

**See Also**

- "To reread Login Manager configuration files" in Chapter 12 explains how to reread **Xconfig**.

# Configuring X-Terminals

An X-terminal system consists of a display, keyboard, and mouse that runs only the X server; clients, including HP VUE, are run on one or more "host" systems on the networks. Output from the clients is directed to the X-terminal display.

Wherever possible, you should use terminals that support XDMCP.

## To configure terminals that support XDMCP

1. Make sure Login Manager is running on the host system.

2. Enable XDMCP on the X-terminal and direct it to contact Login Manager on the host system.

XDMCP (X Display Manager Control Protocol) provides a mechanism by which X-terminals can request login services from a network host. It ensures that the X-terminal is communicating with a valid login manager, and provides the protocol for exchanging authentication information between the X-terminal and the host login manager.

**See Also**
- "To limit access by X-terminals to a host".
- Documentation for your X-terminal covers the procedure for enabling XDMCP.

## To limit access by X-terminals to a host

- Edit **/usr/vue/config/Xaccess** on the host. List only those X-terminals permitted to access Login Manager.

If **Xaccess** is empty, any host can connect.

## To use a workstation as an X-terminal.

- From a command line, execute:

  `/usr/bin/X11/X -query` *hostname*

  The X server of the workstation acting as an X terminal must:

- Support XDMCP and the **-query** command-line option.

- Provide **xhost** permission (in **/etc/X\*.hosts**) to the terminal host.

## To configure non-XDMCP Terminals

1. Edit **/usr/vue/config/Xservers** to include an entry for each terminal. The *displayType* of each terminal must be **foreign**.

2. Reread the Login Manager configuration files.

When Login Manager receives a SIGHUP, it rereads **Xconfig** and the **Xservers** file (or the file specified by the **Vuelogin.servers** resource). If it finds a new entry, **vuelogin** starts managing that display. If an entry has been removed, the process associated with that entry is immediately terminated.

**Example**

The following lines in **Xservers** directs **vuelogin** to manage sessions on two non-XDMCP terminals.

```
ext1:0 NPD200X foreign
ext2:0 QCP-19 foreign
```

**See Also**

- "To reread Login Manager configuration files" in Chapter 12 explains how to reread the Login Manager configuration files.

# Configuring Diskless Clusters

In a diskless cluster, each cnode runs its own Login Manager (**vuelogin**). However, all the cnodes share these configuration files in **/usr/vue/config**:

- **Xconfig**.
- **Xstartup** and **Xreset**.
- **Xsession**.
- **Xresources**.

Special configuration for diskess clusters include:

- Starting the server on each node.
- Configuring Login Manager resources for each display.
- Configuring networking.

## To start a server on each node

- If the cnode does not match the default configuration, you must edit **/usr/vue/config/Xservers** for each cnode. **Xservers** is a context-dependent file.

The default configuration assumes each of the cnodes has a single bitmap display that is the system console.

**See Also**

- "To run a different X server" in Chapter 12 explains how to edit **Xservers**.

# To provide different resources for each node

1. On the cluster server, edit **/usr/vue/config/Xconfig** to specify a different resource file for the cnode. (There is only one **Xconfig** file per cluster.)

   > Vuelogin.*DisplayName*.resources:  *path/file*

2. Create each of the resource files specified in **Xconfig**.

3. In each file, place the **vuelogin** resources for that display.

**Example**

The following lines in **Xconfig** specify different resource files for three cnodes.

```
Vuelogin.hpcvaaa_0.resources: /usr/vue/config/Xresources.hpcvaaa
Vuelogin.hpcvbbb_0.resources: /usr/vue/config/Xresources.hpcvbbb
Vuelogin.hpcvccc_0.resources: /usr/vue/config/Xresources.hpcvccc
```

**See Also**

- "To specify the display name in 'Xconfig'" explains the syntax for display name in **Xconfig**.
- "To change the login screen appearance" in Chapter 12 covers setting resources in **Xresources**.

# Context-dependent files

These Login Manager files in **/usr/vue/config** are context dependent:

- **Xservers**

- **Xpid**

- **Xerrors**

## To configure networking for a diskless cluster

- Edit /etc/adm/inetd.sec. This file determines which systems can use the system remotely.

**See Also**

- "Configuring diskless clusters for remote data access" in Chapter 22 and "Specifying remote data on a diskless cluster" in Chapter 22 describe configuring a diskless cluster for networking.

# Configuring Multiple-Displays

This section explains how to configure Login Manager to run on systems with two or more displays.

When a system includes multiple displays, the following configuration requirements must be met:

- A server must be started on each display.

- No Windows mode must be configured for each display.

- It may be necessary or desirable to invoke different **vuelogin** resources for each display.

- It may be necessary or desirable to use different system-wide environment variables for each display.

## To start the server on each display

- Edit **/usr/vue/config/Xservers** to start an X server on each display.

**Syntax**

The general syntax for starting the server is:

   *DisplayName DisplayClass DisplayType*[ *@ite*] *Command*

Only displays with an associated ITE can operate in No Windows Mode. No Windows mode temporarily disables HP VUE for the display and runs a *getty* process if one is not already started. This allows the user to log in and perform tasks not possible under HP VUE. When the user logs out, HP VUE is restarted for the display. If a *getty* is not already running on a display, Login Manager starts one when No Windows Mode is invoked.

**Default configuration**

When *ite* is omitted, display :0 is associated with the ITE (**/dev/console**).

**To specify a different display as ITE**

- On the ITE display, set *ite* to the character device.
- On all other displays, set *ite* to **none**.

**Example**

The following entries in **Xservers** start a server on three local displays on
**hpcvaaa:0**. Display **:0** will be the console (ITE).

```
hpcvaaa:0 Local local /usr/bin/X11/X :0
hpcvaaa:1 Local local /usr/bin/X11/X :1
hpcvaaa:2 Local local /usr/bin/X11/X :2
```

**Example**

On host **hpcvbbb**, the bitmap display **:0** is *not* the ITE; the ITE is associated
with device **/dev/ttyi1**. The following entries in **Xservers** start servers on
the two bitmap displays with No Windows Mode enabled on **:1**.

```
hpcvaaa:0 Local local@none /usr/bin/X11/X :0
hpcvaaa:1 Local local@ttyi1 /usr/bin/X11/X :1
```

**See Also**

- "To run a different X server" in Chapter 12 covers the syntax for starting a
  server in **Xservers**.

# To specify the display name in 'Xconfig'

You cannot use regular *hostname*:0 syntax for the display name in
**/usr/vue/config/Xconfig**.

- Use underscore in place of the colon.
- In a fully-qualified host name, use underscores in place of the periods.

**Example**

```
Vuelogin.claaa_0.resource: value
Vuelogin.hpcvaaa_prsm_ld_edu_0.resource: value
```

# To use different Login Manager resources for each display

1. Use the **resources** resource in **/usr/vue/config/Xconfig** to specify a different resource file for each display (this file will be the equivalent to **/usr/vue/config/Xresources**):

   **Vuelogin.**_DisplayName_**.resources:** _path/file_

2. Create each of the resource files specified in **Xconfig**.

3. In each file, place the **vuelogin** resources for that display.

**Example**

The following lines in **Xconfig** specify different resource files for three displays.

```
Vuelogin.hpcvaaa_0.resources: /usr/vue/config/Xresources0
Vuelogin.hpcvaaa_1.resources: /usr/vue/config/Xresources1
Vuelogin.hpcvaaa_2.resources: /usr/vue/config/Xresources2
```

**See Also**

- "To change the login screen appearance" in Chapter 12 covers setting resources in **Xresources**.
- "To specify the display name in 'Xconfig'" explains the syntax for display name in **Xconfig**.

# To run a different startup script for each display

- Use the `startup` resource in `/usr/vue/config/Xconfig` to specify a different startup script for each display (this file will be the equivalent of the `Xstartup` file):

    `Vuelogin*`*DisplayName*`.startup:` */path/file*

The startup script is run as root after the user has logged in, before the HP VUE session is started.

The script `/usr/vue/config/Xreset` can be used to reverse the setting made in `Xstartup`. `Xreset` runs when the user logs out.

**Example**

The following lines in `Xconfig` specify different startup scripts for three displays.

```
Vuelogin.hpcvaaa_0.startup: /usr/vue/config/Xstartup0
Vuelogin.hpcvaaa_1.startup: /usr/vue/config/Xstartup1
Vuelogin.hpcvaaa_2.startup: /usr/vue/config/Xstartup2
```

**See Also**

- "To specify the display name in 'Xconfig'" explains the syntax for display name in `Xconfig`.

# To set different system-wide environment variables for each display

- Set the **environment** resource in **/usr/vue/config/Xconfig** separately for each display:

    **Vuelogin**\**DisplayName*\***environment:** *value* ...

    Separate variable assignments with a space or tab.

    Do *not* use the environment resource to set TZ and LANG.

    There is no shell processing within **Xconfig**.

**Example**

The following lines in **Xconfig** set variables for two displays.

```
Vuelogin*hpcvhere_0*environment: EDITOR=vi \
                                SB_DISPLAY_ADDR=0xB00000
Vuelogin*hpcvhere_1*environment: EDITOR=emacs \
                                SB_DISPLAY_ADDR=0xB00000
```

**See Also**

- For information about environment variables:
  □ "Setting System-Wide Environment Variables" in Chapter 14 provides more information about system-wide variables.
  □ "Setting Personal Environment Variables" in Chapter 14
- "To specify the display name in 'Xconfig'" explains the syntax for display name in **Xconfig**.

# Configuring Multiple Screens

Both the Workspace Manager and Session Manager support multiple screens, with certain limitations.

Configuring systems for multiple screens is covered in *Using the X Window System*.

## How HP VUE Supports Multiple Screens

HP VUE supports multiple screens in these ways:

- The Window Manager can be started on multiple screens. screen-specific resources can be specified for Window Manager resources.

- Session Manager will save screen-related information about clients and restore clients to their proper screens at the beginning of the next session.

Multiple screens are not supported by the following portions of HP VUE:

- Only one front panel is allowed per X server (display). It will be displayed on the default screen, or on the screen specified in the command line that starts Window Manager (see "Using the Window Manager with Multiple Screens" in chapter x).

- Session Manager can run on only one screen. (However, it can save clients running on other screens.)

- The Broadcast Message Server and Command Invoker will run only on the screen running Session Manager. This affects communication between clients, and may produce unpredictable behavior. For example, clients invoked from File Manager on screen 1 may display on screen 0.

# To manage multiple screens

There are two ways to manage multiple screens.

**Using window manager resources**

1. Use the **vuewm** resource **multiScreen** to tell the Window Manager to manage multiple screens:

   `Vuewm*multiscreen:   true`

2. Use the **vuewm** resource **screenList** to specify the screen names:

   `Vuewm*screenList:` *name    name    ellipsis;*

The front panel starts on the first screen listed. The new resources take effect when the Window Manager is restarted.

**Using Session Manager resources**

- Use the Session Manager resource **wmStartupcommand** to specify a command line for starting the Window Manager. Command-line options let **vuewm** manage multiple displays.

  □ **-display** specifies the display. It has the syntax:

  `-display` *hostname:display.screen*

  The front panel will be displayed on the screen specified by the *screen* parameter.

  □ **-multiscreen** causes the Window Manager to manage all the screens on the specified display.

  □ **-screens** specifies the screen names used to obtain screen-specific resources.

**Example**

The following resources tell the Window Manager to manager two screens:

```
Vuewm*multiscreen:   true
Vuewm*screenList: screen_0   screen_1
```

**Example**          The resource:

```
Vuesession*wmStartupcommand: vuewm -display local:0.1 \
                               -multiscreen \
                               -screens zero one
```

causes **vuewm** to manage all the screens on display 0. Screens 0 and 1 are named **zero** and **one**. The workspace manager is displayed on screen 1 (named **one**).

# Running Starbase Applications

If you plan to run Starbase applications in HP VUE, you need to:

- Allocate for Starbase applications.

- Set certain environment variables *before* the X server is started by Login Manager.

## To allocate colors to Starbase applications

1. Edit **/usr/vue/config/Xsession** to include a line that runs **xinitcolormap**. Use the **-f** option to specify the file containing the colors.

2. Create the file containing the colors to be allocated to Starbase.

Starbase applications can create their own colormap. However, if a Starbase application uses the default colormap, then colors must be allocated to it before an HP VUE session is run. The colors allocated to Starbase are not available to HP VUE.

Starbase applications do not communicate with the HP VUE color server.

# To set environment variables for starbase applications

- Use Login Manager's **environment** resource to set SB_DISP_ADDR and
  WMSHMSPC in **/usr/vue/config/Xconfig**.

  Vuelogin*environment:   SB_DISPLAY_ADDR=*value* \
                                         WMSHMSPC=*value*

SB_DISPLAY_ADDR and WMSHMSPC must be set before the X server is
started. Meaningful defaults are provided for these variables.

**Example**

The following entry in **Xconfig** set the SB_DISPLAY_ADDR and WMSHMSPC
variables.

Vuelogin*environment:   SB_DISPLAY_ADDR=0xB00000 \
                                       WMSHMSPC=0x200000

# Partial HP VUE Environments

Ordinarily, the HP VUE Session Manager is invoked automatically by Login Manager. Session Manager, in turn, automatically starts the Window Manager and saved applications. It may be preferable in certain configurations to have manual control over sessions.

## To run an HP VUE session from an X Startup Script

1. Place the following line in the X Window System startup script:

   /usr/vue/bin/vuesession -norestore

   You must use the full path to **vuesession**, and it must be run in the foreground.

2. Remove the line that starts a window manager from the startup script (**vuesession** automatically starts **vuewm**).

3. Edit the script so that the only client started before **vuesession** is **xrdb**. You *must* run **xrdb** before **vuesession** if **vuesession** requires resources—for example, if you are using the **wmStartupCommand** resource to start a different window manager.

When the session manager is started this way:

- The HP VUE Login Manager is not run.

- Session Manager does not provide the ability to save and restore sessions. It does not run a **vue.session** file; the only clients that are started are the workspace manager and clients specified in the startup script. Resources are note loaded from **vue.resources**; clients get their resources from the file loaded with **xrdb** in the startup script, or from system **app-defaults** files.

- Session Manager runs **vuewm** workspace manager. To run a different window manager:

  □ Use the resource **wmStartupCommand** to specify a different window manager.

  □ Use **xrdb** to read in this resource prior to running **vuesession**.

- Session Manager runs the Broadcast Message Server.

- The user cannot log out from the front panel logout control. To stop the server and return to a system console, press (Shift) (Ctrl) (Reset) ((Shift) (Ctrl) (Pause) for HP C1429A Enhanced Vectra keyboards).

- Fonts may not be properly set for the HP VUE clients.

**See Also**
- "To integrate X11 resource files into HP VUE" in Chapter 28.

# To run 'vuewm' without HP VUE

- Set the following resource before running the Window Manager:

      Vuewm*useMessaging:   False

This allows **vuewm** to run without the HP VUE Broadcast Message Server.

# Moving to HP VUE from Softbench

HP VUE and Softbench are both built on the same Tool Integration Platform, which includes:

- The Broadcast Message Server (BMS).

- Sub-Process Control Daemon (SPCD).

The Tool Integration Platform includes these configuration files in `/usr/softbench/config`.

`softenv`        Environment variables settings for processes started through the SPCD.

`softinit`       Specifications used by the BMS for starting messaging tools.

`softtypes/C`    Filetype specifications used by the BMS with Softbench. HP VUE does not use this file.

# Installing HP VUE over Softbench

**Files saved during installation**

When the BMS fileset is installed, any existing Tool Integration Platform configuration files are saved in `/usr/softbench/oldconfig/`*release*`/usr/softbench/config`.

If you customized any of the configuration files, you may want to compare your old version of the file to the new one and make any desired modifications.

**Softbench 1.0**

To create a `softinit` file containing entries for both products:

- Execute the following command as super-user:

```
cat /usr/softbench/config/softinitsrc/class-defaults/vue.softinit \
/usr/softbench/oldconfig/release/usr/softbench/config/softinit \
> /usr/softbench/config/softinit
```

- If you reinstall HP VUE 2.01 after executing this command, the merged `softinit` file is saved in **/usr/softbench/oldconfig**, and you can merely restore it. If you reinstall Softbench 1.0 after having performed this step, you will need to rexecute the command to merge the files.

**Softbench 1.1**

HP VUE and Softbench 1.1 can be installed on the same system in either order without any problems. Their **softinit** files are combined.

**Removing Softbench 1.0**

If you use the **sysrm** command to remove Softbench 1.0, you will also remove the Tool Integration Platform files required by HP VUE. To recover from this situation you must reinstall the BMS fileset from the HP VUE distribution.

# To run HP VUE if you've customized Softbench

1. Rename **$HOME/.softinit** file to some other name.

2. Copy **/usr/softbench/config/softinit** to **$HOME/.softinit**

3. Add any modifications you had previously made to **$HOME/.softinit** to your new copy.

If you have created a **$HOME/.softinit** file, you must modify this file using the following steps before running HP VUE.

Any time you modify your **$HOME/.softinit** file, you must restart both HP VUE and Softbench to make use of your modifications. The best way to accomplish this is to simply log out and restart your HP VUE session.

**See Also**

- The *Installing HP Softbench* manual explains how to customize Softbench.

# To quit Softbench

- To stop Softbench applications, select the application in Tool Manager window and choose Stop from the Tool menu.

# Configuring Audio Devices (HP-UX only)

Your system must meet the following configuration requirements to record and play back audio.

- The Audio Application Program Interface (AAPI) must be installed.
- The NCS Local Location Broker Daemon (**llbd**) must be running on every system that is involved with the audio operation.
- The audio server must be running.
- The TCP Daemon must be running.

The AUDIO environment variable can be set to specify the system running the audio server. If it is not set, it defaults to the local audio server.

**See Also**

- The *Audio Users Guide* contains additional information about configuring and using audio.

## To start the NCS Local Location Broker Daemon at boot time

- Edit **/etc/netncsrc**. Change the line:

      START_LLBD=0

  to

      START_LLBD=1

# To start the audio server

**Manually**

- Execute:

  `/usr/audio/bin/Aserver`

**At boot time**

- As superuser, create an **/etc/audiorc** file containing these lines:

```
#!/bin/sh
if [ -x /usr/audio/bin/Aserver ]
then
        echo "Starting Aserver"
        /usr/audio/bin/Aserver
fi
```

- Add the following section to the `localrc` procedure in **/etc/rc**:

```
# start audio server

localrc()
{
    # This function is intended for adding local...
    .
    .
    .
    #     /usr/bin/slp -i0 > /dev/lp

    # start audio server
    if [ -x /etc/audiorc ]
    then
            /etc/audiorc
    fi
}
```

- Set the permissions, owner and user on these files as follows:

```
/usr/audio/bin/Aserver      -r-xr-xr-x    root    sys
/etc/rc                     -r-xr--r-x    root    sys
/etc/audiorc                -r-xr--r-x    root    sys
```

**To see if the audio server is running**

Execute:

```
ps -e | grep Aserver
```

One server (two processes) can handle multiple clients, but it is not desirable to have more than one server running on a node.

# 14

# Environment Variables

# Environment Variables

HP VUE acquires environment variables from several sources.

- Certain default variables are built into Login Manager.

- System-wide environment variables can be set in Login Manager configuration files.

- Special resouces exist for setting time zone (TZ) and language (LANG).

- Personal environment variables are set in the the user's `.vueprofile` file.

- The `.vueprofile` file can source in the user's shell environment file. There are special requirements for doing this (see "To use a shell environment file with '.vueprofile'").

**Precedence of Variable Definitions**    Since environment variables can be defined in a number of files, it is necessary to understand which definition takes precedence. A later definition overrides a previous one. For example, a definition in `.vueprofile` overrides a setting in `/usr/vue/config/Xsession`.

```
┌──────────────────────────────────┐
│        Built into vuelogin        │
└──────────────────────────────────┘
                 │
                 ↓
┌──────────────────────────────────┐   System–wide variables
│ Resource in /usr/vue/config/Xconfig │   available to the X server
└──────────────────────────────────┘   and HP VUE session.
                 │
                 ↓
┌──────────────────────────────────┐   System–wide variables
│     /usr/vue/config/Xsession      │   available to the
└──────────────────────────────────┘   HP VUE session.
                 │
                 ↓
┌──────────────────────────────────┐
│        $HOME/.vueprofile          │   Personal variables
└──────────────────────────────────┘
```
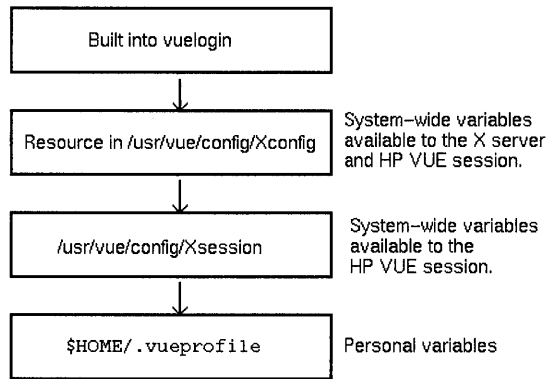
**Personal variables have precedence over system-wide variables.**

# Default Environment Variables

Default environment variables are either:

• Built into Login Manager.

• Set in the script **/usr/vue/config/Xsession**.

## Variables built into Login Manager

These variables are set separately for each display after the user login and password have been validated.

| Variable | Default Value |
|---|---|
| DISPLAY | The first field in the **Xservers** file. |
| USER | User name. |
| HOME | Home directory specified in **/etc/passwd**. |
| PATH | Value of the resource **userPath** in the **/usr/vue/config/Xconfig** file. |
| SHELL | The shell specified in **/etc/passwd**. |
| LANG | The display's current NLS language, if any. |
| LC_ALL | Set to the value of LANG. |
| LC_MESSAGES | Set to the value of LANG. |
| LOGNAME | User name. |
| TZ | Value of the **timeZone** resource in **Xconfig** (or in the operating system file that sets the time zone). |
| XAUTHORITY | May be set to an authority file. |

# Variables set by 'Xsession'

The script **/usr/vue/config/Xsession** is run by Login Manager after the X server has been started. Therefore, these variables are *not* available to the X server.

| Variable | Default Value |
|----------|---------------|
| EDITOR | **/usr/vue/bin/vuepad** |
| KBD_LANG | Set to the value of LANG for appropriate languages. |
| TERM | **xterm** |
| MAIL | **/usr/mail/**$USER |

The **Xsession** script also includes a section for setting the XVHHELPSYSTEMSEARCHPATH environment variable, which determines which online help is found. By default, this variable is commented (not set). However, if you are setting up a "help server," you may need to edit **Xsession**. (See also "To configure access to a help server" in Chapter 4.)

# Setting System-Wide Environment Variables

System-wide variables are available to all users on a system. Variables can be set so that they are:

- Available both to the HP VUE session and to the X server.
- Available only to the HP VUE session.

In addition, in multi-display systems, the scope of the variable can be set so that it is available to:

- All displays in a system.
- A particular display.

## To set system-wide variables for X server and session

- Set the `environment` resource in `/usr/vue/config/Xconfig`:

      Vuelogin*environment: *variable=value* ...

Separate variable assignments with a space or tab.

Do *not* use the `environment` resource to set TZ and LANG.

If the X server does not use the variable, it will be applied only to the HP VUE session.

There is no shell processing within `Xconfig`. Thus, you cannot use shell programming constructs such as variable assignments (for example, *VARIABLE=value*) and conditional expressions (for example, `if` ... `fi`).

**Multiple-display systems with different environments**

For multiple-display systems, where different environments are needed for the displays, use this syntax for the `environment` resource in `/usr/vue/config/Xconfig`

      Vuelogin*display*environment: *variable=value* ...

If *display* includes a colon (:), replace it with an underscore character—for example, hpcvhere_0.

**Example**

The following line in Xconfig sets the EDITOR variable for all displays.

```
Vuelogin*environment: EDITOR=vi
```

**Example**

The following line in Xconfig sets the SB_DISPLAY_ADDR variable for display hpcvxhere:1.

```
Vuelogin*hpcvhere_1*environment: EDITOR=vi \
                                  SB_DISPLAY_ADDR=0xB00000
```

# To set system-wide variables for the session only

**Single-display systems**

- Edit /usr/vue/config/Xsession to include variable definitions using the syntax:

  *variable=value*
  ⋮
  **export** *variable* [*variable* ... ]

Xsession is a shell (ksh) script, so shell processing is available.

**Multiple-display systems with different environments**

If the system has multiple displays, and different session variables are needed for the displays, use the **session** resource in /usr/vue/config/Xconfig to specify a different "Xsession" file for each display:

Vuelogin*display*session: /path/filename

If *display* includes a colon (:), replace it with an underscore character—for example, hpcvhere_0.

- Create the specified files.

- Edit the files to include the variable definitions.

**Example**

The following lines in **Xsession** set values for the MAIL and TERM variables and then export the variables.

```
MAIL=/usr/mail/$USER
TERM=xterm
export MAIL TERM
```

**Example**

The following line in **Xconfig** specifies the Xsession file for display **hpcvhere:1**:

```
Vuelogin*hpcvhere_1.session: /usr/lib/X11/vue/Vuelogin/Xsession.hpcvhere
```

# To set a system-wide timezone (TZ)

- Set the **timeZone** resource in **/usr/vue/config/Xconfig**:

    **Vuelogin[*display]*timeZone:** *value*

    If *display* includes a colon (:), replace it with an underscore character—for example, **hpcvhere_0**.

    The time zone will apply to all users logging in using HP VUE, and will override any time zone value set by the operating system.

**Example**

The following line in **Xconfig** sets the time zone for all displays:

```
Vuelogin*timeZone: PST8PDT
```

You can set a personal value for TZ in **.vueprofile**.

# To set a system-wide language (LANG)

- Set the `language` resource in `/usr/vue/config/Xconfig`:

    `Vuelogin[*`*display*`]*language:` *value*

    If *display* includes a colon ( : ), replace it with an underscore character—for example, `hpcvhere_0`.

**Example**

The following line in `Xconfig` sets the value of LANG to swedish for display hpcvxpae:0:

    `Vuelogin*hpcvxpae_0*language: swedish`

# The DISPLAY environment variable

The DISPLAY environment variable sets the host, display number, and screen number to which a system sends bitmapped output for clients.

**Default value for DISPLAY**

The default value of `DISPLAY` is set automatically by `vuelogin` to *hostname*`:0`, which is display 0, screen 0 of the local display. If you must change it, edit the first field of the appropriate entry in the `Xservers` file.

**System-wide value for DISPLAY**

When specifying a display in `/usr/vue/config/Xconfig`, replace the colon ( : ) with an underscore character—for example, `hpcvhere_0`.

**Personal value for DISPLAY**

DISPLAY can be set in `.vueprofile`.

# Setting Personal Environment Variables

User-specific environment variables are set in
/*HomeDirectory*/ .vueprofile.

> **NOTE**
>
> When a user logs in to HP VUE, the .profile or .login file is not automatically read.
> Instead, .vueprofile is read.

# To set personal environment variables

- Copy the template /usr/vue/config/sys.vueprofile to
  $HOME/.vueprofile. Give the new file write permission.

The .vueprofile file should contain only variable settings; it should *not* contain shell commands that require terminal I/O or commands that run in the foreground.

If you want the file interpreted according to a shell other than the default shell, specify the shell for the file by placing #!/bin/sh, #!/bin/csh, or #!/bin/ksh at the top of the file. (HP VUE supports these three shells; it does not support other shells.)

> **NOTE**
>
> Set *only* environment variables in .vueprofile. Commands like those for terminal input or
> output will corrupt your session environment.

**Example**

The following line in `.vueprofile` sets the value for the MAILPATH environment variable:

```
MAILPATH="${MAIL}?You have new mail."
```

## To use a shell environment file with '.vueprofile'

1. Place variables and shell commands in `.vueprofile` that apply only to HP VUE.

2. Add lines to the end of `.vueprofile` to source in the shell environment file.

3. Edit the shell environment file to provide two sections:

   - One section contains commands that do not apply to HP VUE (for example, commands that require terminal I/O, or variables for which you want to replicate the HP VUE default values).

   - The other section contains variables that apply, whether or not HP VUE is run—for example, when logging into the system from a remote system.

If you have an existing shell environment file (`.profile` or `.login`), this procedure lets you continue to use that file. This avoids duplicate variable assigments between `.vueprofile` and the shell environment file.

If a terminal emulator is started with the `-ls` option, `.login` or `.profile` will be read automatically.

**Example (ksh)**

The following lines at the end of `.vueprofile` sources in `.profile`:

```
VUE=true; export VUE
. $HOME/.profile
unset VUE
```

The following sample `.profile` shows how to separate it into a non-HP VUE section and a section for variables that apply both to HP VUE and non-HP VUE environments.

```
if [ ! "$VUE" ]; then
      stty options
      tset options
      DISPLAY=value
      MAIL=value
      export DISPLAY MAIL
      Additional shell commands that do not apply to HP VUE
      Additional variable assignments that do not apply to HP VUE
fi
PATH=value
Assignments for common variables
```

**Example (csh)**

The following line at the end of `.vueprofile` sources in `.login`:

```
setenv VUE true
source $HOME/.login
```

The following sample `.login` shows how to separate it into a non-HP VUE section and a section for variables that apply both to HP VUE and non-HP VUE environments.

```
if ( ! "$VUE" ) then
      stty options
      tset options
      setenv DISPLAY value
      set mail = value
      Additional shell commands that do not apply to HP VUE
      Additional variable assignments that do not apply to HP VUE
endif
setenv PATH value
Assignments for common variables
```

15

Administering HP VUE
Sessions

# Administering HP VUE Sessions

Session Manager provides the ability to save and restore sessions.

**Regular HP VUE sessions**

These things, characteristic of a particular session, are saved and restored by Session Manager:

- Which applications are running.

- What application windows look like (colors, fonts, size, location, etc.).

- Other settings controlled by the X server, such as mouse behavior, audio volume, and keyboard click.

Session Manager also lets you:

- Run a customized shell script after the user logs in.

- Change the window manager.

- Change the fail-safe session.

**HP VUE Lite sessions**

HP VUE Lite provides an alternative to the full HP VUE environment. HP VUE Lite sessions are managed separately from regular sessions.

**See Also**

- "Administering HP VUE Lite Sessions" in Chapter 23 describes how to configure HP VUE lite sessions.
- "Providing Networked Sessions" in Chapter 22 covers distributing an HP VUE session among multiple systems.
- "Session Management (HP VUE 2.01 to HP VUE 3.0)" in Chapter 28 covers the differences between the HP VUE 3.0 and HP VUE 2.01 Session Managers.

# How Session Manager Works

Session Manager is responsible for loading the appropriate resources, restoring server settings, and starting the clients.

Session Manager is implemented by the client **/usr/vue/bin/vuesession**.

## Types of HP VUE sessions

Sessions are divided into these categories:

- Current sessions. A **current session** is stored at logout, providing the user with the ability to "start where he left off."

- Home sessions. The **home session** is explicitly stored by the user at some other time during a session, allowing the user to always return to some "known" session.

## Location of Session Data

**Files**          Session information is stored under the following file names.

**vue.session**     The names of active clients, their window geometries, their workspace presence status (which workspaces they are in, their state (normalized or minimized), and startup string.

**vue.resources**   The resources for the active clients (including the Workspace Manager) in the session.

**vue.settings**    Server settings and session manager settings, such as screen saver timeout, audio (bell) settings, and keyboard repeat settings.

> **NOTE**
>
> Session Manager uses special syntax when writing to **vue.session** and **vue.settings**.
> You should not edit these files.

**Directories**

The location of the session files depends on the session type.

| | |
|---|---|
| current | /*HomeDirectory*/.vue/sessions/current |
| home | /*HomeDirectory*/.vue/sessions/home |
| HP VUE lite | /*HomeDirectory*/.vue/lite |
| Back-up | /*HomeDirectory*/.vue/sessions/current.old |
| | /*HomeDirectory*/.vue/sessions/home.old |

**Back-up sessions**

When the current session is saved at logout, the old session files are saved to a **current.old** directory. Similarly, when the user saves a home session, the previous home session is saved to **home.old**. If a session directory is lost, you can copy a back-up sessions. For example, the following command, executed from the /*HomeDirectory*/.vue/sessions directory, restores a back-up session:

```
cp -r current.old  current
```

**Display-dependent sessions**

HP VUE 2.01 saved and restored display-dependent sessions. Display-dependent sessions are stored in directories in which **sessions** is replaced by the display name. For example, **/users/dm/.vue/hpkbdc:0/current** stores the display-dependent current session for user **dm** on display **hpkbdc:0**. In HP VUE 3.0, display-dependent sessions have precedence over display-independent sessions; therefore, HP VUE 3.0 will continue to use a display-dependent session if it exists.

# How Session Manager chooses the session to use

When Session Manager starts a regular session, it searches for the existence of a session directory and uses the first session it finds.

**Order of search**

| | |
|---|---|
| /*HomeDirectory*/ .vue/*display*/current | Display-dependent current session. |
| /*HomeDirectory*/ .vue/*display*/home | Display-dependent home session. |
| /*HomeDirectory*/ .vue/sessions/current | Display-independent current session. |
| /*HomeDirectory*/ .vue/sessions/home | Display-independent home session. |
| /usr/vue/config | Default session. |

**Home sessions**

There are two user interactions involving home sessions:

- When the user saves a home session using the Startup dialog box in the Style manager, the state of the session is stored in the /*HomeDirectory*/ .vue/sessions/home directory (or /*HomeDirectory*/ .vue/*display*/sessions/home for display-dependent sessions).

- If the user has selected Return to Home session in the Startup dialog box, the current directory is removed at logout.

# The First HP VUE Session

**Login**

When a user logs into HP VUE for the first time, Session Manager uses the default session files in the **/usr/vue/config**: **sys.session** and **sys.resources**.

**Logout**

When the user logs out, the state of the session is saved to the */HomeDirectory/*.**vue/sessions/current** This session is retrieved the next time the user logs in.

**Previous HP VUE 2.01 users**

The first time the user logs into HP VUE 3.0, the previous HP VUE 2.01 session is restored. At logout, the session is stored to the same display-dependent directory. You can continue to use the HP VUE 2.01 session files, which are display-dependent. However, you may find it easier to transfer your session management to display-independent directories.

**See Also**

- "Session Management (HP VUE 2.01 to HP VUE 3.0)" in Chapter 28 covers the differences between the HP VUE 2.01 and 3.0 Session Managers and how to make your sessions display-independent.
- "Administering HP VUE Lite Sessions" in Chapter 23 covers how HP VUE Lite sessions are started.

# Customizing Sessions

There are several ways to customize a session:

- Creating a customized first session and distributing it to other systems.
- Running a script.
- Starting a different window manager.
- Creating display-dependent sessions.

## To create a customized first session

1. Log in to obtain the HP VUE 3.0 default session.

2. Set up the session. Start the clients you want in your session, and set the appearance and behavior you want with Style Manager or resources.

3. Log out. This creates a directory */HomeDirectory/*.vue/**sessions** and session files in that directory containing the actual session information.

4. Copy the newly-created directory to other systems whose first session you want to customize. Since sessions are saved in a display-independent syntax, you can copy sessions to displays of different resolutions.

5. Remove any display-dependent session directories (directories named */HomeDirectory/*.vue/*displayname*) from the other systems.

# To execute additional commands at session startup

1. Create the shell script file */HomeDirectory/*.vue/sessions/sessionetc.
   (Replace **sessions** with *display* for display-dependent sessions.) Make the
   file executable.

2. Add a line for each process.

If the file **sessionetc** exists, it is executed each time the user logs into a
regular HP VUE session.

Use **sessionetc** to execute additional commands at session startup; for
example:

- To execute X commands for processes or settings that are not saved by
  Session Manager. For example, you might want to use **xsetroot** to
  customize the root (workspace) pointer.

- To start X clients (applications) that are not saved by Session Manager. (An
  application cannot be saved and restored by Session Manager if it does not
  set the WM_COMMAND property.)

---

**NOTE**

Processes started by **sessionetc** must be run in the background.

Do not use **sessionetc** to start clients that are automatically restored. Doing so causes multiple
copies of the application to be started; you may not be able to see the copies immediately because
the windows are stacked on top of one another.

---

The following line customizes the root cursor:

```
xsetroot -cursor /users/anna/point /users/anna/mask -fg blue -bg red &
```

## To use a different window manager

1. Set the wmStartupCommand resource:

    vuesession*wmStartupCommand:     *execution_string*

    where *execution_string* is the command to start a different window manager.

2. Log out and back in.

In the default configuration, the HP VUE Workspace Manager (vuewm is automatically started by Session Manager. The wmStartupCommand cancels the default behavior and executes the specified command instead.

The following line starts mwm:

    vuesession*wmStartupCommand:     /usr/bin/X11/mwm

## To create a display-dependent session

1. Log in and out of at least one HP VUE 3.0 session. This will create session files in /*HomeDirectory*/.vue/sessions.

2. Copy /*HomeDirectory*/.vue/sessions to one or more display-dependent directories /*HomeDirectory*/.vue/*display*.

Create a display-dependent directory if you want to run different sessions on different displays. The *display* directory name must be the real, unqualified host name (for example, hpabcd:0 is valid; hpabcd.cv.com:0 or local:0

are *not*). If the display name exceeds 14 characters, truncate the name
starting with the last character.

**Example**

The following command creates a display-dependent session for display
hpcvxdm:0:

>    cp -r */HomeDirectory/*.vue/sessions   */HomeDirectory/*.vue/hpcvxdm:0

The display-dependent session will be used on all subsequent sessions on
display hpcvxdm:0.

**See Also**

- "Session Management (HP VUE 2.01 to HP VUE 3.0)" in Chapter 28
  describes the differences between HP VUE 2.01 and HP VUE 3.0 sessions
  and how to create a display-independent session.

---

# To customize the fail-safe session

- Edit **/usr/vue/config/Xfailsafe**. You can insert commands between
  the line that starts **mwm** and the the line that starts the Terminal Emulator.
  The commands you insert must be run in the background.

>    $XDIR/mwm    sleep $DELAY
>    *new_command*    $XDIR/hpterm -ls

The Terminal Emulator **hpterm** is run in the foreground. The script
terminates and the fail-safe session ends when the user exits the Terminal
Emulator window.

The fail-safe session is useful for correcting problems in Login Manager and
Session Manager configuration files that prevent an HP VUE session from
starting properly.

# Restoring Remote Applications

Applications running on remote systems will be restored by Session Manager if either of the following conditions are met.

- The local and remote systems are configured properly to use actions to run remote applications. This allows Session Manager to use the SPCD to start the remote application.

- *Or*, the local host can access the application using `remsh`. For this to happen, both of these conditions must be met:

  □ The remote system lists the local system in its */HomeDirectory/*`.rhosts` file.

  □ The local system permits the remote system to use its display. This is done in the local system's `/etc/X*.hosts` file.

An application started during a session may not be restarted at the next session if:

- The `xhost` permission was acquired using the `xhost` command prior to running the application. The `xhost` permission must be supplied by the `/etc/X*.hosts` file.

- The application was started by logging into the remote system (using `telnet` or `rlogin`). The local host may not have `.rhost` permission on the remote host.

**See Also**

- "Using Actions to Run Remote Applications" in Chapter 22

# Working with Icons
# and Backdrop Images

# Working with Icons
# and Backdrop Images

Icons and backdrop images are an important part of the visual appearance of your HP VUE display. Icons are associated with actions and filetypes, Front Panel controls, and minimized application windows to make them easier to identify.

A unique **backdrop** can be displayed in each HP VUE workspace (behind all other windows). Images used for backdrops are typically larger than icons, however, the same formats are used in the image files. You select backdrops using Style Manager's Backdrop dialog.

**Introducing
Icon Editor**

The HP VUE Icon Editor is a simple tool for creating image files for use as icons or backdrops. It supports both image formats (bitmap and pixmap) used for icons and backdrops.
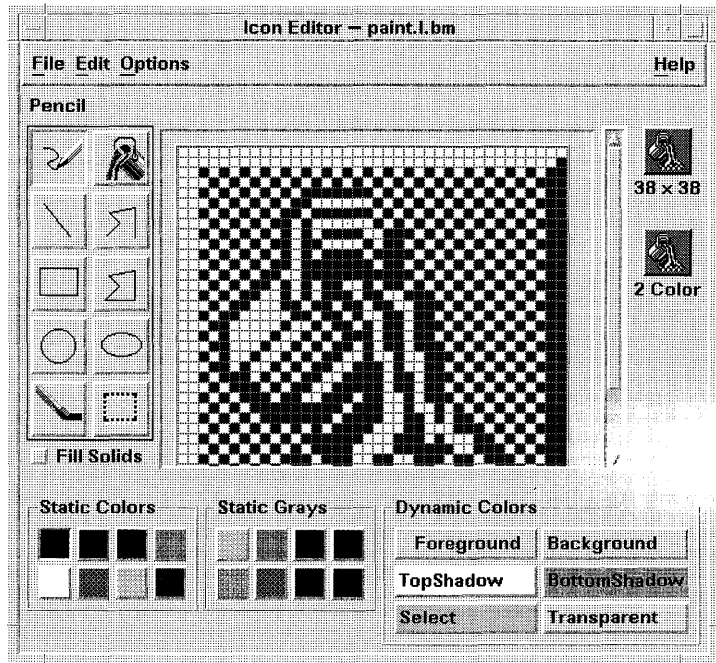


**See Also**

- "Icon Design Recommendations"

# Getting Started with Icon Editor

Icon Editor lets you create and edit images in two formats:

- *X pixmap (XPM format)*—multicolor images that include static and dynamic colors. Pixmap files are normally identified by a `.pm` file name extension.

- *X bitmap (XBM format)*—two-color images limited to a foreground and background color. Bitmap files are normally identified by a `.bm` file name extension.

You draw images by selecting a tool and a color, then drawing in the work area. As you draw, Icon Editor displays an actual-size copy of your icon in both formats. Even if you are drawing an image for use on color systems, you should consider the two-color version since icons may revert to two colors if there are not enough free colors to display in full color.

## To start Icon Editor

- Open your Personal Toolbox, then execute the IconEditor action.

Icon Editor starts with an empty drawing area.

**Other Ways to Start Icon Editor**

You can also start Icon Editor with an icon already loaded:

- Using File Manager, locate the icon file and open it. (The default action for pixmap and bitmap files runs Icon Editor and loads the file.)

- *Or*, drop an icon file on the IconEditor action in your Personal Toolbox.

- *Or*, if you are using HP VUE Lite, enter this command in a Terminal Emulator window:

        vueicon -f *filename* &

# To draw an image

1. Select a drawing tool (such as the pencil, circle, or line).

2. Select a color (static color, static gray, or a dynamic color).

3. Begin drawing with the tool you've selected.

**Summary
of Tools**

- Pencil—for drawing free-hand lines and individual pixels.
  Click to draw individual pixels.
  Press *and hold* to draw free-hand lines.
- Line—for drawing straight lines.
  Press *and hold* to stretch the line. Release to draw the line.
- Rectangle—for drawing solid or outlined rectangles.
  Press *and hold* to stretch ouline of rectangle. Release to draw the
  rectangle.
- Circle—for drawing solid or outlined circles.
  Point to center of circle, press *and hold*, drag to desired size, release.
- Eraser—for erasing large areas of the image.
  Press *and hold*, then drag over area to erase.
  To erase individual pixels, select the Pencil tool and the Transparent
  color.
- Flood—for flooding a region of one color with the selected color.
  Click any pixel within the region to be flooded. The selected pixel, and
  all adjacent pixels, are recolored with the selected color. The filled region
  extends in all directions, bounded only by pixels of different color.
- Open Polygon—for drawing connected straight lines.
  Click the starting point, then click for each segment.
  Double-click to end the last segment.
- Closed Polygon—for drawing connected straight lines, where the first line
  and last line are connected to form a closed polygon.
  Click starting point, then click for each segment.
  Double-click to end last segment.
- Ellipse—for drawing solid or outlined ellipses.
  Press *and hold*, drag to desired size and shape, release.
- Selection—for making a primary selection. Several commands in the Edit
  menu require a primary selection first.
  Point to corner pixel of the region to be selected, press *and hold*, drag to
  desired region, release.

## To undo a drawing operation

- Choose Undo from the Edit menu.

Only the single most recent change can be undone.

## To open an existing icon

1. Choose Open from the File menu.

   If you have unsaved changes, Icon Editor warns you that your changes will be lost by loading another icon. Choose Cancel to cancel the Open command. Choose OK if you want to lose your changes.

2. Enter or select the icon file you want to load.

3. Choose OK.

**NOTE**

When you load a file, the Output Format (in the Options menu) is automatically set to the format of the file you load.

# To save the icon

1. Choose an output format from the Options menu:

   - XBM for two-color X bitmap format.
   - XPM for multicolor X pixmap format (the default).

2. Choose Save from the File menu. (If the icon is unnamed or you choose Save As, Icon Editor prompts you for a file name. Enter the name then choose OK.)

**See Also**
- "Where to put icon files"
- "Where to put backdrop image files"
- "How Image Files are Found"

# To save the icon with a new name

1. Choose Save As from the File menu.

2. Enter (or select) the file name you want to save as.

3. Choose OK.

# To start a new icon

1. Choose New from the File menu.

   If you have unsaved changes, Icon Editor warns you that your changes will be lost by starting a new icon. Choose Cancel to cancel the New command. Choose OK if you want to lose your changes.

2. Enter the width and height of the new icon you want to create.

3. Choose OK.

Icon Editor clears the work area and resizes it to the size you specified. The maximum size for an icon is 256×256.

If you want to start a new icon of the current size, choose Clear Icon from the Edit menu.

## To resize the current icon

1. Choose Resize Icon from the Edit menu. Icon Editor displays a dialog, prompting for the new size.

2. Edit the current width and height to specify the new size.

3. Choose OK.

**NOTE**

If you make the icon smaller, the right and bottom edges are clipped off. If you make the icon larger, the existing image remains in the upper-left of the icon.

# Manipulating Images within Icon Editor

After you become familiar with the basic drawing facilities in Icon Editor, you may be anxious to try some more advanced operations.

Many of these operations require you to first *select* an area of the icon. This is done using the Selection tool (next to the eraser). When an area of the icon is selected, you can:

- Copy or move it.

- Scale (resize) it.

- Flip it (horizontally or vertically).

- Rotate it (right or left).

## To select an area of the icon

1. Select the Selection tool (next to the eraser).

2. Drag a box around the area you want to select.

When an area is selected, a box flashes around the selected area.

## To copy a region of the icon

1. Using the Selection tool, select the area of the icon you want to copy.

2. Choose Copy from the Edit menu.

3. Choose Paste from the Edit menu.

4. Position the outline where you want to paste the copy, then click.

You may repeat steps 3 and 4 as many times as you like to paste multiple copies of the region you copied.

---

# To cut a region of the icon

1. Using the Selection tool, select the area of the icon you want to cut.

2. Choose Cut from the Edit menu.

When you cut a portion of an icon, the exposed area becomes "transparent" (that is, it is filled in with the Transparent color).

After cutting a region, you can paste it back into the image using the Paste command in the Edit menu.

---

# To move a region of the icon

1. Using the Selection tool, select the area of the icon you want to move.

2. Choose Cut from the Edit menu.

3. Choose Paste from the Edit menu.

4. Position the outline where you want to paste the copy, then click.

You may repeat steps 4 and 5 as many times as you like to paste multiple copies of the region you cut.

## To resize (scale) a region of the icon

1. Using the Selection tool, select the area of the icon you want to resize.

2. Choose Scale from the Edit menu.

3. Drag a box that defines the size and shape that you want the scaled image to be. *This determines how the selected portion of the image will be scaled, it does not determine where to place the results.*

4. Position the outline where you want to paste the scaled copy, then click.

## To rotate part of the icon

1. Select the Selection tool (next to the eraser).

2. Drag a box around the region of the icon that you want to rotate. This "selects" the region.

3. Choose Rotate▶Left or Rotate▶Right from the Edit menu.

4. Position the outline where you want to paste the rotated area, then click.

## To flip (mirror) part of the icon

1. Using the Selection tool, select the area of the icon you want to flip.

2. Choose Flip▶Vertical or Flip▶Horizontal from the Edit menu.

Flipping an area creates a pixel-for-pixel mirror image of the selected area.

## To add a hotspot

1. Choose Add Hotspot from the Edit menu.

2. Click the pixel where you want the hotspot.

Bitmap images (XBM format) can be used as special mouse pointer shapes. The hotspot marks a single pixel that is the true "point" of the pointer.

To remove the current hotspot, choose Delete Hotspot from the Edit menu.

## To capture (grab) a region of the display

1. Save your current icon, if necessary.

2. Choose Grab Screen Image from the Edit menu.

3. Drag a box around the portion of the display you want to capture (press, drag, release).

# How Image Files are Found

Each icon and backdrop image is stored as a separate file. Typically, an icon is specified with just the base part of its file name. For example, an icon might be referenced with just the name `mail` when the file is actually stored as:

    /usr/vue/icons/Color/mail.m.pm

The file naming convention of adding suffixes helps group icons by size and type. For HP VUE components, many icon names are in these general formats:

*basename.size.format*
*basename.format*

where *basename* is the image name used to reference the image, *size* is `l` (for large), `m` (for medium), or `s` (for small), and *format* is `pm` for X pixmaps or `bm` for X bitmaps.

For example, suppose you specify an icon named "`mail`" for a filetype you've written. If you have a color display and have set the File Manager preferences to use "small" icons, the assumed icon name is "`mail.s.pm`" (the "`s`" represents small, and "`pm`" is for *pixmap*, the color icon format).

**Search Paths**   The directory where an image is stored is determined by searching for the file in a list of directories. The first match completes the search. By default, HP VUE components look in these directories:

/*HomeDirectory*/.vue/icons/*application*/Color/
/*HomeDirectory*/.vue/icons/*application*/
/*HomeDirectory*/.vue/icons/Color/
/*HomeDirectory*/.vue/icons/
/usr/vue/icons/*country*/*application*/Color/
/usr/vue/icons/*country*/*application*/
/usr/vue/icons/*country*/Color/
/usr/vue/icons/*country*/
/usr/vue/icons/Color/
/usr/vue/icons/Color/
/usr/vue/icons/

If you are using a color display, the search paths look for files ending in .pm. Otherwise, the search path looks for files ending in .bm.

If an image is specified with a complete path name, the search path is not used.

# Where to put icon files

**For Personal Use**
- If the icon is intended for color displays (XPM format), store the file as */HomeDirectory/*`.vue/icons/Color/`*filename*`.pm`.

- Otherwise, store the file as */HomeDirectory/*`.vue/icons/`*filename*`.bm`.

**For Public Use**
- If the icon is intended for color displays (XPM format), store the file as `/usr/vue/icons/Color/`*country/filename*`.pm`.

- Otherwise, store the file as `/usr/vue/icons/`*country/filename*`.bm`.

The *country/* subdirectory is optional. However, if it's used, it must match the country portion of the user's LANG environment variable.

> **NOTE**
>
> You must be superuser to place new files into the **/usr/vue/icons/** directory (or any of its subdirectories).

# Where to put backdrop image files

Each backdrop listed in Style Manager's Backdrop dialog represents an image file. By default, Style Manager looks in **/usr/vue/icons/Vuebackdrops/** and **/usr/vue/icons/Vuebackdrops/Color/** for backdrops. To override these locations, you must set the **Vuestyle*backdropDirectory** resource.

**For Personal Use**
- Create a custom backdrop directory, add the new image to the new directory, then set the **Vuestyle*backdropDirectory** resource.

**For Public Use**
- If the backdrop is intended for color displays (XPM format), store the file as **/usr/vue/icons/Vuebackdrops/Color/***filename*.**pm**.

- Otherwise, store the file as **/usr/vue/icons/Vuebackdrops/***filename*.**bm**.

**NOTE**

You must be superuser to add files to the **/usr/vue/icons/Vuebackdrops/** directory (or any of its subdirectories).

**See Also**
- "To customize backdrops" in Chapter 24 describes in more detail how to set up a personal backdrop directory.

# Making Icon Associations

By associating an icon with a particular object you make it easier to recognize. You can associate icons with:

- Actions and filetypes
- Controls in the Front Panel and subpanels
- Minimized application windows

**Specifying Icon Files**

For icons used for actions, filetypes, and in the Front Panel or subpanels, specify only the basename of the icon (no extensions). The correct extensions are added automatically based on your display resolution, color support, and File Manager preferences (such as "use small icons").

For minimized window icons, specify the entire icon name, including extensions.

For any icon, you can also provide the complete path and name of the icon to override the search path.

## To associate an icon with an action or filetype

- Specify a large icon with the L-ICON keyword.
- Specify a small icon with the S-ICON keyword.

If you follow the appropriate naming conventions for icon files, you can specify only the base name of the icon. The correct icon will be displayed based on the resolution and color support of your display.

**Example:**
**Action Icons**

The following example is an action definition for starting Island Paint (a drawing tool). The icons `Ipaint.l` and `Ipaint.s` are associated with the action.

```
ACTION            IslandPaintOpenDoc
   WINDOW-TYPE    NO-STDIO
   L-ICON         Ipaint.l
   S-ICON         Ipaint.s
   EXEC-HOST      machine2
   EXEC-STRING    /usr/bin/IslandPaint $(File)Arg_1"File to open:"%
END
```

If you are using color icons, HP VUE first appends `.pm` when looking for the actual icon files. Otherwise (or if no match is found with `.pm`), HP VUE appends `.bm`.

**Example:**
**Filetype Icons**

The following filetype definition associates the icons `comprsd.l` and `comprsd.s` with compressed files:

```
FILETYPE COMPRESSED
        FILE-PATTERN    *.Z
        MODE            !d
        L-ICON          comprsd.l
        S-ICON          comprsd.s
        ACTIONS         Uncompress
        DESCRIPTION     A COMPRESSED file has been compressed by the \
                        'compress' command to take up less space.
    END
```

**See Also**

• Chapter 19

# To display an icon in a Front Panel control

- Specify the image name using the IMAGE field.
- If the control is a `monitor_file` type control, use the ALT_IMAGE field to specify the image displayed when the file is not empty. For the `mail` control, use ALT_IMAGE to specify the image used when the file gets larger.

You can also provide animation for buttons and drop zone controls.

**Example**

The following control changes appearance when a file named `report` is placed in the `/doc/ftp/pub/` directory. When the file is not there, the `NoReport.pm` icon is displayed; when the file is there, `Report.pm` is displayed.

```
CONTROL MonitorReport
{
    TYPE              monitor_file
    MONITOR_FILE      /doc/ftp/pub/report
    IMAGE             NoReport.pm
    ALTERNATE_IMAGE   Report.pm
}
```

**See Also**

- Chapter 17 provides the details for modifying Front Panel controls.
- "To animate a button" in Chapter 17

# To associate an icon with an application window

1. Set the `clientIcon` resource for Workspace Manager as follows:

   `Vuewm*`*ClientName*`*clientIcon:`    *IconFilename*

2. Restart Workspace Manager. (Choose Restart Workspace Manager from the Workspace Menu, then choose OK.)

The HP VUE Workspace Manager and the OSF/Motif Window Manager allow only X bitmap images (XBM format) for window icons. So, *IconFilename* must refer to an X bitmap image, *not* a multicolor X pixmap.

To determine the correct value for *ClientName*, execute the ListWinProps action in the "Unsupported" directory of the General Toolbox. When you select a window, its properties are listed. The WM_CLASS property displays the window's class name in quotes.

To verify that the icon has been recognized by Workspace Manager, minimize the window whose icon you are trying to modify.

**NOTE**

Some applications do not allow their default icon to be overridden.

• Chapter 25 describes how to set X resources.

# To use File Manager as an icon browser

1. Copy the file **/usr/vue/examples/IconBrowse.vf** into your
   /*HomeDirectory*/ **.vue/types/** directory.

2. Reload the action database by executing the ReloadActions action in the
   System_Admin toolbox within the General Toolbox.

When you change to a directory that contains icons ( **.bm** and **.pm** files),
each icon is displayed next to its name. For example, if you change to the
**/usr/vue/icons/** directory, you'll see many of the HP VUE icons.

> **Notes**
>
> - When you enable icon browsing, File Manager must do a lot more work to display each directory.
>   On low memory systems this may cause delays in displaying some directories.
> - Images larger than 256×256 are not displayed in the default configuration. (You can override this
>   default limit using the `instanceIconHeight` and `instanceIconWidth` resources.)

To disable icon browsing, remove your personal copy of the `IconBrowse.vf`
file, then load the action database again.

**See Also**    • Chapter 19

# Icon Design Recommendations

Try to use a common theme among related icons. For example, if you are designing icons for an application, have purposeful similarities between the application's icon and icons for data files.

Be sure the two-color version of any color icon you design is acceptable. If the icon is displayed on a monochrome or grayscale display (or if there are not enough colors available), the icon is automatically displayed in its two-color form.

## Color Usage

Icons used in HP VUE use a palette of 22 colors:

- Eight static grays.

- Eight static colors: red, blue, green, cyan, magenta, yellow, black, and white.

- Six dynamic colors: foreground, background, top shadow, bottom shadow, select, and transparent.

This palette is rich enough to create attractive, easy-to-read icons without overtaking many color resources that may be needed by other applications. Most icons provided with HP VUE use mostly grays with color for accent.

The dynamic colors are useful for icons that change color as different color palettes are selected in Style Manager.

The transparent color is useful for creating icons that have the illusion of being non-rectangular, since it allows the color behind the icon to show through.

# Icon Size Recommendations

Following are the recommended sizes—in pixels, width×height—for creating new icons.

File Icons:

> Small: 18×22
> Large: 28×38

Directory Icons:

> Small: 22×18
> Large: 38×28

Executable Icons:

> Small: 22×22
> Large: 38×38

Minimized Application Windows:

> 50×50

Front Panel, High Resolution (1280×1024 Screens):

> Top row controls: 56×64
> Bottom row controls: 22×52

Front Panel, Medium Resolution (1024×768 Screens):

> Top row controls: 42×48
> Bottom row controls: 17×42

Front Panel, Low Resolution (640×480 Screens):

> Top row controls: 34×40
> Bottom row controls: 13×34

Backdrop images can be any size. The pattern is repeated to fill the entire workspace.

Working with Icons
and Backdrop Images
**Icon Design Recommendations**

# 17

# Customizing the Front Panel

# Customizing the Front Panel

There are a number of ways you can customize the Front Panel:

- Adding and deleting controls in the top and bottom row.

- Adding and deleting subpanels.

- Adding and deleting controls in subpanels.

You can also change the number of rows. In networked environments, you can distribute all or portions of a customized Front Panel to other users on the same system or other systems.

# Modifying the Front Panel: An Introduction

The Front Panel is defined in a set of files:

- The top and bottom rows are defined in the Workspace Manager configuration file:

  □ System file: `/usr/vue/config/sys.vuewmrc`.

  □ Personal file: `/HomeDirectory/.vue/vuewmrc`.

- Each subpanel is defined it its own file.

## To create a personal Workspace Manager configuration file ('vuewmrc')

The Workspace Manager configuration file defines the structure of the Front Panel.

**Using ListVuewmrc**

- Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox. This creates the personal Workspace Manager configuration file (`/HomeDirectory/.vue/vuewmrc`) and opens it for editing. ListVuewmrc uses the editor defined by the EDITOR environment variable.

EditVuewmrc checks to see if `/HomeDirectory/.vue/vuewmrc` exists:

- If the file does not exist, it is created by copying `/usr/vue/config/sys.vuewmrc`. The new file is given write permission as is then opened for editing.

- If the file already exists, the existing file is opened for editing.

**Manually creating the file**
- Copy `/usr/vue/config/sys.vuewmrc` to `/`*HomeDirectory*`/.vue/vuewmrc`. Give the new file write permission.

  If a personal Workspace Manager configuration files does not exist, the system file is used.

**Previous HP VUE 2.01 users**
If you used a personalize HP VUE 2.01 front panel, `/`*HomeDirectory*`/.vue/vuewmrc` will be present. You can convert this file to HP VUE 3.0 syntax or remove the file.

**See Also**
- "Moving an HP VUE 2.01 Front Panel to HP VUE 3.0" in Chapter 28 covers converting an HP VUE 2.01 Front Panel to HP VUE 3.0.

# How the Front Panel definition is organized

The Front Panel is defined as a set of constructs—panel, boxes, and controls. The Front Panel definition uses these constructs to build the panel from the outside in.

- The Front Panel, or "panel," is built from a set of boxes (rows).
- Each box is a container for a set of controls.

**Panel definitions**
There are two Front Panels defined in the Workspace Manager configuration file. To find the one you want, search for:

```
PANEL front_panel_name
{
  BOX box_name
     . . .
}
```

where *front_panel_name* is:

| Name | Description |
| --- | --- |
| FrontPanel | Front Panel for regular HP VUE sessions. |
| LiteFrontPanel | HP VUE lite front panel. |

The panel contains a set of boxes, one for each row.

**Row definitions**
Each row is a box containing a list of controls. To find the row you want, search for:

```
BOX box_name
{
    CONTROL control_name
       ...
}
```

where *box_name* is:

Top                Top row of Front Panel for regular sessions.

Bottom             Bottom row for Front Panel for regular sessions.

LiteTop            Top row of HP VUE Lite Front Panel.

LiteBottom         Bottom row of HP VUE Lite Front Panel.

**Control definitions**
Controls contain a list of *fields* that describe the appearance and behavior of the control.

```
CONTROL control_name
{
    field
       ...
}
```

The *control_name* is the same in the row (BOX) definition and the control definition. You can use the row definition as an "index" of control names to help you find the control you want to edit.

**Top-row controls have subpanels**
A subpanel is attached to a control on the top row by the control's SUBPANEL field:

```
CONTROL control_name
{
       ...
    SUBPANEL      subpanel_name
}
```

**Subpanel files are included into 'vuewmrc'**

Each subpanel, and the controls it contains, is defined in a separate file. The Workspace Manager configuration file "includes" (reads in) the subpanel files:

```
INCLUDE
{
  "/usr/vue/config/panel/fp.help"
      ...
}
```

**Subpanel definition**

Each subpanel is a box containing its controls. For example, /usr/vue/config/panels/fp.help contains the Help subpanel.

```
BOX HelpSubpanel
{
  TYPE      subpanel
  CONTROL   control_name
     ...
}
```

# Modifying the Top or Bottom Row

You can add, delete, or modify controls in the top and bottom rows. You can also rearrange the rows and move controls from one row to another.

> **NOTE**
>
> You do not need to edit controls in the Front Panel to change your default terminal, printer, editor, or mailer.

## To add or delete a control from the top or bottom row

1. Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox. This opens your personal Workspace Manager configuration file (*/HomeDirectory/*.vue/vuewmrc) for editing.

2. Locate the front panel to be edited using its name (**PANEL** *panel_name*).

3. Locate the description for the row to be edited: **BOX Top** (top row) or **BOX Bottom** (bottom row).

4. Examine the list of controls in the box:

   **CONTROL** *control_name*

   - To delete a control, remove or comment out the line for that control.

   - To add a control, insert a line for the control.

   - To replace a control, replace *control_name*.

5. If you've added or replaced a control, define the control elsewhere in the file (see "Defining Front Panel Controls").

6. Save the file and choose Restart Workspace Manager from the Workspace Menu.

**Example**

To remove the logo from the bottom row, comment the Logo control.

```
BOX Bottom
{
   TYPE            secondary
# CONTROL         Logo
   CONTROL         Lock
       . . .
}
```

**Example**

You want to deactivate the Trash drop zone to prevent users from removing files. First, find the name of the control:

```
BOX Bottom
{
    . . .
   CONTROL         Trash
}
```

Next, find the control definition and edit the lines providing the drop functionality. This example assumes you've created an action DisplayTrashError that displays a message informing the user that the control has been deactivated.

```
CONTROL Trash
{
    . . .
# DROP_ACTION     f.action DisplayTrashError # Displays error message
   DROP_ANIMATION  TrashDrop
       . . .
}
```

**See Also**

- "Defining Front Panel Controls" explains how to define various types of controls.
- "Reference: Front Panel" provides additional information about syntax and keywords.
- "Example: Adding a control to the bottom row" is a step-by-step example of creating a control.
- "Changing the Default Terminal, Editor, and Mailer" in Chapter 18 covers how to modify those controls.
- "Customizing Printing" in Chapter 18 covers configuring the Printer controls.

# To modify a control in the top or bottom row

1. Find the control definition using its control name:

   ```
   CONTROL  control_name
   {
       ...
   }
   ```

   If you don't know the control name, you can find it using the row defintion.

   ```
   BOX Top       or     BOX Bottom
   {
       CONTROL      control_name
           ...
   }
   ```

2. Edit the control definition. You can add or remove fields, or simply change the value of a field.

3. Save the file and choose Restart Workspace Manager from the Workspace Menu.

You cannot change the position of the busy (logout) button. It must be on the right side of the bottom (secondary) row.

# Example: Adding a control to the bottom row

You want to add a button to the bottom row to start the Wingz application. You want the button to be between the Terminal and TextEdit controls.

Ordinarily, the command line you use to start the application is:

   ```
   Wingz
   ```

The icon you'll use for the new control is in file is in /users/anna/bitmaps/WingzIcon (/users/anna is your home directory).

**Create the action Wingz**  1. Double-click CreateAction in the Utilities subdirectory of the General Toolbox.

2. Fill out the fields:

   ```
   Name:           StartWingz
   Command line:   Wingz
   ```

3. Select X Windows from the Window Type options menu.

4. Choose Apply, then Close.

**Add the control to the Front Panel**  5. Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox. This opens your personal Workspace Manager configuration file (*/HomeDirectory/*.vue/vuewmrc) for editing.

6. In vuewmrc find the definition for the bottom row and insert the control at the proper place in the box:

   ```
   BOX Bottom
   {
     TYPE            secondary
       ...
     CONTROL         Terminal
     CONTROL         WingzControl
     CONTROL         TextEdit
       ...

   }
   ```

7. Insert the control definition elsewhere in the file:

   ```
   CONTROL WingzControl
   {
     TYPE            button
     HELP_STRING     "Start the Wingz" application.
     PUSH_ACTION     f.action StartWingz
     IMAGE           WingzIcon
   }
   ```

8. Choose Restart Workspace Manager from the Workspace menu.

# Creating and Modifying Subpanels

A subpanel is "attached" to a control on the top row. There are two entries in the Workspace Manager configuration file that attach the subpanel.

- The control definition contains a line:

      SUBPANEL *subpanel_name*

  that attaches the subpanel to the control.

- Since the subpanels are defined in other files, the Workspace Manager configuration file must include those files:

      INCLUDE
      {
       *path_to_subpanel_file*
             . . .
      }

**See Also**

- "To configure a control in the Printers subpanel" in Chapter 18
- "To add an application to the Tools subpanel" in Chapter 23

## System subpanel configuration files

Subpanels and their controls are defined in these files in the `/usr/vue/config/panels` directory.

| | |
|---|---|
| `fp.clock` | Attached to Clock control. Default file is empty (no subpanel). |
| `fp.date` | Attached to Date control. Default file is empty (no subpanel). |
| `fp.load` | Attached to Load control. Default file is empty (no subpanel). |
| `fp.style` | Attached to Style control. Default file is empty (no subpanel). |
| `fp.help` | Help subpanel. Contains default help subpanel. |
| `fp.printer` | Printers subpanel. Contains configurable print controls. |
| `fp.mail` | Attached to Mail control. Default file is empty (no subpanel). |
| `fp.home` | Attached to Home control. Default file is empty (no subpanel). |
| `fp.toolbox` | Toolboxes subpanel for regular HP VUE sessions. Contains buttons for built-in toolboxes. |
| `fp.tool` | Tools subpanel for HP VUE lite sessions. |
| `fp.trash` | Trash subpanel. Default file is empty (no subpanel). |

If the subpanel configuration file is non-empty, the control to which the subpanel is attached has an up arrow for displaying the subpanel.

# To edit a system-wide subpanel configuration file

1. Log in as root.

2. Open the file for editing.

3. Remove the paragraph near the top of the file containing the `=@@=` symbol. This prevents a future update from automatically replacing the edited file.

# To create a personal subpanel configuration file

- Copy the system subpanel file in **/usr/vue/config/panels** to the */HomeDirectory/*.**vue** directory. Give the new file write permission.

- Edit the new subpanel configuration file to add or remove controls.

- Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox. This opens your personal Workspace Manager configuration file (*/HomeDirectory/*.**vue/vuewmrc**) for editing.

- Edit **vuewmrc** to include the new file by replacing the line that includes the system subpanel file with a line that includes the new file.

```
INCLUDE
{
    "path_to_new_file"       # Replaces path to system file.
}
```

**Example**

You want to "create" and use a personal subpanel that will be attached to the Mail control:

1. Copy **/usr/vue/config/panels/fp.mail** to */HomeDirectory/*.**vue/fp.mail**. Give the file write permission.
2. Edit */HomeDirectory/*.**vue/fp.mail**.
3. Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox to open your personal Workspace Manager configuration file for editing.
4. Edit the line in **vuewmrc** that includes the **fp.mail** file so that it reads:

```
INCLUDE
{
    ...
    $HOME/.vue/fp.mail
}
```

5. When you are done editing **fp.mail**, choose Restart Workspace Manager from the Workspace menu.

**See Also**

- "To create a personal Workspace Manager configuration file ('vuewmrc')" covers how to create **vuewmrc**.

# To add or remove a control from an existing subpanel

1. Open the configuration file for the subpanel:

   - **/usr/vue/config/panel/***filename* for system-wide changes (you must be superuser).

   - /*HomeDirectory*/.**vue/***filename* for personal changes.

2. To add or remove controls, edit the BOX definition for the subpanel.

   ```
   BOX subpanel_name
   {
      TYPE     subpanel
      CONTROL  control_name

        . . .
   }
   ```

3. If you've added or replaced a control, define the control elsewhere in the file.

4. Save the file and choose Restart Workspace Manager from the Workspace Menu.

You cannot put the Switch control (the workspace switch) in a subpanel.

- "To create a personal subpanel configuration file" explains how to create a personal configuration file for a subpanel.

# To add a new subpanel

1. Open the configuration file for the subpanel:

   - **/usr/vue/config/panel/***filename* for system-wide changes.

   - /*HomeDirectory*/.**vue/***filename* for personal changes. (See "To create a personal subpanel configuration file" for instructions on creating this file.)

2. Uncomment the lines that define the box, and edit the box definition to list the controls that will be in the subpanel.

```
BOX subpanel_name
{
  TYPE     subpanel
  CONTROL  control_name
   . . .
}
```

3. Define each control.

4. Save the file and choose Restart Workspace Manager from the Workspace Menu.

The subpanels shipped with HP VUE use the convention of repeating the Front Panel primary control at the top of the subpanel. For example, you can display the top level of help using either the Help control on the top row of the Front Panel or the Top Level control in the Help subpanel.

When you are creating your own subpanel, you can choose to duplicate the primary control in the subpanel unless the primary control is a client window (for example, the Load control) or a one-instance control (such as the Style Manager and Trash controls).

**See Also**

- General information about subpanels:
  - □ "To create a personal subpanel configuration file") covers how to create a personal subpanel configuration file.
  - □ "Defining Front Panel Controls" describes how to create different types of controls.
- Configuring special subpanels:
  - □ "To configure a control in the Printers subpanel" in Chapter 18 covers adding printers to the Printers subpanel.
  - □ "To add a control to the Terminals subpanel" in Chapter 23 covers modifying the Terminals subpanel for HP VUE Lite sessions.
  - □ "To add an application to the Tools subpanel" in Chapter 23 covers adding applications to the HP VUE Lite Tools subpanel.

# Defining Front Panel Controls

Controls can have various behaviors:

- Push button.
- Drop zone.
- File monitor.
- Client window.
- One-instance (toggle) button, which allows only one process to be started.

To a certain extent, you can mix behaviors. For example, you can create a control that is a push button, file monitor, and drop zone. However, you cannot make a client window into a push button.

# To create a push button for an application

- Use the PUSH_ACTION field to describe the behavior of the push button. There are two ways to describe the command to be executed:

  □ Use **f.action** to specify an action:

    **PUSH_ACTION    f.action** *action_name*

  If the action accepts a file argument, you can specify the file argument using the syntax:

    **PUSH_ACTION f.action** "*action_name filename*"

  □ Use **f.exec** to specify a command line:

    **PUSH_ACTION    f.exec** "*command*"

**Example**     The following control runs a game the user has acquired:

```
CONTROL Ball
{
   TYPE         button
   HELP_STRING  "Choose this control to play Ball."
   IMAGE        ball
   PUSH_ACTION  f.exec "/users/anna/bin/games/Ball"
}
```

**Example**     This control starts an action named CutDisp.

```
CONTROL StartCutDisp
{
   TYPE         button
   HELP_STRING  "Choose this control to run cutdisp."
   IMAGE        cutdisp
   PUSH_ACTION  f.action CutDisp
}
```

**Example**     The following control starts Text Editor with the data file
`/users/ellen/PhoneList`.

```
CONTROL EditPhoneList
{
   TYPE         button
   IMAGE        PhoneBook
   PUSH_ACTION  f.action "TextEditor /users/ellen/PhoneList"
}
```

**See Also**    • "PUSH_ACTION field" reference.

# To create a push button and drop zone

1. Use the PUSH_ACTION field to describe the functionality of the push
   button. Use `f.action` or `f.exec` to specify the command.

2. Use the DROP_ACTION field to describe the functionality of the drop zone.
   You must use `f.action` for the command, and write the action so that it
   accepts a file argument.

You can use the same action for the push and drop action.

**Example**

The following control runs the X client **xwud**, which takes a file argument.

```
CONTROL Project1
{
   IMAGE       /usr/vue/icons/Project1.1.bm
   PUSH_ACTION  f.action RunXwud
   DROP_ACTION  f.action RunXwud
}
```

To create the RunXuwd action, double-click CreateAction in the Utilities subdirectory of the General Toolbox and supply values for the fields:

```
Name:            RunXwud
Command Line:    xwud -in $1
Window Type:     X Windows
Filename Prompt: File to display:
```

**See Also**

• "PUSH_ACTION field" reference.
• "DROP_ACTION field" reference.

# To create a control that monitors a file

There are two types of controls that monitor a file—**monitor_file** and **mail**.

**Monitor empty file**

This control type monitors a file, detecting whether it is empty or non-empty.

1. Use the TYPE field to specify type **monitor_file**.

2. Use the MONITOR_FILE field to specify the file to be monitored.

3. Use the IMAGE field to specify the icon used when the monitored file is empty, or if the file doesn't exist.

4. Use the ALTERNATE_IMAGE field to specify the icon used when the file is not empty.

**Monitor mail file**

This control type monitors a file, changing appearance when information is added to the file.

1. Use the TYPE field to specify type `mail`.

2. Use the MONITOR_FILE field to specify the file to be monitored.

3. Use the IMAGE field to specify the icon used when the monitored file is unchanged since the last time it was opened.

4. Use the ALTERNATE_IMAGE field to specify the icon used when the file gets larger. (The IMAGE icon is restored when you open the file—for example, when you start the mail program.)

**Example**

The following control looks for the presence of a file named `calendar` that you expect to be anonymously ftp-ed to your system.

```
CONTROL MonitorCalendar
{
  TYPE            monitor_file
  MONITOR_FILE    /users/ftp/calendar
  PUSH_ACTION     f.exec "/usr/vue/bin/vuepad /users/ftp/calendar"
  IMAGE           calendar_no
  ALTERNATE_IMAGE calendar_yes
}
```

**See Also**

• "MONITOR_FILE field" reference.
• "Reference: Front Panel" provides additional information about syntax and keywords.

# To create a one-instance (toggle) control

1. Use the PUSH_RECALL field to set push-recall to True.

2. Use the CLIENT_NAME to attach the client to the control. The CLIENT_NAME you provide must match the window class name (WM_CLASS) of the client (usually the same as the client name).

3. Use the PUSH_ACTION field to describe the functionality of the push button.

A one-instance control checks to see whether the process started by the button is already running. If the process is not running, the push button functionality is invoked. If the process is already running, the window is moved to the top of the window stack in the current workspace.

**Example**

The following control runs one instance of the datebook application.

```
CONTROL
{
   IMAGE          $HOME/.vue/icons/datebook
   PUSH_RECALL    True
   CLIENT_NAME    datebook
   PUSH_ACTION    f.exec "/usr/local/bin/X11/datebook
}
```

**See Also**

- "PUSH_RECALL field" reference.
- "Reference: Front Panel" provides additional information about syntax and keywords.
- The **xprop(1)** man page explains using **xprop** to obtain the WM_CLASS of a client.
- The **xwininfo(1)** man page covers how to find out the size of a window in pixels.

# To create a client window control

1. Use the TYPE field to specify that the control contain a client.

   **TYPE     client**

2. Use the CLIENT_NAME field to specify the client to be started. Choose a unique client name (*note* the name of the executable file).

   **CLIENT_NAME** *client_name*

3. Optional: use the CLIENT_GEOMETRY field to specify the size of the client window in pixels. If you do this, the LAYOUT_POLICY field must be included, set to **as_needed**.

4. Save the configuration file and choose Restart Workspace Manager from the Workspace menu.

5. Start the client from a command line. The client is linked to the empty client control through the *client_name*:

   *executable* **-name** *client_name*

In order to place a client window in the Front Panel, the client must have a **-name** command-line option.

**Example**

The following control displays a 30 × 20 pixel load meter.

```
CONTROL RemoteLoad
{
  TYPE            client
  CLIENT_NAME     ScribeLoad
  CLIENT_GEOMETRY 30x20
  PUSH_ACTION     f.exec
}
```

To start the xterm window, execute:

```
remsh hpwrite -n "/usr/bin/X11/xload -nolabel -name ScribeLoad" -display hphere:0
```

**See Also**

- "CLIENT_NAME field" reference.

# To animate a button

1. Add the PUSH_ANIMATION and/or DROP_ANIMATION fields to the control definition.

2. Specify the animation sequence using the ANIMATION construct.

Only buttons and drop zones can be animated.

**Example**

The following lines animate a control that starts the Framemaker™ application. The time delay between icons is 300 milliseconds. The example assumes you've create icon files **frame1**, **frame2**, etc.

```
CONTROL Frame
{
  ...
 PUSH_ANIMATION FrameAnimation
    ...
}

ANIMATION FrameAnimation
{
   frame1        300
   frame2
     ...
}
```

**See Also**

- Chapter 16 covers creating icons.
- "PUSH_ANIMATION field" reference.
- "DROP_ANIMATION field" reference.

# Example: Creating a drop zone to start a terminal

This procedure creates a drop zone on the bottom row of the Front Panel. Dropping a directory on the control opens a Terminal Emulator set to that directory. If the user chooses (clicks) the control, it displays a prompt for the directory name.

**Locate the Front Panel Description**

1. Open the personal Workspace Manager configuration file (/*HomeDirectory*/ .vue/vuewmrc).

2. Locate the description of the Front Panel by searching for PANEL FrontPanel.

**Add the control**
3. Add the control to the bottom row box:

```
BOX Bottom
{
  TYPE              secondary
  ...
  CONTROL           TextEdit
  CONTROL           DirectoryTerm
  CONTROL           Busy
}
```

4. Create the control:

```
CONTROL DirectoryTerm
{
  TYPE              button
  IMAGE             DirectoryTerm
  DROP_ACTION       f.action StartDirectoryTerminal
  PUSH_ACTION       f.action StartDirectoryTerminal
}
```

5. Create the file */HomeDirectory/*`.vue/types/DirTerm.vf` and enter the action. When you are done, save the file.

```
ACTION        StartDirectoryTerminal
  ARG-TYPES   DIRECTORY
  WINDOW-TYPE NO-STDIO
  EXEC-STRING hpterm
END
```

6. If the pixmap doesn't exist, use the Icon Editor to create it.

**Activate the changes**
7. Save the file.

8. Open the System_Admin subdirectory of the General Toolbox and double-click ReloadActions.

9. Choose Restart Workspace Manager from the Workspace Menu.

**See Also**
- "To create a personal Workspace Manager configuration file ('vuewmrc')" covers how to create **vuewmrc**.

# General Front Panel Configuration

Front Panel syntax allows you to:

- To configure the workspace switch. This may be useful when you change the number of workspaces.

- Change the location of the Front Panel.

- Change the color and tile of a row or subpanel.

- Include other files as part of the Front Panel definition.

The ability to include other files provides a convenient way to distribute new controls from one system to another.

# To change the number of workspaces

1. Log into the session (HP VUE or HP VUE Lite) for which you want to change the number of resources.

2. Start the EditResources action:

   - Double-click EditResources in the System_Admin subdirectory of the General Toolbox.

   - *Or*, execute **vueaction EditResources**.

3. Add or edit the resource:

   > **Vuewm*workspaceList:** *name* [*name* ... ]

   where *name* is the workspace name. Enclose names containing spaces in quotation marks.

4. Choose restart Workspace Manager from the Workspace Menu.

> **NOTE**
>
> Application windows use the workspace name to appear in the proper workspace. When you change workspace names, you may need to move your applications to different workspaces.

**Example**

This line creates four workspaces with the names listed.

```
Vuewm*workspaceList:  Main Remote "Project 1" "Project 2"
```

## To configure the workspace switch

1. To change the number of rows, edit the switch description:

```
switch  Switch
{
  NUMBER_OF_ROWS    n
}
```

2. To change the width of the switch, use the LAYOUT_POLICY field:

```
switch  Switch
{
  LAYOUT_POLICY    as_needed
}
```

The default value for LAYOUT_POLICY (**standard**) creates a fixed switch width. When LAYOUT_POLICY is set to **as_needed**, the workspace buttons are set to a width that accommodates the longest workspace name.

---

## To change the Front Panel location

**Personal change**

1. Make the Front Panel active (click on the frame or press (Alt) (Tab) repeatedly).

2. Press (Alt) (Space) to display the Front Panel menu, then press (Enter).

3. Move the panel using the mouse or arrow keys.

4. Click the mouse or press (Enter).

The new position will be saved and restored by Session Manager.

**System-wide change**

Set the following resource in **/usr/vue/app-defaults/Vuewm**:

> **Vuewm*geometry:** [±*x-offset*] [±*y-offset*]

---

## To include other files

- Use **INCLUDE** syntax to source in other files:

      INCLUDE
      {
      *path/file_name*

        . . .
      }

The included file must contain an entire panel, box, or control. The **INCLUDE** section cannot be imbedded in existing panel, box, or control definitions.

If you specify a relative path in your personal **vuewmrc** file, the path is interepreted as relative to the home directory.

**Example**

You've defined two controls named Graphics and Spreadsheet in the files
`graphics.fp` and `spreadsheet.fp` in */HomeDirectory/*`.vue` This is how
you would edit `vuewmrc` to add the controls to the bottom row.

```
BOX Bottom
{
   ...
  CONTROL Graphics
  CONTROL Spreadsheet
   ...
}
 ...
INCLUDE
{
  $HOME/.vue/graphics.fp
  $HOME/.vue/spreadsheet.fp
}
```

**Example**

You want to restrict individual users from editing the Top row. To do this:

1. Move the definition of the top row box from
   `/usr/vue/config/sys.vuewmrc` to `/usr/vue/config/fp.TopRow`.
   This is the portion of the file you would move:

   ```
   BOX Top
   {
     TYPE    primary
       ...
   }
   ```

2. Include the new file in `sys.vuewmrc`.

   ```
   INCLUDE
   {
       "/usr/vue/config/fp.TopRow"
   }
   ```

**See Also**

- "To include a remote file" explains how to use `INCLUDE` with networked
  systems.

# To create an entirely new front panel

1. Create the new Front Panel. Give it a unique name:

   **PANEL** *front_panel_name*
   **{**
   **BOX** *box_name*

   . . .

   **}**

2. Create the new boxes and controls.

3. In your resource database use the **name** resource to specify the new front panel.

   **Vuewm*frontPanel*name:** *front_panel_name*

4. Choose Restart Workspace Manager from the Workspace menu to active the changes.

Creating a new Front Panel may be preferable when you want to make extensive changes. The new Front Panel can be:

- In /*HomeDirectory*/.**vue/vuewmrc**.

- In another file included into **vuewmrc**.

You can also create an entirely new Workspace Manager configuration file. Use the resource:

   **Vuewm*configFile:** *path/filename*

to specify the new configuration file.

# Administering a Networked Front Panel

When systems are highly networked, it may be desirable to centralize Front Panel customization. A new Front Panel or an individual control can be created in one location and then made available to other systems.

**See Also**

- "Configuring Access to Remote Data" in Chapter 22 covers configuring systems to access remote files.

## To include a remote file

- If the file is available via an NFS mount, use the syntax:

  ```
  INCLUDE
  {
  hostname:path/filename
  }
  ```

- If the file is not available by an NFS mount, copy it to the other systems.

**Example**

A file **/usr/local/vue/controls/CTerm** on system **central** contains the definition for a control named CentralTerminal. If **central**'s file system is NFS-mounted to all the systems that need the control, these lines include the control:

```
INCLUDE
{
central:/usr/local/vue/controls/CTerm
}
```

# Example: Distributing a control

You want to distribute a control that starts the emacs editor, either by choosing the control or by dropping a file. Both the control and the action will be created on the host **appserver**, which acts as application server.

This procedure assumes:

- The application server is configured to export actions to the recipient hosts.

- The recipient hosts is configured to import actions from the application server.

- The application server's file system is available to the recipient hosts via NFS mounts.

If these conditions are not met, the control must be distribed by copying the files containing the control and action definitions.

**On the application server**

1. Create the file **/usr/vue/config/fp.emacs** for the control definition:

```
CONTROL       EmacsButton
{
  IMAGE        emacs
  PUSH_ACTION  f.action StartEmptyEmacs
  DROP_ACTION  f.action StartEmacsWithFile
}
```

2. Create the file **/usr/vue/config/export/emacs.vf** for the action definitions:

```
ACTION        StartEmptyEmacs
  WINDOW_TYPE NO-STDIO
  EXEC-STRING /usr/local/bin/emacs
END

ACTION        StartEmacsWithFile
  WINDOW_TYPE No-STDIO
  EXEC-STRING /usr/local/bin/emacs %(File)Arg_1%
END
```

**On each recipient
system**

3. Add the control to the bottom row:

```
PANEL Bottom
{
   ...
   CONTROL EmacsButton
}
```

4. Include the control in the Workspace Manager configuration file:

```
INCLUDE
{
appserver: /usr/vue/config/fp.emacs
}
```

5. Open the System_Admin subdirectory of the General Toolbox and double-click ReloadActions.

6. Choose Restart Workspace Manager from the Workspace Menu.

# Reference: Front Panel

The front panel is built hierarchically, from the outside in.

- The panel is a set of boxes, and

- Each box contains a set of controls, and

- Each control is defined by a set of keywords describing its appearance and behavior.

- Subpanels are boxes attached to controls. Only controls in the top (primary) row can have subpanels, and there can be only one subpanel per control.

- Definitions of the panel, boxes, controls, and subpanels consist of a series of fields.
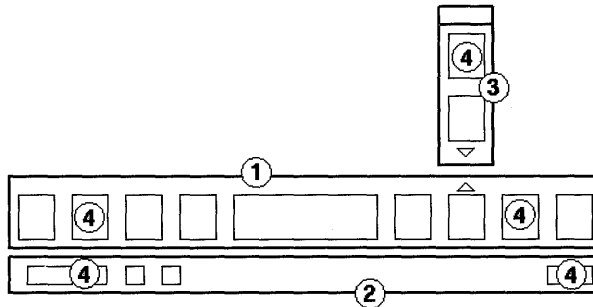


**The front panel consists of primary ①, secondary ②, and subpanel ③ boxes containing controls ④.**

**The entire panel**

The panel consists of a set of boxes, each representing a row. There may be additional fields that define attributes of the entire Front Panel.

```
Panel   Front_panel_name
{
    BOX             Row_name
    BOX             Row_name

        . . .

    other panel fields
}
```

**Row and subpanel boxes**

Boxes consist of a TYPE field and a set of controls. The TYPE field describes whether the box is a top row, bottom row, or subpanel. There may be additional fields that define attributes of the entire box.

```
BOX           Box_name
{
    TYPE      type_of_box
    CONTROL   control_name
    CONTROL   control_name
    CONTROL   control_name

    . . .
    Other box fields
}
```

**Controls**

Controls are defined by a series of fields:

```
CONTROL  control_name
{
    TYPE          control_type
    KEYWORD       value
    KEYWORD       value

    . . .
}
```

**Keywords are not case sensitive**

You can use upper- or lower-case letters in keywords—for example, CONTROL, Control, or control.

# ALTERNATE_IMAGE field

- Control field.
- Specifies icon used for a control when:
  □ Button is pushed.
  □ File monitor detects a change in the monitored file.
- Allowed only for types `monitor_file` amd `mail`.

**Example**

```
CONTROL         MonitorErrorLog
{
  TYPE              monitor_file
  ALTERNATE_IMAGE   Errorchange
      . . .
}
```

# BACKGROUND_TILE field

- Box field.
- Specifies the tiling used for the box background.
- Default for top row and subpanels: no tiling
- Default for bottom row: rib tiling.

**Example**

```
BOX Bottom
{
  BACKGROUND_TILE  50_foreground
      . . .
}
```

# CLIENT_GEOMETRY field

- Control field.
- Specifies size of client window in pixels: *widthxheight*
- Allowed only for type `client`.
- Default is normal size of the control, based on resolution and location (top row, bottom row, or subpanel).

If you specify a CLIENT_GEOMETRY, you must also use the LAYOUT_POLICY field with a value of `as_needed`.

**Example**

```
CONTROL
{
  TYPE              client
  CLIENT_GEOMETRY   100x64
  LAYOUT_POLICY     as_needed
        . . .
}
```

# CLIENT_NAME field

- Control field.
- Specifies a name used to associate the control with the executable.
- Required for types `client` and for buttons where PUSH_RECALL is set to True.

Clients running in the Front Panel for regular HP VUE sessions are saved by Session Manager.

For HP VUE Lite sessions, the client must be started from the startup script.

**Example**

```
CONTROL
{
    TYPE        client
    CLIENT_NAME panelterm
    ...
}
```

To start the terminal, execute (notice the same name, **panelterm**, in the control definition and command line):

```
xterm -name panelterm
```

# DROP_ANIMATION field

- Control field.
- Describes the set of files used to animate the drop action.
- Not allowed for **client**-TYPE controls.
- The control must have a DROP_ACTION field. The file you drop on the control will be used as the argument for the action.

**Example**

```
CONTROL
{
    DROP_ACTION    f.action GraphicsEditor
    DROP_ANIMATION OpenGraphicsEditor
        ...
}
ANIMATION OpenGraphicsEditor
    graphicsed1
    graphicsed2
        ...
```

# DROP_ACTION field

- Control field.
- Specifies the function invoked by dropping a file, usually:

    f.action *action_name*

or

    f.exec "*command*"

**Example**

```
CONTROL
{
    ...
    DROP_ACTION    f.action StartVi
}
```

# HELP_STRING field

- Specifies online help for the entire panel, entire box, or a single control.
- Control help takes precedence over box help, which takes precedence over panel help.

**Example**

```
PANEL MyFrontPanel
{
   HELP_STRING  "Use the Help subpanel for more help."
   BOX          Top
       ...
}
BOX Top
{
   HELP_STRING  "Choose an arrow button to display a subpanel."
   CONTROL      Frame
       ...
}
CONTROL Frame
   HELP_STRING  "Choose this control to start Framemaker."
       ...
```

## IMAGE field

- Control field.
- Specifies the control icon.
- If a relative path is used, the icon must reside in directories on the HP VUE icon search path.
- Allowed for types `button`, `monitor_file`, `mail`, and `busy`.

If no image file is found, the control name is used as a label.

**Example**
```
CONTROL Frame
{
    TYPE   button
    IMAGE frame
    ...
}
```

## LABEL field

- Control field.
- Allowed for types `button`, `monitor_file`, `mail`, and `busy`.

**Example**
```
CONTROL Print_Deskjet
{
    Label "Deskjet (dj3)"
    ...
}
```

## LAYOUT_POLICY field

- BOX field.

- Two values:

standard       The height of each control is determined by display resolution.

as_needed      The height of each control is determined by the image size or the CLIENT_GEOMETRY field.

You must set LAYOUT_POLICY to **as_needed** if you want to put a client window in the front panel with a non-default CLIENT_GEOMETRY.

**See Also**

- "CLIENT_GEOMETRY field"

---

# MONITOR_FILE field

- Control field.
- Specifies the file to be monitored.
- Allowed for types **monitor_file** and **mail** only.

**monitor_file type**

ALTERNATE_IMAGE icon is displayed when the monitored file is non-empty. IMAGE is displayed if the file is empty or non-existent.

**mail type**

ALTERNATE_IMAGE icon is displayed when the file increases in size.

**Example**

```
CONTROL CheckForNewData
{
    TYPE            monitor_file
    MONITOR_FILE    /finance/accounts/July
    IMAGE           NoData
    ALTERNATE_IMAGE DataArrived
      ...
}
```

# PUSH_ACTION field

- Control field.
- Allowed for types `button`, `client`, `monitor_file`, `mail`, and `busy`.
- Specifies the action or command invoked by choosing the control:
  □ To specify an action:

    `f.action` *action_name*

  or

    `f.action` *"action_name file_argument"*

  □ To specify a command:

    `f.exec` *"command"*

**Example**      `PUSH_ACTION     f.action Frame`

**Example**      `PUSH_ACTION f.action "TextEditor /users/anna/folksongs/Lostine"`

**Example**      `PUSH_ACTION     f.exec "/usr/bin/X11`

# PUSH_ANIMATION field

- Control field.
- Specifies the icons to use when button is pushed.
- Allowed for types `button`, `monitor_file`, `mail`, and `busy`.

**Example**

```
CONTROL Frame
{
  PUSH_ANIMATION FrameAnimate
  ...
}
  ...
ANIMATION FrameAnimate
{
  frame1
  frame2
    ...
}
```

# PUSH_RECALL field

- Control field.
- Allowable values: **True** or **False**.
- When **True**, specifies that only one process can be started by the control.
- Allowed for types **button**, **monitor_file**, **mail**, and **busy**.
- Default: **True**.

Set PUSH_RECALL to True to provide behavior like the Style Manager button. The control must use the CLIENT_NAME field to specify the client started. When first chosen, the button starts the process. Once the process is running, choosing the control shuffles the control to the top of the window stack, and, if necessary, moves it to the current workspace.

**Example**

```
PUSH_RECALL   True
```

# TYPE field

**Boxes**

- Allowable types:

  primary        Larger box, usually the top row. Primary boxes can have
                 attached subpanels.

  secondary      Smaller box, usually the bottom row. No subpanels can be
                 atttached.

  subpanel       A box containing a subpanel.

- Default: secondary.

**Controls**

- Allowable types:

  blank          Space-holder control.

  busy           Busy light.

  button         A push button. Buttons can also be drop zones.

  client         A client window.

  clock          Front Panel clock.

  date           Front Panel date.

  mail           Front Panel mail control. The icon changes when the
                 monitored mail file gets larger.

  monitor_file   A control that changes appearance when the contents of
                 a specified file switches from empty (or non-existent) to
                 non-empty.

- Default: button.

**Example**

```
BOX Top
{
 TYPE  Primary
   ...
}

CONTROL
{
  TYPE     monitor_file
   ...
}
```

# 18

Terminal, Mailer, Editor,
and Printer Configuration

# Terminal, Mailer, Editor, and Printer Configuration

The Front Panel provides access to a Terminal Emulator (**hpterm**), a mail program (**elm**), and editor (Text Editor). You may want to substitute an alternative application for one of these.

In addition, HP VUE provides access to a default local printer. However, in many configurations, it is desirable to provide multiple printers and remote printers. Furthermore, it may be desirable to configure the Printer control in the Front panel to discern the type of file being printed and execute a particular print command.

**See Also**

These topics cover modifying the controls specific to HP VUE Lite:

- "Customizing HP VUE Lite Terminal Emulators" in Chapter 23.
- "Adding Applications and Utilities to HP VUE Lite" in Chapter 23.

# Changing the Default Terminal, Editor, and Mailer

The functionality of the Terminal, Editor, and Mail controls are set by actions. To make it easier to modify these controls, the actions they invoke are grouped into a single file: **/usr/vue/types/user-prefs.vf**.

**To customize user-prefs.vf**

The file can be copied to:

- For personal customizations: the */HomeDirectory/*.vue/types directory.
- For system-wide customizations: the **/usr/vue/config/types** directory.

**user-prefs.vf maps the actions to other actions**

The action definitions in **user-prefs.vf** are MAP-type actions. For example, the Front Panel control that starts the mail program invokes an action called "Mailer." The definition for the mailer in **user-prefs.vf** is:

```
ACTION Mailer
     TYPE      MAP      Elm
END
```

The action definition maps the Mailer action to the Elm action. The Elm action is defined in another file (**/usr/vue/types/uxstd.vf**).

**Remapping actions**

The easiest way to change the Mailer, Terminal, Editor, and Printer controls is to remap the action in **user-prefs** to a new action that starts a mailer:

```
ACTION Mailer
     TYPE      MAP      mail_action
END
```

**See Also**

- "Creating Actions Manually" in Chapter 21 and "Example: creating a MAP action" in Chapter 21 describe creating MAP actions.

# To change the default Terminal Emulator

1. If */HomeDirectory/*`.vue/types/user-prefs.vf` does not exist, create it by copying `/usr/vue/types/user-prefs.vf`. Give the new file write permission.

2. Remap the TERMINAL action in `user-prefs.vf` to a different action.

3. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

**Example**

The following lines in */HomeDirectory/*`.vue/types/user-prefs.vf` set the default terminal to `xterm`. (`Hpterm` is replaced with `Xterm`).

```
ACTION Terminal
        TYPE         MAP        Xterm
END
```

**See Also**

- "Customizing HP VUE Lite Terminal Emulators" in Chapter 23 covers configuring the Terminal button and subpanel for HP VUE lite.
- "Creating a Simple Action" in Chapter 19 covers how to use CreateAction.

# To change the default editor

1. If */HomeDirectory/*`.vue/types/user-prefs.vf` does not exist, create it by copying `/usr/vue/types/user-prefs.vf`. Give the new file write permission.

2. Remap the EditText action in `user-prefs.vf` to a different action.

3. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

Remapping the editor also changes the editor started from the Open command in File Manager's Actions menu when a data file is selected.

**Example**

The following lines in */HomeDirectory/*.vue/types/user-prefs.vf set the default editor to vi. (TextEdit is replaced with Vi).

```
ACTION EditText
        TYPE        MAP        Vi
END
```

# To change the default mailer

1. If */HomeDirectory/*.vue/types/user-prefs.vf does not exist, create it by copying /usr/vue/types/user-prefs.vf. Give the new file write permission.

2. **Remap** the MAILER action to a different action.

3. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

**Example**

The following lines in */HomeDirectory/*.vue/types/user-prefs.vf change the default mailer, assuming you've created an action named Mailx that executes a mail program on your system. (Mailx replaces Elm)

```
ACTION Mailer
        TYPE        MAP        Mailx
END
```

The Mailx action can created by starting CreateAction in the Utilties subdirectory of the General Toolbox, and setting these fields:

```
Name:  Mailx
Command Line: mailx
Window Type: Terminal
```

**See Also**

• "Creating a Simple Action" in Chapter 19 covers how to use Create Action.

# Customizing Printing

HP VUE provides a number of printing features

- The Front Panel printer control (regular HP VUE sessions only) includes:
  - □ A push button for displaying printer status.
  - □ A drop zone for printing a file.
  - □ A Printers subpanel for additional system printers.
- The Print command in the File Manager Actions menu.
- Printing within Help Manager
- Printing within applications.

**See Also**
- Configuring printing within an application is covered in the documentation for the application.

## To change the printer drop zone and File Manager Print command

1. If */HomeDirectory/*.vue/types/user-prefs.vf does not exist, create it by copying /usr/vue/types/user-prefs.vf. Give the new file write permission.

2. Remap the the Print action in user-prefs.vf to a different action.

To change the drop zone behavior without affecting other print behavior, you can directly edit the definition of the Printer control in vuewmrc.

**Example**
The following lines in */HomeDirectory/*.vue/types/user-prefs.vf configure the Printer control in the top row to execute the following command when a file is dropped:

```
lp -dlj3 filename
```

```
ACTION Print
        TYPE        MAP        PrintLj3
END
```

The PrintLj3 action can be created by starting Create Action in the Utilties subdirectory of the General Toolbox, and setting these fields:

```
Name:           PrintLj3
Command Line:   lp -dlj3 $1
Window Type:    No Output
Filename Prompt: Raw file to print:
```

**See Also**

- "Creating a Simple Action" in Chapter 19 covers how to use CreateAction.
- "To create a push button and drop zone" in Chapter 17 explains how to define a drop zone in the Front Panel.

# To change the functionality of the printer push button

1. Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox. This opens your personal Workspace Manager configuration file (*/HomeDirectory/*.vue/vuewmrc) for editing.

2. Locate the definition of the Printer control:

```
CONTROL Printer
{
   TYPE       button
       ...
}
```

3. Edit the PUSH_ACTION field.

4. Save the file and Choose Restart Workspace Manager from the Workspace Menu.

**Example**

The following change to the PUSH_ACTION field changes the behavior of the printer control. When pushed, the button will prompt for a file name to print.

```
CONTROL Printer
{
       ...
```

```
    PUSH_ACTION   f.action Print
        ...
}
```

# To configure a control in the Printers subpanel

**To change the behavior of a control**

1. Copy `/usr/vue/examples/printerNN.vf` to:

   - For system-wide configuration:
     `/usr/vue/config/types/printernn.vf`.

   - For personal configuration:
     `/HomeDirectory/.vue/types/printernn.vf`.

   where *nn* is the number of the control. The controls are numbered from top to bottom; 01 is the control beneath the **Default** control.

2. Edit `printernn.vf`:

   a. On the line:

      ```
      set PRINTER_NUMBER=NN
      ```

      change NN to the number of the control ( 01, 02, etc.).

   b. On the line:

      ```
      set PRINTER_NAME=name
      ```

      replace **name** with the device name.

   c. If necessary, edit the EXEC-STRING field. By default, it is set up to execute the command:

      `lp -dname filename`

3. Save the file.

4. Open the System_Admin subdirectory of the General Toolbox and double-click ReloadActions.

> **NOTE**
>
> The top control in the subpanel (`Default`) is configured to duplicate the primary control in the top row of the Front Panel. You cannot change the behavior of that control using the `printerNN.vf` file. If you want to change the behavior of this control, you must edit the control named `printer0` in the `fp.printer` file.

**To configure the appearance of a control**

1. Open `fp.printer`:

   - For system-wide changes: `/usr/vue/config/panels/fp.printer` (you must be superuser).

   - For personal changes: create a personal `fp.printer` file (see "To create a personal 'fp.printer' file").

2. In `fp.printer`, edit the LABEL or IMAGE field for the control.

3. Chooose Restart Workspace Manager from the Workspace Menu.

**See Also**

- "Where to put icon files" in Chapter 16

---

## To create a personal 'fp.printer' file

1. Copy `/usr/vue/config/panels/fp.printer` to the `/`*HomeDirectory*`/.vue` directory. Give the file write permission.

2. Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox. This opens your personal Workspace Manager configuration file (`/`*HomeDirectory*`/.vue/vuewmrc`) for editing.

3. In `vuewmrc`, replace the line that includes `/usr/vue/config/fp.printer` with a line that includes the new file.

```
INCLUDE
{
    ...
        HomeDirectory/.vue/fp.printer
}
```

## To add another printer control to the Printers subpanel

1. If the change is personal rather than system wide, create a personal `fp.printer` file (see "To create a personal 'fp.printer' file".

2. Open `fp.printer` and add a control to the subpanel box.

```
BOX PrinterSubpanel
{
    TYPE        subpanel
        ...
    CONTROL   new_control_name
}
```

3. Add the definition for the new control to the file.

4. Save the file.

5. Create the printer action and reload the database (the database is automatically reloaded if you use CreateAction).

6. Choose Restart Workspace Manager from the Workspace menu.

**See Also**
- "To create a personal subpanel configuration file" in Chapter 17 explains how to create a personal `fp.printer` file.
- "Creating a Simple Action" in Chapter 19 covers creating actions with the CreateAction utility.

# Example: Adding a printer control
# to the Printers subpanel

These steps create a personal Printers subpanel with an additional control
that prints multiple copies of a file using the command:

lp -n*number* -dlj3si *filename*

Assume your home directory is **/users/tim**.

**Create a personal
'fp.printer' file**

1. Copy **/usr/vue/config/panels/fp.printer** to
   **/users/tim/.vue/fp.printer**. Give the new file write
   permission.

2. Double-click EditVuewmrc in the System_Admin subdirectory of
   the General Toolbox. This opens your personal Workspace Manager
   configuration file (*/HomeDirectory/*.vue/vuewmrc) for editing.

3. In **vuewmrc** edit the line that includes **fp.printer**:

```
INCLUDE
{
    ...
  "/usr/vue/config/panels/fp.printer"
```

   to include the personal file instead. Save the file.

**Add the control to
'fp.printer'.**

4. Open **/users/tim/.vue/fp.printer** and add the new control to the
   subpanel box:

```
BOX PrinterSubpanel
  {
  TYPE      subpanel
    ...
  CONTROL   MultiplePrint
}
```

5. Add the control to the bottom of the file.

```
CONTROL MultiplePrint
{
    TYPE          button
    LABEL         "Print n"
    IMAGE         ljet
    PUSH_ACTION   f.action PrintCopies
```

```
              DROP_ACTION  f.action PrintCopies
              HELP_STRING  "This control multiple copies of a file"
          }
```

6. Save `fp.printer`.

**Create a new action**  The action is created manually in order to use a non-file argument (the number of copies).

7. Create the file **/users/tim/.vue/types/PrintCopies.vf**.

8. Add the action definition to the file:

```
    ACTION        PrintCopies
      EXEC-STRING lp -n%"Number of copies:"% \
                      -dlj3si %(File)Arg_1"File to print:"%
    END
```

9. Save the file.

10. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

**Restart the Workspace Manager**  11. Choose Restart Workspace Manager from the Workspace menu.

---

# To provide file-type dependent printing

1. Create the file type.

2. Create a special printing action for that file type. You must edit a database configuration file to do this (you cannot use CreateAction).

3. Map the Print action to the new printing action for that file type.

The filetype-dependent printing will apply to:

- The drop zone in the Front Panel.

- The Print command in the File Manager actions menu.

**See Also**

- "Example: providing file-type dependent printing" is a step-by-step example.
- "Creating Filetypes" in Chapter 20 explains how to create filetypes.
- "Creating Actions Manually" in Chapter 21
- "Example: creating a COMMAND action" in Chapter 21 and "Example: creating a MAP action" in Chapter 21 show examples of creating actions.

# Example: providing file-type dependent printing

Your current project is using the file naming convention *.glp. You print all these files on a particular printer using the command:

    lp -dlj4 -oc *filename*

You want to be able to drop a project file on the Front Panel Printer control, or choose Print from the Actions menu.

1. Open a new file */HomeDirectory/*.vue/types/GlpPrint.vf for editing.

2. Add this file type definition:

```
FILETYPE        Glp
   FILE-PATTERN  *.glp
   DESCRIPTION   Data file for the GLP project.
   ACTIONS       Print
END
```

   If you have icons for the filetype, you specify them using the L-ICON and S-ICON fields.

3. Add this action definition:

```
ACTION          GlpPrint
   WINDOW-TYPE   NO-STDIO
   EXEC-STRING   lp -dlj4 -oc %(File)Arg_1%
END
```

4. Add the mapping action:

```
ACTION          Print
   ARG-TYPES     Glp
   TYPE          MAP       GlpPrint
END
```

5. Save the file.

6. Open the System_Admin subdirectory of the General Toolbox and double-click ReloadActions.

# To configure Help Manager printing

Resources for Help Manager let you change the destination printer used when you print from Help Manager.

**See Also**

- "To choose a printer for yourself" in Chapter 4

# Introduction to Actions and Filetypes

# Introduction to Actions and Filetypes

Actions make it easier to run applications by letting you represent the application as an icon that you can manipulate. When you create an action, you integrate the application into the HP VUE environment and "teach" HP VUE how to run the application.

Filetypes provide the ability to provide different appearance and behavior for different types of files. For example, you can create different filetypes for the data for various applications.

There are two ways to create actions:

- Using the Create Action utility in the General Toolbox.

- Manually editing a database file.

Creating filetypes requires you to edit a database file.

# Creating a Simple Action

Create Action creates an action for an application and adds the action to your Personal toolbox.

The Create Action utility is located:

- HP VUE sessions: Initially, it is placed in your Personal Toolbox. There is another copy in the Utilities subdirectory of the the General Toolbox.

- HP VUE Lite sessions: The Tools subpanel.
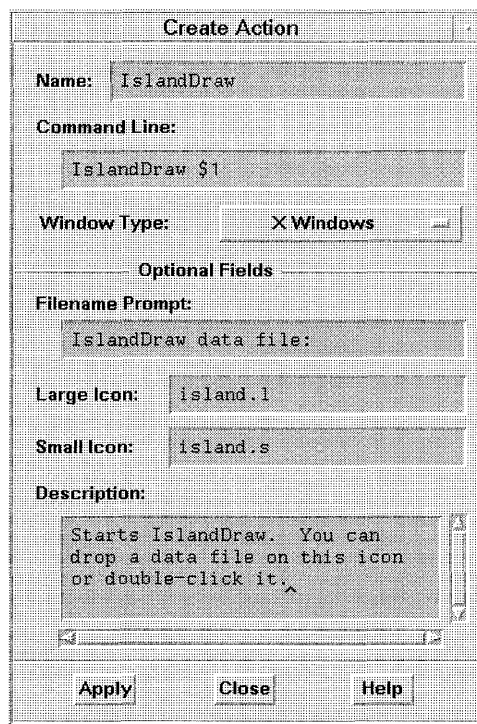


**Use the Create Action utility to create new actions.**

**See Also**
- "Limitations of Create Action"
- Chapter 6 explains how to use the toolboxes.

- "Creating Actions Manually" in Chapter 21 covers writing actions by editing the database directly.

# What Create Action does

Use Create Action to define new actions. HP VUE uses actions to provide icons representing applications or other utility programs and scripts.

Create Action automatically does these things:

- Displays the Create Action dialog that lets you create the definition for the action. When you are done filling in the dialog, Create Action creates a file containing your new action definition.

- Rereads the database so that the action takes effect immediately (regular HP VUE sessions only).

- Creates an icon for the action in your Personal Toolbox (regular HP VUE sessions only). You can double-click the icon to start your application.

Create Action is useful for creating new actions. It cannot be used to edit existing actions.

**See Also**

- "Limitations of Create Action"

# To use the Create Action utility

> **NOTE**
>
> The procedure for using Create Action is different for regular HP VUE sessions and HP VUE Lite sessions. See "To use Create Action in HP VUE Lite" in Chapter 23.

**Start Create Action**

1. Double-click CreateAction in your Personal Toolbox, or in the Utilities subdirectory of the General Toolbox.

**Supply the required information**

2. In the Name field, type a unique name for the action. The name can't include spaces. Try to keep the name under 11 characters long so that the action will work on systems that do not support long file names.

3. In the Command Line field, type the command that starts the application. Type the command exactly the way you would type it in a command line, except that, where you would type a file name, substitute $n, where n is an integer. Here are some sample command lines:

```
emacs
bitmap $1
diff $1 $2
lp -oraw $1
```

The command lines are not passed to a shell unless you explicitly specify the use of a shell. These lines use use shell processing:

```
/bin/sh -c 'ps | lp'
/bin/sh -c 'spell $1 | more'
```

4. Use the Window Type options menu to select how the action's output will be displayed:

| | |
|---|---|
| X Windows | The application creates it's own window. |
| No Output | The command has no display output. |
| Terminal | HP VUE will run the application in a new Terminal Emulator window. The window will remain open until you explicitly close it. |
| Terminal (auto-close) | HP VUE runs the application in a new Terminal Emulator window that closes automatically when you exit the application. |

If you don't know which option to select, leave it set to the default (Terminal).

**Supply a prompt for data**

5. If the command takes a data file (the Command Line contains $1), type the text of the prompt into the Filename Prompt field. This is the text that will be displayed when the action displays a dialog box requiring you

to enter a file name. Leave this field empty if the command does not take a data file.

**Supply optional information**

6. If you have special icons for the action, type the filenames in the Large Icon and Small Icon Fields. (You will have to supply the full path if the icon file is not located in a directory on the icon search path.) These are the icons that will be used for the action is the toolbox.

7. In the Description field, type the item ([F1]) help for the action icon. You can use [Enter] to wrap text to the next line in the text box.

**Store the action**

8. Choose Apply. A dialog box appears telling you the name of the database file created for the new action, and the system is busy momentarily as the database is reread. Do *not* close the Create Action window yet.

9. Open your Personal Toolbox. You should see an action icon for your new action.

**Test the action**

10. Double-click the icon. If the action accepts file data, you can also drop a data file onto it.

11. If necessary, edit the fields in Create Action and reApply the changes.

   As long as the Create Action window is open, you can edit the action, apply the changes, and test the modified action. Once Create Action is closed, you cannot use it to edit the action— you must manually edit the file created by Create Action for the action definition.

12. When the action behaves properly, choose Close in the Create Action dialog.

**See Also**

- "Where to put icon files" in Chapter 16 explains where you should place personal icons.
- "Limitations of Create Action" describes the types of actions you must create by editing the database manually.
- "To provide shell capabilities in an action" in Chapter 21 explains how to write actions that use shell processing.

# To use an action created with Create Action

**Regular HP VUE sessions**

- To run an action that doesn't require file data, double-click the action icon.
- To run an action that requires file data:
  - □ Drop a data file on the action to run the application with that file.
  - □ *Or*, if you included a Filename Prompt when you defined the action, you can double-click the action icon to prompt for the file name. Supply the absolute path to the file, or a path relative to your home directory.

**HP VUE Lite sessions**

- You can attach the action to the Tools control in the top row (see "To assign an application to the HP VUE Lite Tools control" in Chapter 23).
- You can add the action to the Tools subpanel (see "To add an application to the Tools subpanel" in Chapter 23).

**See Also**

- "To make a personal action more readily available" in Chapter 6 explains copying action icons to other directories.

# Limitations of Create Action

You cannot use Create Action to define an action if:

- The action is to be limited to a particular filetype.
- The action requires a non-file parameter. For example, you cannot use Create Action to write an action for the command:

      lp -d*device*

  where the user has to supply *device* each time the command is executed. (You can manually create an action that does this.)

- The action uses complex shell syntax. Instead, you can:

□ Write a shell script, and use Create Action to write an action that starts the script.

□ Create the action manually by editing a database file.

• The action is a MAP-type action.

You can use Create Action to write actions that run applications on remote systems by using **remsh** in the Command Line. However, actions using **remsh** will not take advantage of the automatic xhosting mechanism provided by actions written manually (see "To configure the local host for remote execution" in Chapter 22).

**See Also**
• "Creating Actions Manually" in Chapter 21 covers how to create actions by editing a database file.

## To make a general action with Create Action

1. Create the action using the Create Action utility. When you apply the action, make note of the file to which the definition is saved.

2. Log in as root.

3. Copy the definition file to **/usr/vue/config/types**.

4. Create an action icon in the General Toolbox:

• Copy the action icon (Ctrl+drag) from the Personal Toolbox to the General Toolbox.

• *Or*, create a new file in **/usr/vue/config/types/tools** with the same name as the action name. Make the file executable.

# Example: Creating an action for an application

You want to create an action to start the Island Draw™ application on your system and simutaneously opens a data file in appication. The command line for this is:

IslandDraw *filename*

**Start Create Action**

1. Open the Utilities subdirectory of the General Toolbox.

2. Double-click CreateAction to display the Create Action dialog box.

**Define the action**

3. Fill out these required fields:

        Name:          IslandDraw
        Command Line:  IslandDraw $1

4. Select X Windows from the Window Type options menu.

5. Supply a prompt for the data file:

        Filename Prompt: IslandDraw data file:

**Create and test the action**

6. Choose Apply. When the new action has been written to the database, a message is displayed telling you the name of the file containing the new definition.

7. Open the Personal Toolbox.

8. Doubleclick IslandDraw to display a prompt for a data file. Supply a data file and choose OK.

9. When you are satisfied that IslandDraw is working properly, choose Close in the Create Action dialog.

**See Also**

- "Adding Applications and Utilities to HP VUE Lite" in Chapter 23 explains how to add an application to HP VUE lite.

# Creating a Simple Filetype

Filetypes connect applications with their data. For example, you can create a filetype for an application's data files. The filetype lets you define:

- A unique File Manager icon for the data files.

- Custom double-click behavior for the data files. For example, you might want to be able to start the application with that data simply by double-clicking the data file.

## To create a simple filetype

1. Create a database file, or open an existing file, in
   */HomeDirectory/*.**vue/types**. The file name must end with
   .**vf**.

2. Start the filetype definition by adding the line:

   **FILETYPE**    *unique_name*

3. Specify the icons for the filetype:

   **L-ICON**    *filename*
   **S-ICON**    *filename*

4. Set the file name criteria for typing the file:

   **FILE-PATTERN**    *pattern*

   "To specify the criteria for categorizing the file" in Chapter 20 introduces how to use FILE-PATTERN.

5. Give the filetype its own item help.

   **DESCRIPTION**    *text*

6. Define the double-click behavior. (Any additional actions listed are added to the Actions menu in the File Manager.)

   **ACTIONS**  *Action_name*[,*Action_name* ... ]

7. End the filetype definition.

   **END**

8. Save the database file.

9. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

## Example: Creating a personal action and filetype

You want to make it easier to run the **xgif** program, which displays "gif" pictures. Ordinarily, you run the program by executing:

   **xgif** *filename*

You want to be able to display "gif" pictures several ways:

- Using the Personal Toolbox: by double-clicking the application icon or dropping a "gif" data file on it.

- Using File Manager: by double-clicking a "gif" data file, or selecting the data file and choosing the application from the Actions menu.

**Start Create Action**

1. Open the Utilities subdirectory of the General Toolbox and double-click Create Action.

**Define the action**

2. Fill out the required fields:

   ```
   Name:           GifViewer
   Command Line:   /usr/local/bin/X11/xgif $1
   ```

3. Select X Windows from the Window Type options menu.

4. Fill out the optional fields

```
Filename Prompt: Gif file to view:
Description:     Gif viewer.
                 Double-click the icon or drop a gif file on it.
```

**Create and test the action**

5. Choose Apply

6. Open the Personal Toolbox and double-click GifViewer. You should see a prompt `Gif file to view:`. Supply a "gif" filename and choose OK. The Gif Viewer should open.

7. Drop a "gif" file on GifViewer. The Gif Viewer should open.

8. When you are satisfied that GifViewer is working properly, choose Close in the Create Action dialog.

**Define the filetype**

9. Open */HomeDirectory/*`.vue/types/GifViewer.vf` for editing.

10. Move to the bottom of the file and start the filetype definition:

```
FILETYPE        Gif
```

11. If necessary, create File Manager icons for the gif filetype in the */HomeDirectory/*`bitmaps` directory. Then, specify the icons in the definition:

```
L-ICON        gif.l.bm
S-ICON        gif.s.bm
```

12. Provide item help on Gif file icons.

```
DESCRIPTION    Gif image file.
```

13. Specify the file naming criteria for the filetype:

```
FILE-PATTERN  *.gif
```

14. Define the Actions menu and default action.

```
ACTIONS  GifViewer
```

15. End the definition. The entire definition looks like this:

```
FILETYPE        Gif
  L-ICON        gif.l.bm
  S-ICON        gif.s.bm
  DESCRIPTION   Gif image file.
  FILE-PATTERN  *.gif
  ACTIONS       GifViewer
END
```

**Update the database**　16. Save the file `gif.vf`.

17. Open the System_Admin subdirectory of the General Toolbox and double-click ReloadActions to reread the database. The new filetype will now be in effect.

**Test the filetype**　18. Select a "gif" file in File Manager.

19. Choose GifViewer from the Actions menu. The Gif Viewer should display the "gif" image.

# General Action and Filetype Concepts

In many cases, Create Action will easily integrate your application into HP VUE. However, the action and filetype database provides additional functionality. To take full advantage of its features, it is helpful to understand some basic concepts. You can use these concepts to design how the application will be integrated into HP VUE.

## Basic concepts of actions and filetypes

**Actions replace command lines**

Actions provide a visual representation of a command. The action definition specifies the command to be executed when the action is invoked, and connects the command with a visual representation (an icon). That icon becomes a graphical user interface for the command.
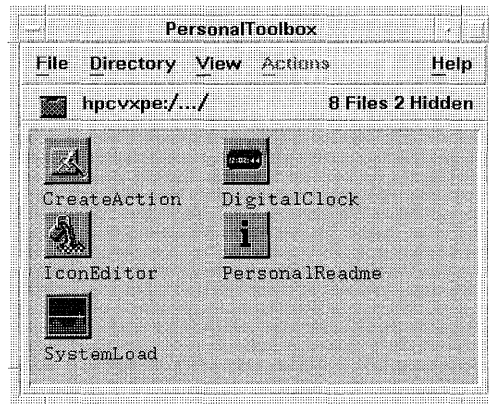


**Actions provide icons that start applications and execute commands.**

**Filetypes connect applications with their data**

A filetype provides both a visual and behavioral distinction for data files:

- It defines a unique icon for the file in File Manager windows.
- It can have a custom Actions menu for tasks you do with the data files.
- It provides "context-sensitivity" for actions. For example, you can write different versions of the Print action for different filetypes; this creates a filetype-sensitive Printer control for the Front Panel.

**Actions can be used different ways**

Once you've created an action for an application, you can set up your environment to start the application from:

- A toolbox.
- The Front Panel or one of its subpanels.
- The Window and Workspace menus.
- By creating a filetype, you can start the application from a File Manager view of one of the application's data files.

**Actions and filetypes have scope**

Scope defines who can create and use actions and filetype:

- Personal actions and filetypes are configured in a subdirectory of the home directory. Therefore, they can be created by an individual user and are available only to that user.
- General (system-wide) actions and filetypes are created in a directory generally restricted to system admininstrators (superusers). These are available to all users on a system.
- Exported actions and filetypes are available to other systems. This allows the system administrator to create and maintain one set of actions and filetypes on a central host.

**Actions are network transparent**

Actions let you start an application exactly the same way regardless of where in the network it resides:

- An action definition specifies where to execute the application. The user doesn't have to worry about such things as host permissions, remote shells, or proper search paths each time the application is started.
- The action can specify a remote directory as the default data location.

• Actions can be defined remotely. An application server can provide both the application and the action, thus centralizing administration.

**See Also**

• "Creating a Simple Filetype" and "Creating Filetypes" in Chapter 20 covers creating filetypes.
• "Creating Actions Manually" in Chapter 21 covers writing actions by manually editing database files.
• These topics cover networking with actions:
  □ "Using Actions to Run Remote Applications" in Chapter 22
  □ "Using Actions to Access Remote Data" in Chapter 22
  □ "Importing and Exporting Actions" in Chapter 22

# Preparing to Integrate an Application Into HP VUE

Before writing filetypes and actions, there are some design issues to consider.

**Defining the scope**    The scope of the action is determined by the database in which it is defined:

• Personal database: for user-specific actions.

• General database: for system-wide actions.

• Exported database: for actions to be exported to other systems. Your database can also include imported actions—actions located on remote systems.

**Connecting applications with data**    If an application uses data files, you may want to create a filetype for the application. At the very least, the filetype can provide a unique icon in File Manager.

The next thing you need to know is whether or not the application can be started with a file argument. This depends on the syntax of the command to start the application. For example, Text Editor can be started with an empty tablet or with a file argument.

    vuepad [*filename*]

Some applications require that data be loaded into the application after the application is started. In this case, no file argument is allowed on the command line.

**Commands without file arguments**

You can provide these behaviors:

- The data file can have its own icon (filetype). Item help on the data file can identify the application.

- The application can be started by double-clicking its icon in a toolbox.

**Commands with file arguments**

You can provide these behaviors:

- The data file can have its own icon (filetype). You can add the applications to the File Manager Actions menu for the filetype.

- The application can have an icon in a toolbox. The application can be started by:

  □ Double-clicking its icon in a toolbox or the Front Panel. The action displays a prompt for the file argument.

  □ Dragging a file of the correct filetype to the application icon.

  □ Selecting a datafile and then choosing the application from the Action menu.

**See Also**

- "The local database directories" covers the locations of local database files.
- These topics cover networking with actions:
  □ "Using Actions to Run Remote Applications" in Chapter 22
  □ "Using Actions to Access Remote Data" in Chapter 22
  □ "Importing and Exporting Actions" in Chapter 22

# The Actions and Filetype Database

**Database files**

Actions and filetypes are defined in a database that consists of a collection of files. The files must:

- Have file names *name*.**vf**.
- Be located in directories along the specified search path.

**Action icons and toolboxes**

Actions are represented visually by action icons. An action icon is simply an executable file with the same name as the action. The contents of the file doesn't matter—the file can be empty.

Action icons can be placed in any directory. However, special directories, called toolboxes, are designed to act as a central, known location.

The default toolboxes are:

- Personal Toolbox.
- General Toolbox.
- Network Toolbox.
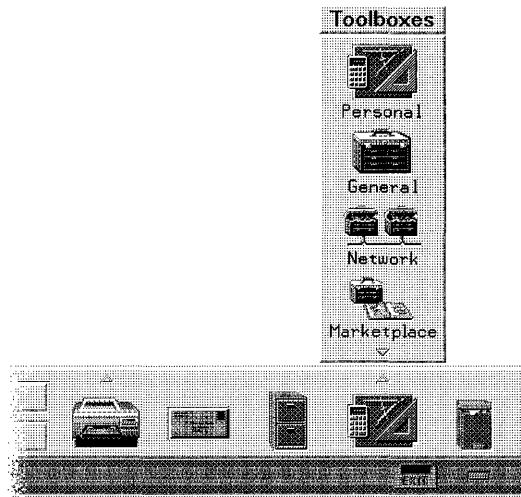
**Use the Tools control and Toolboxes subpanel to display the three toolboxes.**

File Manager navigation within a toolbox is restricted to subdirectories of the toolbox.

## The local database directories

The database is created from definitions in these directories.

- /*HomeDirectory*/.vue/types, the personal database.

- /usr/vue/config/types, the general database.

- /usr/vue/types, the built-in general database. These files may be overwritten by subsequent updates.

Subdirectories of these directories are used for language-dependent definitions.

**Precedence in the database**

Where duplicate definitions exists, the definition read first is used.

- Personal definitions are read before general definitions. Built-in definitions are read last.

- Within a given directory, the `*.vf` files are read in alphabetical order.

**HP VUE 2.01 users**

HP VUE 3.0 will continue to find action and filetype definitions created with HP VUE 2.01. See "What happens to HP VUE 2.01 actions" in Chapter 28.

**Example**

The following list shows the order in which a hypothetical database would be read.

1. */HomeDirectory/*`.vue/types/user-prefs.vf`
2. */HomeDirectory/*`.vue/types/vue.vf`
3. `/usr/vue/config/types/print.vf`
4. `/usr/vue/config/types/vue.vf`
5. `/usr/vue/types/laserrom.vf`

**See Also**

- "Localizing actions and filetypes" in Chapter 27 describes the location for local-language action and filetype definitions.

## The location of the toolboxes

The toolboxes are special directories:

- Personal Toolbox: */HomeDirectory/*`.vue/types/tools`.

- General Toolbox: `/usr/vue/config/types/tools`.

- Network Toolbox: `/usr/vue/config/imports`. The `imports` directory contains directory links to directories `/usr/vue/config/export/tools` on other systems.

You can add an action icon to a toolbox from the command line by creating an executable file with the action name in the toolbox directory. (You must be superuser to write to subdirectories of `/usr/vue`.)

**See Also**

- "To add a host to the Network Toolbox" in Chapter 22 describes how to configure the Network Toolbox.

# To add an action to a toolbox

- If the action was defined using the Create Action utility, the action icon is automatically created. You can copy the icon to other directories.
- If the action was defined by editing a database file:

    1. Create a file named *Action_name* in the toolbox.

    2. Make the file executable.

Superuser permission is required to write to the system-wide database and General Toolbox.

You can also create the action icon directly in the toolbox directory—for example, by creating an executable file in /*HomeDirectory*/.vue/types/tools.

# To reload the database

- Open the System_Admin subdirectory of the General Toolbox and double-click ReloadActions.
- *Or*, log out and back in.

You must reread the database after you've manually edited the database. (The Create Action utility automatically rereads the database.)

# Troubleshooting Actions and Filetypes

If an action or filetype fails to behave as expected, you should check to see that you've used the correct syntax in the definition. HP VUE provides several tools for helping you find the source of the problem.

## To locate the problem in an action or filetype definition

1. Make sure you reloaded the database after creating the action. This is done automatically when you use the Create Action utility. If you create an action manually, you must double-click ReloadActions in the System_Admin subdirectory of the General Toolbox, or log out and back in.

2. Check the HP VUE errorlog (see "To see database errors (errorlog)").

3. Use the ListVueTypes utility. It will list any errors found in the database (see "To list database errors").

4. Correct any errors recorded in ListVueTypes, and then rerun ListVueTypes. (You do not have to reload the database—ListVueTypes reads the database files directly).

**Testing Front Panel actions**

If you added the action to the Front Panel

- Make sure you've restarted the Workspace Manager (choose Restart Workspace Manager from the Workspace Menu.

- You should test the action outside the Front Panel (from the Personal Toolbox or by executing:

    **vueaction** *action_name*

If the action runs correctly outside the Front Panel, then the problem is in the definition of the Front Panel control.

# Common Database Problems

- If the Command Line (EXEC-STRING) uses the executable name (rather than the absolute path), substitute the absolute path to the executable—for example, **/usr/local/bin/emacs** rather than **emacs**. This will correct the problem if the executable file is not in a directory specified by the PATH environment variable.

- If you created the action manually, make sure you used proper syntax:

  □ The action must begin with an ACTION field and end with an END field.

  □ If a field continues to the next line, make sure you used \ to indicate the field includes the next line.

  □ Make sure you have not commented partial lines. For example, the following line is *not* valid syntax:

    ```
    ACTION EditGraphics # Starts the editor for my graphics files.
    ```

- If the action executes an application on another system, make sure both the local and application hosts are configured properly. See "Using Actions to Run Remote Applications" in Chapter 22. If you are using **remsh** to start the application, the execution host must have xhost permission to the local display.

- If the action uses a data file located on another system, make sure the systems are set up to access remote data. See "Using Actions to Access Remote Data" in Chapter 22.

- If the action itself is defined locally, make sure your system and the system containing the action are set up properly. See "Importing and Exporting Actions" in Chapter 22.

- If your action specifies an ARG-TYPE, make sure the filetype is defined.

- If your action is mapped to another action, make sure the mapping is defined properly. The action must eventually map to a COMMAND action.

**See Also**

- "To see database errors (errorlog)"
- "To list database errors"

# To see database errors (errorlog)

- Read the contents of */HomeDirectory/*.vue/errorlog

Any database errors that occur when the database is reloaded are written to the errorlog. The database is reloaded when:

- You apply an action with Create Action.

- You manually reload the database with the ReloadActions action in the System_Admin directory of the General Toolbox.

- You log out and back in.

- You restart the Workspace Manager.

The WatchErrors utility lets you monitor the errorlog continuously (see "Monitoring errors" in Chapter 29).

# To list database errors

- In regular HP VUE sessions, double-click ListVueTypes in the Unsupported subdirectory of the General Toolbox.

- In HP VUE Lite sessions, execute one of the following commands:

      vueaction ListVueTypes
          *or*
      vuetypes

  to run the ListVueTypes action.

ListVueTypes lists the database search path and all the filetypes and actions found along the search path.

To direct the database information to a file, rather than to the display, execute the command:

      vuetypes > *filename*

# General Reference: Actions and Filetypes

These general topics apply to actions and filetypes:

- How the local database search path is determined and changed.
- Using string variables in definitions.
- Using environment variables in definitions.

**See Also**

- "Reference: Action Definitions" in Chapter 21
- "Reference: Filetype Definitions" in Chapter 20

## Changing the local database search path

The default search path for action and filetype definitions includes these directories, listed in the search order:

> *HomeDirectory*/`.vue/types/`*language*
> *HomeDirectory*/`.vue/types`
> `/usr/local/lib/X11/vue/types@local_apps`
> `/usr/lib/X11/vue/types@system_apps`
> `/usr/vue/config/types/`*language*
> `/usr/vue/config/types`
> `/usr/vue/types/`*language*
> `/usr/vue/types`

The linked directories (`local_apps` and `system_apps`) on the search path are needed to find HP VUE 2.01 definitions.

To add a directory to the search path, set a value for the VUEACTIONSEARCHPATH environment variable in `/usr/vue/config/Xsession` (system-wide) or */HomeDirectory/*`.vueprofile` (personal). The order of directories in the variable's value corresponds to the search order. Make sure you include all the default search path directories in the value.

**See Also**

- "To configure the local (importing) host" in Chapter 22 explains how to add a remote directory to the search path.

## String variables in definitions

- Define the string variable in the file using that variable, Use the syntax:

    set   *variable_name=value*

- Reference the string variable using the syntax:

    $ [{] *variable_name* [}]

The variable definition remains in effect from the location of the definition to the end of the file. (There are no global string variables for the database.)

Variable names can contain any alphanumeric characters and underscore (_). Each variable definition must be on a separate line.

**Example**

```
set Remote_Application_Server=hpapp
set Remote_File_Server=hpdata
 :
 :
EXEC-HOST $Remote_Applicaton_Server
CWD       $Remote_File_Server:/doc/project
```

# Environment variables in definitions

- Reference the variable using the syntax: $variable.

The variable is expanded (replaced by its value) when the database is loaded. If a string variable and environment variable have the same name, the string variable has precedence.

**Example**

This execution string prints a file with a banner containing the login name.

```
EXEC-STRING lp -t$LOGNAME %(File)Arg_1%
```

**See Also**

- "How environment variables are handled during remote execution" in Chapter 22 covers how to use environment variables with actions that run remote applications.

**General Reference: Actions and Filetypes**

# More About Filetypes

# More About Filetypes

You can think of filetypes, used in conjuction with actions, as HP VUE components that create a grammar for your system. If you think of files as nouns, then filetypes are the adjectives and actions are the verbs. Like grammar, the pieces are related to one another. There are rules that govern how you put them together and how they affect one another.

**Filetypes are visual**

The visual distinction between filetypes is provided by using a unique icon for the filetype.

**Filetypes are connected to actions**

- A filetype can have a custom default (double-click) action and Actions menu.

- Filetypes provide "context-sensitivity" for actions. For example, you can write different versions of the PRINT action for different filetypes; this creates a filetype-sensitive Print control for the Front Panel.

- Filetypes can restrict actions. You can write an action that accepts only a particular filetype as data.

**Criteria for filetypes**

You can use any of the following criteria for assigning a file to a filetype:

- File name.

- File location (entire path).

- File content.

**Example**

The following definition creates a filetype for the documents for a project named "QS". The files are distinguished by filenames beginning with QS and ending with .doc.

```
FILETYPE          QS_Doc
  FILE-PATTERN    QS*.doc
  DESCRIPTION     This file contains a document for the QS project. \
                  Double-click the file to open it for editing.
  L-ICON          Word_Doc.l
  S-ICON          Word_Doc.s
  ACTIONS         OPEN
END
```

# Creating Filetypes

Creating a filetype involves:

- Choosing the criteria used to type the file.

- Defining the filetype in the database.

- Creating any necessary icons for the filetype.

- "To see database errors (errorlog)" in Chapter 19 and "To list database errors" in Chapter 19 describe ways to correct errors in the filetypes database.

# To define a filetype in the database

1. Open an existing database file or create a new one. New database files must use the naming convention *filename*.**vf** and be located in directories on the database search path:

   - **/usr/vue/config/types** for system-wide filetypes.

   - */HomeDirectory/*.**vue/types** for personal filetypes.

   - **/usr/vue/config/export** for filetypes to be exported to other systems.

2. Add the new filetype definition to the file. Store the file.

3. If necessary, create the bitmaps for the filetype.

4. If necessary, create the actions listed in the ACTIONS field of the filetype defintion.

5. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

**See Also**     • "Getting Started with Icon Editor" in Chapter 16 explains how to make
your own icons.

## To specify the criteria for categorizing the file

• Choose from the following criteria:

| Criteria | Description |
|---|---|
| File name | The filename must match a specified pattern. Use the FILE-PATTERN field. |
| File location | The path must match a specified pattern. Use the PATH-PATTERN field. Use the form *hostname:/path*. |
| File contents | A specified portion of the file's contents must match specified data. Use the CONTENT field. |
| File mode | The file must possess the specified permissions (read, write, execute, directory). Use the MODE field. |

You can use more than one criteria for a filetype. However, you should not
use the FILE-PATTERN and PATH-PATTERN criteria in the same filetype.

> **NOTE**
>
> Use of content-based filetyping will result in lower system performance. Wherever possible, use name-
> and location-based typing instead.

**Example**

The Executable_Object_File filetype defined below applies to all files named with `.o` extensions that are executable (`x`) and (`&`) non-directories (`!d`).

```
FILETYPE            Executable_Object_File
  FILE-PATTERN      *.o
  MODE              x&!d
  ...
END
```

**Example**

The QS_Source_File filetype defined below applies to all files with `.c` extensions in subdirectories of `/projects/QS` on any host.

```
FILETYPE            QS_Source_Files
  PATH-PATTERN      *:/projects/QS/*/*.c
  ...
END
```

**Example**

The Final_pcl_graphics filetype applies to all files named `QS*.pcl`, located on host `hphere`, that are read-only (`!w`) and (`&`) not directories (`!d`) and not executable (`!x`).

```
FILETYPE            Final_pcl_graphics
  PATH-PATTERN      hphere:*/QS*.pcl
  MODE              !w&!d&!x
  ...
END
```

**Example**

The Wingz_data filetype applies to all files with write permission containing the string WNGZ in the first 100 bytes.

```
FILETYPE            Writable_Wingz
  CONTENT           0 100 string WNGZ
  MODE              w&!d
  ...
END
```

**See Also**

- "CONTENT field" reference.
- "PATH-PATTERN field" reference.
- "FILE-PATTERN field" reference.
- "MODE field" reference.

# To associate actions with filetypes

1. Use the ACTIONS field in the filetype definition to list the actions that will appear in the File Manager Actions menu. The first action in the list will be the default (double-click) action.

2. Define the actions listed in the filetype definition. If you want the action to apply only to that filetype, use the ARG-TYPES field in the action definition.

**Example**

The following filetype definition creates a filetype for special "readme" files created by your system administrator that use the naming convention *.rm.

```
FILETYPE SysReadMe
    FILE-PATTERN  *.rm
    L-ICON        SysReadMe.l
    S-ICON        SysReadMe.s
    ACTIONS       Open,Edit
END
```

A special Respond action is defined below for the file. It opens a writeable copy of the file in Text Editor. When the files is saved and Text Editor is exited, the file is mailed to the system administrator (mail address sysadmin@utd).

```
ACTION Respond
    ARG-TYPES    SysReadMe
    EXEC-STRING  /bin/sh -c 'cp %(File)Arg_1% $HOME/readme.temp;      \
                 chmod +w $HOME/readme.temp;                          \
                 vuepad $HOME/readme.temp;    n                       \
                 cat $HOME/readme.temp | /usr/bin/mailx sysadmin@utd; \
                 rm $HOME/readme.temp'
    WINDOW-TYPE  NO-STDIO
END
```

# Example: creating a personal filetype

The following steps create a personal filetype for graphics files for the QS project. The typing criteria is based on the entire path— the files must be on system **hpaaa** in subdirectories of */HomeDirectory/*QS/graphics and end in *.tiff.

**Example**

1. Create the file */HomeDirectory/*.vue/types/Project_graphics.vf, containing:

```
FILETYPE        Project_Graphics
   DEFINITION   Graphics file for the QS project. Double-click the \
                icon to see the graphic.                           \
   L-ICON       QSgraphics.l.bm
   S-ICON       QSgraphics.s.bm
   PATH-PATTERN hpaaa:/$HOME/QS/graphics/*/*.tiff
   ACTIONS      View_Graphics
END
```

2. If necessary, create the two icons specified in the filetype defintion.

3. If necessary, create the View_Graphics action.

4. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

# Example: creating a system-wide content-based filetype

The following steps create system-wide filetype for **shar**-ed files.

1. Log in as root on your system.

2. Create the file **usr/vue/config/types/Shar_file.vf**, containing:

```
FILETYPE        Shar_file
   DESCRIPTION  This is a shar-ed file.  To unpack it, \
                double-click the icon.
   CONTENT      0 80 string "# This is a shell archive."
   L-ICON       shar.l.bm
   S-ICON       shar.s.bm
   ACTIONS      Unpack
END
```

3. If necessary, create the two bitmaps specified in the filetype definition.

4. If necessary, create the Unpack action.

5. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

# Reference: Filetype Definitions

Reference sections cover:

- General syntax.
- A summary of fields.
- An alphabetical listing of all the fields.

**See Also**
- "General Reference: Actions and Filetypes" in Chapter 19
- "Reference: Action Definitions" in Chapter 21

# General syntax for filetype definitions

**Field structure**

A filetype definition consists of a series of fields. The definition begins with the mandatory FILETYPE field and ends with the keyword END. Each field begins with a keyword and ends with a newline character:

```
FILETYPE        filetype_name
    KEYWORD     value
    KEYWORD     value

    . . .
END
```

**Spaces**

Leading spaces in a field are ignored. Therefore, you can use spaces to indent lines for easier reading.

**Fields longer than one line**

The \ character at the end of a line indicates that the field continues onto the next line.

**Comments**    To comment a line, use the **#** character at the beginning of the line. The **#** can be proceeded by spaces. You cannot comment partial-lines.

**Example**    This filetype definition creates a filetype for read-only files whose filenames follow the naming convention **\*.doc**. It assumes that the View_Document action is defined in the database.

```
FILETYPE        ReadOnlyDocument
  DESCRIPTION   This document is not writable. Double-clicking \
    runs your editor with a read-only copy of the file.
  FILE-PATTERN  *.doc
  MODE          !d&!x&!w
                # The file cannot be a directory, executable, or writeable.
  L-ICON        read_only.l.bm
  S-ICON        read_only.s.bm
  ACTIONS       View_Document
END
```

**Example**    This definition creates a filetype for directories named **Demo_**$n$ where $n$ is 0 through 9.

```
FILETYPE         Demo_directory
  DESCRIPTION    This is a directory.  Double-click to open it.
  FILE-PATTERN   Demo_[0-9]
  MODE           d
  L-ICON         control.l.bm
  S-ICON         control.s.bm
  ACTIONS        OpenInPlace,OpenNewView
END
```

**Example**    The following filetype applies to all files named **ch**$nn$**.**$xxx$ where $n$ is 0 through 9, and $xxx$ is any three-character dot extension. The filetype uses the default icons.

```
FILETYPE
  DESCRIPTION    Chapter file for the project document.
  FILE-PATTERN   ch[0-9][0-9].???
  ACTIONS        Edit,Print
END
```

**Example**    This filetype creates a special icon for directories for which you lack both read and write permission.

```
FILETYPE         Directory_With_No_Permissions
```

```
        DESCRIPTION     You cannot open this directory unless you are logged \
                        in as root.
        MODE            d&!r&!w
        L-ICON          admindirectory.l.bm
        S-ICON          admindirectory.s.bm
        ACTIONS         OpenInPlace,OpenNewView
      END
```

**See Also**

• "FILE-PATTERN field" and "PATH-PATTERN field" describe the syntax for pattern matching.

---

## Summary of Filetype Fields

**Required fields**

| Field | Description |
|---|---|
| FILETYPE | Begins the definition, contains the filetype name. |
| END | Ends the definition. |

**Icons**

| Field | Description |
|---|---|
| L-ICON, S-ICON | Absolute path of the large (L) and small (S) icons in File Manager. (Default is data file icon) |
| DESCRIPTION | A description of the filetype displayed when the user obtains item help on the icon. |

**File recognition fields**

| Field | Description |
|---|---|
| FILE-PATTERN | Describes the pattern for matching the file name. |
| PATH-PATTERN | Describes the pattern for matching the entire path. |
| MODE | Mode requirements for the file. Use characters d, r, w, and x, for directory, readable, writable, and executable. Use & or logical AND, ! for logical NOT. (Default is non-directory, other permissions irrelevant). |

|  | CONTENT | Specifies the portion of the file to search and the content to search for. |
|---|---|---|
| **Behavior field** | ACTIONS | List of the entries in the Actions menu, separated by commas. The first action in the list is the default (double-click) action. |

# ACTIONS field

- Lists the actions to be displayed in the Actions menu.
- Optional.
- Defaults to no actions. The Actions menu is inactive.

This field contains a list of actions displayed in the file manager Actions menu when a file of this filetype is selected. Items in the list are separated by commas. There can be no spaces in the list.

The first action in the list is the default action taken when the user double-clicks the file icon.

Each action must have a corresponding entry in the action definition database.

**Example**          The following ACTIONS field lists four actions for the Actions menu.

```
ACTIONS  Edit,Print,Compress,Archive
```

# CONTENT field

- Describes the content requirements for the filetype.
- Optional.
- Can be used in conjuction with FILE-PATH or FILE-PATTERN.

Content-based filetyping searches a specified portion of the file for the existence of a particular string or number. The search starts at the position *starting_byte* and ends at the position *ending_byte*.

> **N O T E**
>
> Wide-spread use of content-based filetyping will result in lower system performance. You should use name- and location-based filetyping instead wherever possible.

**String contents**

Use the syntax:

CONTENT    *starting_byte ending_byte* string "*string*"

**Numeric contents**

Use one of the following syntaxes, depending on the type of number:

CONTENT *starting_byte ending_byte* byte *hexadecimal_number*
CONTENT *starting_byte ending_byte* short *decimal_number*
CONTENT *starting_byte ending_byte* long *decimal_number*

Search is conducted from the *starting_byte* to *ending_byte*.

**Example**

| CONTENT Value | Interpretation |
|---|---|
| 0 30 string 'Chapter' | All files containing **Chapter** in the first 30 bytes. |
| 0 20 byte 0xF | All files containing a byte **F** in the first 20 bytes. |

# FILE-PATTERN field

- Specifies the filename requirements for the filetype.
- Optional.
- Defaults to *.

Use the FILE-PATTERN field to specify the filename criteria for the filetype.

| Character | Meaning |
|---|---|
| ? | Matches any single character. |
| * | Matches any sequence of characters (including a null string). |
| [cc ... ] | Matches any of the characters (c) enclosed in brackets. |
| [c−c] | Matches any of the characters in the range c through c. |

**Example**

| FILE-PATTERN Value | Interpretation |
|---|---|
| *.c | All files ending with .c. |
| chapter?.text | All files named chapter$x$.text, where $x$ is any character. |
| chapter[1−9].text | All files named chapter$x$.text, where $x$ is a character in the range 1 through 9. |

# L-ICON and S-ICON Fields for filetypes

- Specify the icons for the filetype.
  - □ L-ICON: displayed in File Manager views By Name and Icon.
  - □ S-ICON: displayed in File Manager views By Name and Small Icon.

The value for the L-ICON and S-ICON fields can be:

- The absolute path.

- A filename. The system searches the icon search path.

The default icons are in **/usr/vue/icons**.

**Icon file names**    There are no name requirements. You can use `.bm` or `.pm` extensions to distinguish between bitmaps and pixmaps.

**Icon sizes**    Icons can be any size. These are the default sizes.

| Icon Type | Icon Size (Width×Height) |
|---|---|
| Data file L-ICON | 22 × 30 pixels |
| Data file S-ICON | 14 × 17 pixels |
| Executable file L-ICON | 32 × 32 pixels |
| Executable file S-ICON | 16 × 16 pixels |
| Directory L-ICON | 30 × 23 pixels |
| Directory S-ICON | 20 × 16 pixels |

**See Also**    • "Where to put icon files" in Chapter 16 explains where the system looks for icons when you use a filename for L-ICON or S-ICON.

# MODE field

- Contains a boolean expression specifying the mode requirements for the file.

- Optional.

- The default value is a non-directory, all other permissions are irrelevant.

The boolean expression is built from the following operators and values:

| Character | Meaning |
|---|---|
| ! | Logical operator NOT. |
| & | Logical operator AND. |
| d | The file is a directory. |
| r | The file is readable by the current user. |
| w | The file is writable by the current user. |
| x | The file is executable by the current user. |

You can simulate a logical OR operator by defining the filetypes more than once using different values in the MODE field.

**Example**

The following MODE fields specify criteria for file typing.

| | |
|---|---|
| !d | Not a directory. |
| !d&!x | Non-directory, not executable by the user. |
| !d&r | Non-directory, must be readable by the user. |

**Example**

If the mode criteria involves a logical OR, you must define the filetype twice using the two OR criteria. The filetype Doc_Utility, below, applies to an executable or directory with the naming convention Doc*.

```
FILETYPE        Doc_Utility
    FILE-PATTERN  Doc*
    MODE          d
    ...
END

FILETYPE        Doc_Utility
    FILE-PATTERN  Doc*
    MODE          x
    ...
END
```

# PATH-PATTERN field

- Describes the location (path) requirements for the filetype.
- Optional.
- Defaults to *.

PATH-PATTERN filetyping is based on the entire path to the file:
*hostname:/path/filename.*

| Character | Meaning |
|---|---|
| ? | Matches any single character. |
| * | Matches any sequence of characters (including a null string). |
| [cc ... ] | Matches any of the characters (c) enclosed in brackets. |
| [c–c] | Matches any of the characters in the range c through c. |

**Example**

| FILE-PATH Field Value | Interpretation |
|---|---|
| hpaaa:$HOME/templates/* | All files in the directory $HOME/templates on system hpaaa. |
| *:*/Mail/* | All files in any Mail directories and their subdirectories on any system. |
| *:/doc/*/chapter[1–9].text | All files in subdirectories of /doc named chapter*x*.text on any system., |
| hpcvxlp:/doc/sysadm/*.b | All files ending in .b in the /doc/sysadm directory or any of its subdirectories on system hpcvxlp. |

# 21

## More About Actions

# More About Actions

Actions can be created two ways:

- Using the CreateAction utility.
- Manually editing the database.

You can use the CreateAction utility in the General Toolbox to create simple actions that run local applications with or without file data.

You must create an action manually if you want to write an action that:

- Executes a remote application.
- Uses data other than a data file—for example, a device name.
- Maps to other actions.
- Is filetype-specific.
- Has other capabilities not supported by the CreateAction utility.

**See Also**

- "Creating a Simple Action" in Chapter 19 covers using the CreateAction utility.

# Creating Actions Manually

Creating an action manually (by editing a database file) gives you full access to all the functionality of actions.

**Types of actions**     You can create two basic types of actions.

| Action type | Description |
|---|---|
| COMMAND | An action that executes a command. A command action can start an application or utility, run a shell script, or execute an operating system command. |
| MAP | An action that is "mapped to" a command action. Mapping provides the ability to "enrich" an action name by giving it different functionality for different filetypes. You cannot use Create Action to create MAP actions. |

**See Also**

- "To see database errors (errorlog)" in Chapter 19 and "To list database errors" in Chapter 19 describe ways to correct errors in the actions database.

## To create an action manually

1. Open an existing database file or create a new one. New database files must use the naming convention *filename*.`vf` and be located in directories on the database search path.

   - `/usr/vue/config/types` for system-wide actions (you must be superuser to create system-wide actions.

   - /*HomeDirectory*/`.vue/types` for personal actions.

   - `/usr/vue/config/export` for actions to be exported to other systems.

2. Add the new action definition to the file. Store the file.

3. If necessary, create the bitmaps for the action.

4. Open the Utilities subdirectory of the General Toolbox and double-click
   ReloadActions.

5. Create an executable file with the same name as the action in the Personal
   Toolbox or General Toolbox. This file becomes the action icon. The file
   will use the icon and item help specified in the action definition.

**See Also**

- "Example: creating a COMMAND action" and "Example: creating a MAP
  action" are step-by-step examples.

- "Building the Execution String for a COMMAND Action" explains how to
  write an EXEC-STRING field.

- "General syntax for action definitions" covers the syntax of action
  defintions.

## Example: creating a COMMAND action

The following steps create an action that prints one or more PCL graphics file
to a printer named lj3.

1. Create the file /*HomeDirectory*/.vue/types/Print_actions.vf.

2. Put the following action definition into the file:

```
ACTION        Print_PCL_file
  TYPE        COMMAND
  DESCRIPTION Prints a PCL file.
  WINDOW-TYPE NO-STDIO
  EXEC-STRING /bin/sh -c 'lp -dlj3 -oraw \
              %(File)Arg_1"File(s) to print:"% %(File)Args%'
END
```

The syntax %(File)Arg_1"File(s) to print:"% %(File)Args%
accepts one or more dropped files, or prompts for the file argument if the
action icon is double-clicked.

3. Save the file.

4. If necessary, create the PCL filetype. (You can use the same database file.)

```
FILETYPE      PCL
  FILE-PATTERN *.pcl
  L-ICON      pcltype.l
  S-ICON      pcltype.s
  DESCRIPTION Pcl-type file.  Double-click to print.
  ACTIONS     Print_PCL_file
END
```

5. Open the Utilities subdirectory of the General Toolbox and double-click ReloadActions.

6. Open the Personal Toolbox and create an executable file named `Print_PCL_file`.

The Print_PCL_file action can be invoked two ways using its action icon:

- Double-clicking the action icon displays a prompt for the file(s) to print.

- Dropping one or more files on the icon prints the file.

- "To create an action that accepts a dropped file or prompts for one".
- "To write actions with multiple, interchangeable file arguments"

# Example: creating a MAP action

The following steps create an action that "enriches" the Print action for PCL-type files so that the drop zone in the Front Panel and the Print entry in the Actions menu prints a PCL file properly.

**Example**

1. Create the file /*HomeDirectory*/`.vue/types/Print_actions.vf`.

2. Define an action Print_Pcl_File that prints a PCL graphics file (such an action is defined in "Example: creating a COMMAND action").

3. Create an action that maps the Print action to the new Print_Pcl_File action. Use the ARG-TYPES field to specify that the mapping applies only to the PCL filetype.

   ```
   ACTION          Print
      ARG-TYPES    PCL
      TYPE         MAP       Print_PCL_file
   END
   ```

4. Create the PCL filetype.

   ```
   FILETYPE        PCL
      FILE-PATTERN *.pcl
      L-ICON       pcltype.l
      S-ICON       pcltype.s
      DESCRIPTION  Pcl-type file.  Double-click to print.
      ACTIONS      Print
   END
   ```

5. Open the Utilities subdirectory of the General Toolbox and double-click ReloadActions.

# Building the Execution String for a COMMAND Action

The mandatory fields of a COMMAND action are:

> ACTION *action_name*
>     EXEC-STRING *execution_string*
> END

The execution string (EXEC-STRING) is generally the most complex field in an action definition. It uses syntax similar to the command line you would execute in a Terminal Emulator window, but includes additional syntax for handling file and non-file arguments.

The execution string is executed directly, rather than through a shell. However, you can explicitly invoke a shell in the execution string.

**The EXEC-STRING uses the PATH**

If your application is located in a directory listed in the PATH variable, you can use the simple executable name. If the application is elsewhere, you must use the absolute path to the executable file.

**Example**

This execution string starts the client **xclock** with a digital clock. The command line requires no arguments or shell processing, so the execution string is very simple:

> EXEC-STRING xclock -digital

**Example**

Here is a more complex execution string that requires shell processing and accepts a file argument.

> EXEC-STRING /bin/sh -c 'troff -man %(File)Arg_1"Man Page To Print:"%'

# To create an action that uses no data

- Use the same syntax for the EXEC-STRING as you would use to start the application from a command line.

**Example**

This execution string is part of an action that starts the X client **xcutsel**.

```
EXEC-STRING xcutsel
```

# To create an action that accepts a dropped file

- Use this syntax for the file parameter:

   **%(File)Arg_*n*%**

This syntax substitutes the *n*th file argument into the command line. The file can be a local or remote file.

**Example**

This execution string executes **xrdb -load** using a dropped file as the **-load** parameter.

```
EXEC-STRING xrdb -load %(File)Arg_1%
```

**Example**

This portion of an action definition shows that the action works only with directories. When a directory is dropped on the action icon, the action displays list of all the files in the directory with read-write permission.

```
ARG-TYPES      DIRECTORY
   ...
EXEC-STRING /bin/sh -c 'll %(File)Arg_1% | grep rw-'
```

# To create an action that prompts for a file argument

- Use this syntax for the file parameter:

   **%(File)"*prompt*"%**

When the user initiates this action by double-clicking the action icon, a dialog box appears containing a prompt for the file name. If the user enters the path

to a remote file in the form *hostname:/path/filename*, the path is translated to **/nfs**/*hostname/path/file* so that it can be found.

**Example**

This execution string displays a dialog box that prompts for a resource file to load, and uses the supplied file path as the argument for **xrdb -load**.

```
EXEC-STRING xrdb -load %(File)"Resource file:"%
```

# To create an action that accepts a dropped file or prompts for one

- Use this syntax for the file parameter:

    **%(File)Arg_***n*"*prompt*"**%**

If no file is dropped (for example, the user double-clicks the action icon), a dialog box appears with a prompt for the file path.

If the user enters the path to a remote file in the form *hostname:/path/filename*, the path is translated to **/nfs**/*hostname/path/file* so that it can be found.

**Example**

This execution string performs **lp -oraw** on a dropped file. If the action is started without dropping a file, a dialog box appears prompting for the file name.

```
EXEC-STRING lp -oraw %(File)Arg_1"File to print:"%
```

# To use the host:file format in a commmand line

- Use the syntax:

    **%Arg_***n***%**

to use a file name as an argument without converting the syntax from the form:

*hostname*:*/path/filename*

to the form:

**/nfs/***hostname/path/filename*

This syntax is used when a parameter calls for a file name to be provided for some reason other than to specify the *location* of the data—for example, to print a file name banner on output.

You should not use this syntax for parameters requiring the location of the file unless you are certain that the application using the data file understands the *hostname*:*/path/filename* syntax.

**Example**

This execution string prints a file with a banner containing the file name using the command **lp -t***banner file*. For remote hosts, the banner will be printed in the form *hostname*:*path/filename*.

```
EXEC-STRING lp -t%Arg_1% %(File)Arg_1%
```

# To prompt for a non-file argument

- Use this syntax for the non-file parameter:

  *%"prompt"%*

Do not use this syntax when prompting for a file name.

**Example**

This execution string runs the **bitmap** application for a new file. Since the bitmap does not yet exist, you must enter a geometry.

```
EXEC-STRING bitmap -size %"Geometry <width>x<height>:"%
```

## To provide shell capabilities in an action

- Specify the shell in the execution string. Use the syntax:

  **/bin/sh -c** *'command'*

  or

  **/bin/ksh -c** *'command'*

**Example**    This execution string illustrates an action that uses shell piping.

```
EXEC-STRING /bin/sh -c 'ps | lp'
```

**Example**    This execution string requires that the argument be a compressed file. The action uncompresses the file and prints it using **lp -oraw**.

```
EXEC-STRING /bin/sh -c '/bin/cat %(File)Arg_1"File to print:"% | \
                        /usr/bin/uncompress |                    \
                        /usr/bin/lp -oraw'
```

**Example**    This execution string starts a shell script.

```
EXEC-STRING /usr/local/bin/StartGnuClient
```

## To provide different double-click and drop function for an action

1. Create an action definition for the double-click functionality. Use the ARG-COUNT field to specify no arguments. Use a syntax for the EXEC-STRING that does not accept a dropped argument.

2. Create a second action definition for the drop functionality. Use the ARG-COUNT field to specify that action applies to one or more argument (>1). Use a syntax for the EXEC-STRING that accepts a dropped file.

3. Create an action icon in the applications directory.

You can assign actions to controls in the Front Panel.

**Example The following**
**two actions create drop**
**and double-click**

functionality for an action named Vedit. Dropping a file on the action icon displays a read-only copy of the data file in a terminal window. Double-clicking the action icon starts **vedit** with an empty file ready to be edited. The first action ARG-COUNT of 0 has precedence when the database is searched for a match, **ARCOUNT 0** is more specific than **ARG-COUNT** *.

```
ACTION Vedit
# Double-click functionality.
  ARG-COUNT      0
  WINDOW-TYPE    TERMINAL
  EXEC-STRING    vedit
END

ACTION Vedit
# Drop functionality
  ARG-COUNT      *
  WINDOW-TYPE    TERMINAL
  EXEC-STRING    vedit -R %(File)Arg_1%
END
```

**See Also**

• "ARG-COUNT field" reference.

# How HP VUE chooses between two actions with the same name

When an action is invoked, the system searches the database of available actions for a name match. When more than one action exists with that name, the one with the most specific ARG-TYPE or ARG-COUNT takes precedence.

If the actions cannot be distinguished by how specific they are, they are distinguished by scope. The order of the search is: personal actions, general actions (in **/usr/vue/config/types**), built-in actions (in **/usr/vue/types**); the first action encountered in the search is used.

**Example**       Consider the following portions of action definitions.

```
ACTION EditGraphics
# EditGraphics-1
   ARG-TYPES    XWD
       ...
END

ACTION EditGraphics
# EditGraphics-2
   ARG-COUNT  0
       ...
END

ACTION EditGraphics
# EditGraphics-3
   ARG-TYPES  *
       ...
END
```

Double-clicking EditGraphics starts EditGraphics-2, since ARG-COUNT 0 has precedence. When an XWD-type file argument is provided, EditGraphics-1 is used; EditGraphics-3 is used for all other file arguments.

**See Also**
- "ARG-COUNT field" reference.
- "ARG-TYPES field" reference.

# Creating Actions With Multiple File Arguments

There are two ways to implement an action that accepts multiple file arguments:

- The action can use two or more non-interchangeable file-arguments. For example, the command:

    **xsetroot -cursor** *cursorfile maskfile*

  requires two unique files in a particular order.

- The action can perform the same command sequentially on each file argument. For example, the command:

    **pr** *file* [*file* ... ]

  will print one or many files in one print job.

# To write an action for non-interchangeable arguments

- If you want the action to prompt for the file names, use this syntax for each file argument:

    **%(File)**"*prompt*"**%**

    using different prompts for each argument.

- To accept dropped files, use this syntax for each file argument:

    **%(File)Arg_***n***%**

    using different values of *n* for each argument.

**Example**

This execution string prompts for two files.

```
EXEC-STRING xsetroot -cursor %(File)"Cursor bitmap:"% \
                             %(File)"Mask bitmap:"%
```

**Example**

This execution string accepts two dropped files. Since order is essential, it may be necessary to reorder the files in the file manager before dragging them to the action icon.

```
EXEC-STRING xsetroot -cursor %(File)Arg_1% \
                             %(File)Arg_2%
```

# To write actions with multiple, interchangeable file arguments

- To create an action that accepts dropped files and issues a command in the form: *command file₁ file₂* ... , use this syntax for the file arguments:

    `%(File)Args%`

- To create an action that accepts dropped files or displays a prompt when double-clicked, use this syntax for the file arguments.

    `%(File)Arg_n"prompt"% %(File)Args%`

    The action will issue the command in the form: *command file file* ... .

- To create an action that accepts multiple dropped files and issues a series of commands in the form:

    *command file*
    *command file*

    ...

    use this syntax for the file arguments:

    `%(File)Arg_1`

    Use this syntax when the command must be issued separately for each file argument.

**Example**

This execution string creates an action that executes:

    `pr` *file* *file*

with multiple file arguments. The action has no double-click functionality

    `EXEC-STRING  pr %(File)Args%`

**Example**

This execution string creates an action similar to the previous example, except that the action displays a prompt when double-clicked (no file arguments).

    `EXEC-STRING  pr %(File)Arg_1"File(s) to print:"% %(File)Args%`

**Example**

This execution string creates an action that executes:

xwud -in *file*

repeatedly if you supply multiple file arguments.

```
EXEC-STRING  xwud -in %(File)Arg_1%
```

---

## To write an action for multiple dropped files

- To accept multiple file arguments that are dropped on the action, execute a command in the form:

  *command file file* ...

  use the syntax:

  ```
  %(File)Args%
  ```

**Example**

This execution string executes a script named Checkout for multiple files.

```
EXEC-STRING /usr/local/bin/Checkout \
            %(File)Arg_1"Check out what file?"% %(File)Args%
```

**Example**

This execution string performs the **diff** command on two files. None, one, or both files can be supplied by dropping the file(s) on the action icon. A dialog box with prompt(s) appears if no file or one file is dropped.

```
EXEC-STRING diff %(File)Arg_1"Original File:"% %(File)Arg_2"Altered File:"%
```

**Example**

This execution string executes **lp -oraw** with multiple files:

```
EXEC-STRING /usr/bin/lp -oraw %(File)Arg_1"File to print:"% %(File)Args%
```

# Creating Special Actions with 'vueaction'

The **vueaction** utility in HP VUE starts an action. You can use **vueaction** in the EXEC-STRING of an action to create special capabilities, including:

- Writing an action that executes a command and then invokes another action.
- Running an action as a different user.
- Playing back an audio file.

## To write an action that invokes another action

- Use the syntax:

    **vueaction** *action_name* [*file_argument(s)*]

**Example**

The following action uses the built-in action CheckSpelling. The new action runs TextEditor and the spell checker at the same time, displaying the spelling errors in a separate terminal window.

```
ACTION EditAndSpell
    WINDOW-TYPE NO-STDIO
    EXEC-STRING /bin/sh -c 'vueaction CheckSpelling %(File)Arg_1"File"%; \
                            vuepad %(File)Arg_1%'
END
```

# To create an action that runs as a different user

- Use the following syntax in the EXEC-STRING:

EXEC-STRING    vueaction -user *user_name action_name* [*file_argument*]

**Example**

The following two actions provide the ability to edit the system file
`/usr/vue/config/Xsession`.

```
# This action runs a second action, OpenXsessionEdit, as root.
ACTION EditXsession
  WINDOW-TYPE NO-STDIO
  EXEC-STRING vueaction -user root OpenXsessionEdit
END


# The following action makes Xsession writable, opens it for editing,
# and removes the write permission when the editing session is concluded.
ACTION        OpenXsessionEdit
  WINDOW-TYPE NO-STDIO
  EXEC-STRING /bin/sh -c 'chmod +w /usr/vue/config/Xsession \
              vuepad /usr/vue/config/Xsession               \
              chmod -w /usr/vue/config/Xsession'
END
```

# To attach audio playback to an action

- Use the following syntax in the EXEC-STRING:

        EXEC-STRING vueaction -audio *audio_file* \
                              [-volume *value*] \
                              [*action_name* [*file_argument*] ]

**Example**

The built-in action that puts a file in the trash can when you drop a file on the trash icon is named TrashFile. The following action definitions redefine TrashFile to add audio playback when a file is dropped.

```
# This action is identical to the built-in TrashFile action, except it
# has been renamed.
ACTION NewTrashFile
    TYPE            MESSAGE
    MSG-TOOL        FILEMGR
    MSG-COMMAND     REMOVE_TRASH
    MSG-DATA        %Arg_1"File To Trash:"% %Args%
    DESCRIPTION     The NewTrashFile action places its argument \
                    in the HP VUE trash can.
END

# The TrashFile action will now play the audio file and invoke the
# NewTrashFile action.
ACTION TrashFile
    WINDOW-TYPE  NO-STDIO
    EXEC-STRING vueaction -audio /tmp/lid_open NewTrashFile %Args%
END
```

# Reference: Action Definitions

Reference sections cover:

- General syntax.
- A summary of fields.
- An alphabetical listing of all the fields.

See Also

- "General Reference: Actions and Filetypes" in Chapter 19
- "Reference: Filetype Definitions" in Chapter 20

---

## General syntax for action definitions

**Field structure**

An action definition consists of a series of fields. The definition begins with the mandatory ACTION field and ends with the keyword END. Each field begins with a keyword and ends with a newline character:

```
ACTION  action_name
    KEYWORD   value
    KEYWORD   value

      . . .
END
```

**Spaces**

Extra spaces at the beginning or end of lines are ignored, so you can use spaces to indent lines for easier reading.

**Fields longer than one line**

The \ character at the end of a line indicates that the field continues onto the next line.

**Comments**    To comment a line, use the # character at the beginning of the line. The #
can be preceded by spaces. You cannot comment partial-lines.

**Example**    The following lines define a COMMAND action named Xrdb_Merge_Nocpp.
Many fields are omitted because the default values are appropriate.

```
ACTION              Xrdb_Merge_Nocpp
    DESCRIPTION     Merges a file into your resource database
    WINDOW-TYPE     NO-STDIO
    EXEC-STRING     xrdb -nocpp -merge %(File)Arg_1%
END
```

**Example**    The next action, IslandPaintOpenDoc, includes fields for custom action
bitmaps. It starts Island Paint$^{TM}$ with a file argument.

```
ACTION              IslandPaintOpenDoc
    L-ICON          Ipaint.l.bm
    S-ICON          Ipaint.s.bm
    WINDOW-TYPE     NO-STDIO
    EXEC-HOST       hpthere
    EXEC-STRING     /usr/bin/IslandPaint %(File)Arg_1"File to open:"%
END
```

**Example**    This next action definition creates a synonym action for Open_gif.

```
ACTION          Open_gif_file
    TYPE        MAP   Open_gif
END
```

## Summary of Fields for COMMAND Actions

**Required Fields**

| Field | Description (Default) |
|---|---|
| ACTION | Begins the definition, contains the action name. |
| END | Ends the definition. |
| EXEC-STRING | Contains the command to be executed. |

| **Icons** | Field | Description (Default) |
|---|---|---|
| | DESCRIPTION | A description of the action, displayed when the user obtains item help on the action icon. (No help.) |
| | L-ICON, S-ICON | Absolute path of the large (L) and small (S) action icons in File Manager. (`action.l` and `action.s`.) |

| **Optional Fields** | Field | Description (Default) |
|---|---|---|
| | ARG-COUNT | The number of file arguments the action can have. |
| | ARG-TYPES | Filetypes for which the action is valid (`*`, `0`, or `>1`). |
| | CWD | The current working directory. |
| | EXEC-HOST | The name of the host on which the application or command should be executed `%DatabaseHost%`, `%LocalHost%`, `%DataHost%`, *hostname*, "*prompt*"). |
| | TYPE | `COMMAND`. |
| | WINDOW-TYPE | The windowing support required to run the action (`PERM-TERMINAL`, `TERMINAL`, or `NO-STDIO`). |

# Summary of Fields for MAP Actions

| **Required Fields** | Field | Description (Default) |
|---|---|---|
| | ACTION | Begins the definition, contains the action name. |
| | END | Ends the definition. |

| **Icons** | Field | Description (Default) |
|---|---|---|
| | DESCRIPTION | A description of the action, displayed when the user obtains item help on the action icon. (No help.) |
| | L-ICON, S-ICON | Absolute path of the large (L) and small (S) action icons in File Manager. |

---

# ACTION field

- Begins the definition and specifies the action name.
- Required field for all actions.

The action name cannot contain spaces.

**Example**          `ACTION  MergeResources`

---

# ARG-COUNT field

- Specifies the number of arguments the action can accept.
- Optional for COMMAND and MAP actions.
- Defaults to * (zero or more arguments).
- Allowable values:
    - 0      Zero arguments.
    - >1    More than one argument.
    - *      Zero or more arguments.

    Values of 0 and >1 have precedence over *.

Use the ARG-COUNT field to enrich an action with different drop and double-click functionality. To do this, create two different actions with different ARG-COUNT values and EXEC-STRINGs.

Do not use the ARG-COUNT field to specify that an action prompt you for a file name when the action icon is double-clicked. This functionality is provided automatically by using the EXEC-STRING syntax `%(File)Arg_`$n$`"`*prompt*`"%` for file arguments.

**Precedence of actions using ARG-COUNT**    When two actions have the same name and different ARG-COUNT values, the one with the more specific ARG-COUNT (0 or >1) has precedence over *.

---

**Example**                    `ARG-COUNT >1`

**See Also**
- "To provide different double-click and drop function for an action" explains how to write an action that uses ARG-COUNT.
- "How HP VUE chooses between two actions with the same name" covers using ARG-COUNT and ARG-TYPES together.

# ARG-TYPES field

- Specifies the filetypes that can be used as data.
- Optional field for COMMAND or MAP actions.
- Defaults to * (all filetypes).

This field consists of a list of filetypes for which the action is valid. Items in the list are separated by commas. No spaces are allowed in the list of filetypes.

If this field contains only *, the action is valid for all filetypes.

You can create an action that behaves differently for different filetypes by creating multiple action definitions, each of which specifies a different filetype.

**Precedence of actions using ARG-TYPES**    When two actions have the same name and different ARG-TYPES values, the one with the more specific ARG-TYPE (not *) has precedence over *.

**Example**                    `ARGTYPES    XWD,TIFF`

**See Also**
- "How HP VUE chooses between two actions with the same name" covers using ARG-COUNT and ARG-TYPES together.

# CWD field

- Specifies the current working directory for the action.
- Optional for COMMAND actions.
- Cannot be used with MAP actions.
- Uses the syntax:

  [*hostname*:] */path/filename*

  If *hostname* is omitted, the database host (host containing the action) is used.
- Defaults to */HomeDirectory/* when an action is invoked by double-clicking.

**Default current working directory** If CWD is not set in the action definition:

- When the action is invoked by dropping a file argument, the current-working directory is the directory containing the file.

- When the action is invoked by dropping a directory argument, the argument becomes the current working directory.

- If the action has no file arguments:

  □ If it is started from a toolbox, the home directory is the current working directory.

  □ If it is started elsewhere, the directory containing the action icon becomes the current working directory.

  □ If the action starts a remote application, the home directory on the local system becomes the current working diretory if the directory is NFS-mounted onto the execution host. Otherwise, the home directory on the execution host becomes the current working directory.

**Example**         CWD $HOME/project1

# DESCRIPTION Field

- Provides item help for the action icon.
- Optional field.
- Defaults to NULL (nothing).
- Can be used in COMMAND and MAP actions.

The field can contain any characters but newline (⬚Return⬚). Line breaks are automatically inserted to wrap the text to fit in the item help box. Extra spaces between words are ignored.

**Example**

```
DESCRIPTION  Converts an xwd or tiff graphics file to pcl format.  Start \
             the action by dropping a graphics file on it or by         \
             double-clicking the action icon.
```

# EXEC-HOST field

- Specifies the host where the command will be executed.
- Optional for COMMAND actions.
- Cannot be used for MAP actions.
- Defaults to the host containing the action (`%DatabaseHost%`).

This field specifies which system in the network will execute the command.

| EXEC-HOST Value | Description |
|---|---|
| `%DatabaseHost%` | The host where the action definition resides. |
| `%LocalHost%` | The host where HP VUE is running. |
| `%DataHost%` | The host containing the first file argument. |
| *hostname* | The named host. |
| `%"prompt"%` | Prompts the user for the host name each time the action is invoked. |

**Example**                     EXEC-HOST %DataHost%

**See Also**            • "Using Actions to Run Remote Applications" in Chapter 22 explains how to
                          write actions that run remote applications.

# EXEC-STRING field

- Contains the command executed by the action.
- Required field for COMMAND actions.
- Cannot be used in MAP action.

**Syntax for arguments**     Keyword                          Meaning

%(File)Arg_$n$%                Substitute the $n$th file argument.
                                 If the argument uses the syntax
                                 *remote_hostname*: /*path*/*filename*,
                                 the argument is converted to the form
                                 /**nfs**/*remote_hostname*/*path*/*filename*.

%Arg_$n$%                      Substitute the $n$th file argument. Do
                                 *not* translate names from the form
                                 *remote_hostname*: /*path*/*filename*
                                 to the form
                                 /**nfs**/*remote_hostname*/*path*/*filename*.

%(File)Arg_$n$"*prompt*"%      Substitute the $n$th file argument. If
                                 it is not found, prompt the user for a
                                 file name and substitute the response.
                                 If the argument uses the syntax
                                 *remote_hostname*: /*path*/*filename*,
                                 the argument is converted to the form
                                 /**nfs**/*remote_hostname*/*path*/*filename*.

%(File)"*prompt*"%             Prompt the user for a file name
                                 and substitute the response. If
                                 the argument uses the syntax
                                 *remote_hostname*: /*path*/*filename*,

|  | the argument is converted to the form /nfs/*remote_hostname*/*path*/*filename*. |
| --- | --- |
| %(File)Args% | Substitute all of the remaining arguments. |
| %"*prompt*"% | Prompt the user and substitute the response. Do not use this syntax when prompting for a file name. |

**Syntax to a invoke shell**

| /bin/sh -c '*string*' | Invokes csh. The *string* can be the shell script or the filename of a shell script. |
| --- | --- |
| /bin/ksh -c '*string*' | Invokes ksh. The *string* can be the shell script or the filename of a shell script. |

**See Also**

- "Environment variables in definitions" in Chapter 19 covers using environment variables for actions.
- "Building the Execution String for a COMMAND Action" explains how to write an execution string.
- "To provide shell capabilities in an action" covers writing execution strings that invoke shells.

---

# L-ICON and S-ICON fields for actions

- Specify the icons for the action.
  - □ L-ICON: displayed in File Manager views By Name and Icon.
  - □ S-ICON: displayed in File Manager views By Name and Small Icon.
- Default: action.l, action.s

The value for the L-ICON and S-ICON fields can be:

- The absolute path.

- A filename. The system searches the icon search path. If the action or filetype is defined on a remote system, the remote system's search path is used.

**Icon file names**     There are no name requirements. You can use .bm or .pm extensions to distinguish between bitmaps and pixmaps.

**Icon sizes**     Icons can be any size. These are the default sizes.

| Icon Type | Icon Size (Width×Height) |
|---|---|
| L-ICON | 32 × 32 pixels |
| S-ICON | 16 × 16 pixels |

**Example**

```
L-ICON frame.l.pm
S-ICON frame.s.pm
```

**See Also**

- "Where to put icon files" in Chapter 16 explains where the system looks for icons when you use a filename for L-ICON or S-ICON.

# The TYPE field

- Describes whether the action contains a command or is mapped to another action.
- Allowable values: COMMAND or MAP.
- Defaults to COMMAND.

**COMMAND type**     A COMMAND-type action is a command that will be executed. The action definition for a COMMAND action must include the execution string for the command (EXEC-STRING field).

**Example**

```
ACTION Remote_PCL_Print
   TYPE COMMAND
   EXEC-HOST   hpthere
   EXEC-STRING /bin/sh -c 'cat %(File)Arg_1"File to convert:"% | \
               /usr/bin/X11/xwd2sb | \
               /usr/bin/pcltrans -s -R -e%"magnification"% > \
               %(File)"Destination:"%'
   END
```

**MAP type**

```
ACTION PRINT
      ARGTYPES    XWD
      MAP         PRINT_XWD
      . . .

ACTION PRINT
      ARGTYPES    TIFF
      MAP         PRINT_TIFF
      . . .
```

# The WINDOW-TYPE Field

- Specifies the windowing support required by the command.
- Optional for COMMAND actions.
- Cannot be used with MAP actions.
- Defaults to PERM-TERMINAL.

PERM-TERMINAL    The command must be executed in a permanent terminal emulation window (the window is not closed when the command terminates). Use with commands that take some input, produce some output, then terminate.

TERMINAL    The command must be executed in a terminal window. The window is closed when the command is exited. Use with full-screen commands (for example, `vi`).

NO-STDIO    No windowing support is required. Use with applications and X clients create their own windows, or with commands that produce no displayed output (for example, print commands).

**Example**    `WINDOW-TYPE NO-STDIO`

**Reference: Action Definitions**

22

# Networking and
# Distributed Computing

# Networking and Distributed Computing

HP VUE is designed to work in a highly networked environment. This is especially the case with HP VUE actions, which are designed to provide "network transparency" for the user.

- An action defined locally can run an application on a remote host, or application server.
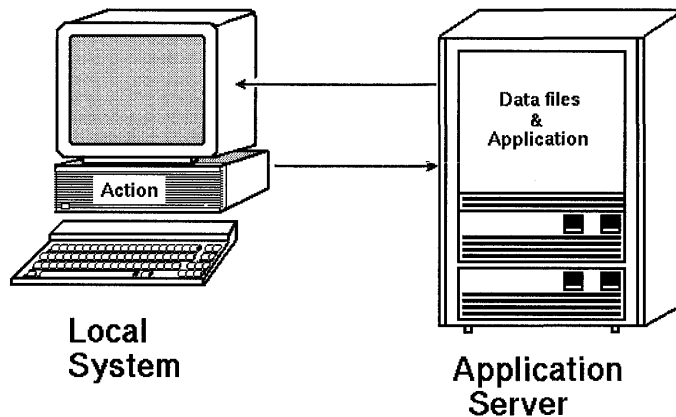


A local action can run a remote application.

- The data can be located on a remote file server. For example, an action defined locally can use an application on one remote system, which in turn uses data on another remote system.

**File server**



**Local System**

**Application Server**
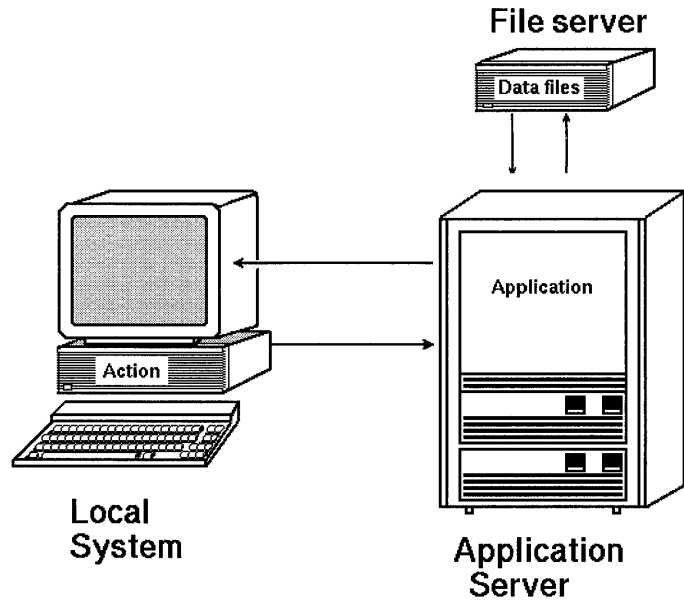
**A remote application can access data from a file server.**

- An action can be defined remotely and used just as though it were defined on your system. Thus, you can use a host in your network as an **action server**.
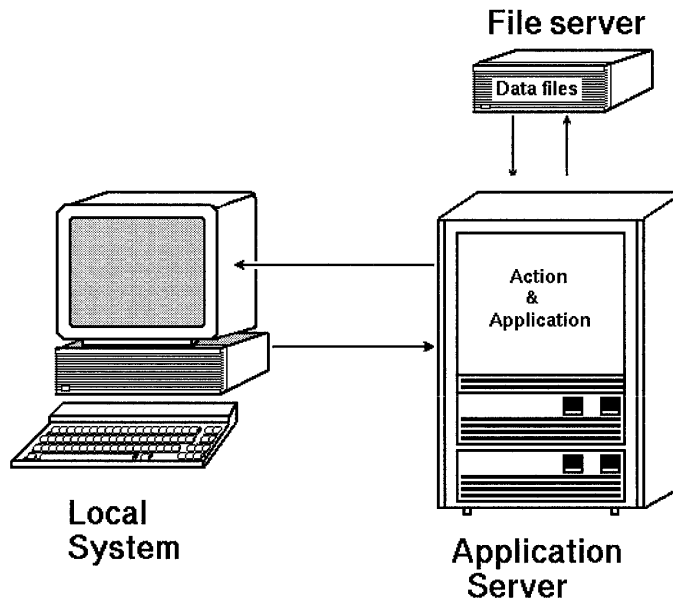


An action can be defined on a remote system.

Thus, when you discuss networked actions, there may be several hosts involved:

local host          The host from which the action is invoked.

execution host   The host where the application invoked by the action runs.

data host          The host where the data is located.

database host   The host where the action is defined.

In X-terminal configurations, an additional host runs the X server.

# Using Actions to Run Remote Applications

In order for a local action to run a remote applicatation:

- The local and remote hosts must be properly configured.

- The action must be writen to invoke the application on the desired host. You must create the action manually by editing a *.vf file; you cannot use the Create Action utility.

---

**NOTE**

You can write actions that run remote applications by using **remsh** in the command line. However, actions using **remsh** will not take advantage of automatic xhosting mechanism provided by actions written using the EXEC-HOST field to specify the remote host.

---

# To configure the local host for remote execution

- HP VUE must be installed and running.

- Configure the hostname properly:

  □ *hostname* and network address must be specified in **/etc/hosts**,

  □ *Or*, the operating system name server mechanism may be used.

- Install and configure the standard **inetd** process. (This should be done automatically when you install HP VUE. See the **inetd(1m)** man page for more information).

- Provide NFS access to the **/tmp** directory of the execution host. (Mount **/** if **/** and **tmp** are on the same disk.) These file systems must be mounted

under **/nfs/***execution_host***/tmp** (for **/tmp**), or **/nfs/***execution_host* (for
**/**). (See "To NFS-mount a remote file system (HP-UX only)".)

- Optional: you can specify the Terminal Emulator used for terminal-based
  applications invoked using an action. (The default depends on the
  operating system of the machine containing the action definition; **hpterm**
  for HP-UX systems and **xterm** for other operating systems.) Use the
  resource:

      *remoteTerminals:   *host_name*:*term_path* [,  ... ]

  where *term_path* is the absolute path to the terminal executable on the
  remote system.

**Automatic xhosting**      When the action is executed, HP VUE automatically provides xhost
permission on the display host for the execution host.

**Example**      The following resource sets terminal emulators for remote applications on two
systems.

      remoteTerminals:  cv_sun:/usr/bin/xterm,cv_dec:/usr/bin/X11/decterm

# To configure the remote execution host

- Install all of HP VUE, or just the BMS fileset and **/usr/vue/bin/vueexec**.
- Configure the hostname properly:
  □ *hostname* and network address must be specified in **/etc/hosts**,
  □ *Or*, the operating system name server mechanism may be used.
- Install and configure the standard **inetd** process. (This should be done
  automatically when you install HP VUE. See the **inetd(1m)** man page for
  more information).
- Edit **/usr/adm/inetd.sec** to allow the local host (the host invoking the
  action) to connect to the SPCD service. Use the syntax:

      spc allow  ...  *local_host*  ...

- Provide the user (the person invoking the action) with a valid login (username and password). The user should have the same user-id (uid) as on the local system.

- If the data resides elsewhere, provide NFS-access to the data.

**The current working directory for remote applications**

When an action executes a remote application, it sets the current working directory for execution of the application as follows:

- If the user's local home directory is NFS-mounted to the execution host, it becomes the current working directory.

- If the execution host cannot access the user's local home directory, the user's home directory on the execution host becomes the current working directory. In addition, the following error message is written to /*HomeDirectory*/`.vue/errorlog`:

      SPCD: Can't connect to *hostname*.  No mount point.

For example, suppose you have an action that executes Framemaker$^{TM}$ on remote host `hphost`. If your local home directory is mounted on `hphost`, the application will behave as though you ran it from your local home directory. Otherwise, the application behaves as though you started it from your home directory on `hphost`.

**See Also**

- "To limit execution access to the system" covers restricting access to the BMS and SPCD.
- "To configure the execution host to access remote data" desribes how provide NFS-access to data.

# To specify the remote execution host

- In the action definition, use the EXEC-HOST field in the action definition to specify the host where the command should be executed.

When the EXEC-HOST field is omitted, it is assumed to be the database host (the host containing the action definition).

| EXEC-HOST Value | Description |
| --- | --- |
| %DatabaseHost% | (The default). The host where the action is defined. |
| *hostname* | The named host. Use this value for environments in which the action should always be invoked on one particular host. |
| %LocalHost% | The host where the action is invoked. |
| %DataHost% | The host containing the data. If the data is on more than one host, the first file argument is used. |
| %"*prompt*"% | Prompts the user for the host name each time the action is invoked. |

**Example**

```
EXEC-HOST   hpthere                              Execution host is hpthere
EXEC-HOST   %"Host containing application:"%     Prompts for host
EXEC-HOST   %DataHost%                           Execution host is same as data
                                                 host.
```

# Example: Creating a system-wide action for a remote application

The application Framemaker™ resides on remote host **hpthere**. Generally, it is started from local host "hphere" by executing the command:

```
remsh hpthere -n /usr/bin/maker -display hphere:0
```

**Create the database file**

1. Create the file **/usr/vue/config/types/FrameMaker.vf**. Open the file for editing.

**Define the action**

2. Start the action definition:

```
ACTION          FrameMaker
```

3. Add a field to provide item help on the FrameMaker application icon in the toolbox.

```
DESCRIPTION  Double-click to start FrameMaker.
```

4. Specify that the application creates its own window and therefore does not need to run in a terminal emulator.

```
WINDOW-TYPE    NO-STDIO
```

5. Specify that the application runs on host `hpthere`.

```
EXEC-HOST      hpthere
```

6. Add the command executed by the application.

```
EXEC-STRING    /usr/bin/maker
```

7. End the definition with the END field. The entire definition looks like this:

```
ACTION         FrameMaker
  DESCRIPTION  Double-click to start Frame.
  WINDOW-TYPE  NO-STDIO
  EXEC-HOST    hpthere
  EXEC-STRING  /usr/bin/maker
END
```

**Define the filetype**

8. Add a few emply lines for readability and start the filetype definition:

```
FILETYPE       Frame_Data
```

9. Specify the file naming criteria for the filetype:

```
FILE-PATTERN  *.fm
```

10. If necessary, create File Manager icons for the filetype in the directory `/usr/vue/icons`. Then, specify the icons in the definition:

```
L-ICON         frame.l
S-ICON         frame.s
```

11. Use the Actions field to specify the contents of the Action menu.

```
ACTIONS        FrameMaker
```

By adding this field, you will be able to start Framemaker by double-clicking a data file.

12. End the definition. The entire definition looks like this:

```
FILETYPE       Frame_Data
  FILE-PATTERN *.fm
  L-ICON       frame.l
  S-ICON       frame.s
  ACTIONS      FrameMaker
END
```

**Update the database and toolbox**

13. Save the file `FrameMaker.vf`.

14. Double-click ReloadActions in the System_Admin subdirectory of the General Toolbox.

15. Open the Applications subdirectory in the General Toolbox and create an executable file named FrameMaker.

**See Also**

- "Where to put icon files" in Chapter 16 explains the directories you should use to store icons.

## How environment variables are handled during remote execution

Environment variables are acquired as follows:

1. The local environment is transferred to the remote execution host.

2. The local file `/usr/softbench/config/softenv` is read. Variables defined there override previous values.

3. `$HOME/.softenv` is read. Variables defined there override existing values.

4. The environment is further modified according to `/usr/softbench/config/softenv`, then `$HOME/.softenv` on the remote host.

**The DISPLAY variable**
If DISPLAY is set to `local:`$n$, `unix:`$n$, or `:`$n$, then it is translated to *hostname*:$n$ before being placed in the remote environment.

**Variable expansion**
Expanding a variable substitutes its value:

- To expand the variable when the database is read, use the form $variable.

- To expand the variable when the execution string is executed, use the form `echo \`$variable. This must be done within a shell.

**Example**         The following execution string does not expand the environment variable
until the action is run. Thus, if the action invokes the application on a remote
system, $HOME will be the value of HOME on the remote system.

```
EXEC-STRING /bin/sh -c 'cat echo "\$HOME"/%(File)Arg_1"Filename:"%'
```

**See Also**        • "Environment variables in definitions" in Chapter 19.

# Using Actions to Access Remote Data

Remote data is data that resides elsewhere than on the execution host.

In order to use remote data:

- The execution host must be properly configured to find the data.

- If the data is elsewhere than on the execution host, the user must specify the host where the data is located.

**NOTE**

"Remote data" is in reference to the execution host. When an action executes a remote application using local data, that data is considered remote data, in that it must be NFS-accessible to the execution host.

# To configure the execution host to access remote data

- Mount the data host to the execution host. The mount point for the remote file system must be:

    /nfs/*data_host*

- If the data host is in a different domain than the execution host, you must use the fully-qualified domain name of the data host (for example, `pronto.dom.hp.com`).

Each user must have the same user id on all machines accessed via NFS.

> **NOTE**
>
> It is recommended that all file systems on the remote data host be mounted to the execution host.

**See Also**

• "To NFS-mount a remote file system (HP-UX only)".

## To specify remote data

• When you drop a file on an action icon, the data host is known automatically.

• When prompted for the filename, use the syntax:

  *data_host* : **/** *path* **/** *filename*

  If *data_host* is remote, then *data_host*: is replaced with **/nfs/***data_host***/**. For example, **hpcvxba:/tmp/data1** becomes **/nfs/hpcvxba/tmp/data1**.

• The action definition can specify the current working directory using the CWD field. If you then respond to a prompt with the filename or a relative path, the action will use the current working directory.

**Example**

Suppose the following action is defined and invoked from local host **here**. The action will execute the **xsetrootgif** command locally. If the user supplies a relative path in reponse to the prompt, the command will look for the data in **/usr/local/gif** on host **paecom**.

```
ACTION        Use_gif_Backdrop
  CWD         paecom:/usr/local/gif
  EXEC-STRING /usr/local/bin/xsetrootgif -gif %(File)Arg_1"gif file:"%
END
```

**Example**

Like the previous example, this action is invoked from host **here** and gets its data from **paecom** if the user specifies a relative path. However, the application is located on host **ellcom**.

```
ACTION          Edit_QS_document
  CWD           paecom:/QSproject/documents/
  EXEC-HOST     ellcom
  EXEC-STRING   /usr/local/bin/IslandWrite %(File)Arg_1"Document:"%
END
```

# Configuring diskless clusters for remote data access

**NOTE**

This discussion applies only to operating systems that support diskless clusters.

- Only a cluster server can export its file system for NFS mounting. Therefore, the cluster server's file system must be mounted to the execution host.

- If the execution host is another machine in the same cluster, one of the following configurations must be used:

  □ Use a symbolic link to mimic NFS mounting the file system onto itself. To create this symbolic link, execute:

    **ln -s / /nfs/***execution_cnode*

- The cluster server's file system can be mounted on to itself. However, this method is not recommended.

**Example**

A cluster with cluster server **hphusker** is to be configured so that its nodes can use actions that specify the cnode **hpabcd** as the execution host. The root directory of **hphusker** should be linked as follows:

```
ln -s / /nfs/hpabcd
```

# Specifying remote data on a diskless cluster

For context-dependent files:

- When the user supplies the file location in response to a prompt, the cluster server should be used as the data host.

  □ To specify data on the cluster server, use:

    *cluster_server*:*/path/filename*

  □ If **/nfs** is a context-dependent directory, to specify data on a particular node, use:

    *cluster_server*:*/path/filename+/node*

- If the execution host is another machine in the same cluster, and the cluster server's file system is NFS-mounted to itself, the cluster server should be the data host.

- If the execution host is another machine in the same cluster, and symbolic links are used instead of NFS mounds, context-dependent files are accessed using the context of the execution host rather than the server.

**Example**

This location specifies data on remote cluster server **hphusker**.

```
hphusker:/etc/inittab
```

**Example**

This location specifies data for node **hpthere** located in remote cluster **hpthere**.

```
hpthere:/etc/inittab
```

**Example**

Cluster server **hphusker** uses symbolic links, rather than NFS mounts, to provide remote execution within the cluster. Suppose **app1** is executing on **node1**, and **app2** is executing on **node2**. Then, data specified as:

hphusker:/etc/inittab

refers to **hphusker/etc/inittab+/node1** on **node1**, and **hphusker/etc/inittab+/node2** on **node2**.

**See Also**

- The **cdf(4)** man page explains how to access context-dependent files using an explicit path

# Importing and Exporting Actions

HP VUE allows local systems to access actions defined on remote hosts. The remote host containing the action definition is called the **database host** or the **action server**. The database host "exports" its actions to other systems, which, in turn, "import" the action.

For example, a system administrator installing an application on a remote execution host may find it desirable to define the action for the application on the same host. Since that action can be made available to the other systems on the network, it is not necessary to copy the definition from system to system.

In a diskless cluster, actions imported by the cluster server are available to all cnodes.

In order for a system to use an action defined remotely:

- The database host must be configured to export the actions to other hosts.

- The local system must be configured to import the remote actions.

## To configure the database (exporting) host

1. Place the files (the **\*.vf** files) containing the definitions for the actions to be exported in **/usr/vue/config/export**.

2. Create action icons for the actions by placing an executable file for each action in **/usr/vue/config/export/tools** or one of its subdirectories.

3. Edit **/etc/export** to include all hosts who will be using the actions. They must be allowed to NFS-mount **/tmp** and **/usr/vue/config/export**.

4. Edit **/usr/adm/inetd.sec** to allow all the hosts using the actions to use the database host's spc.

5. Perform any additional configuration for remote application and data access:

- If the application and data reside on the database host, no additional configuration is necessary.

- If the application resides elsewhere, then the database host and execution host must be configured for remote execution.

- If the data resides elsewhere than with the application, then the database host must be configured for remote data access.

**See Also**
- "Using Actions to Run Remote Applications" covers configuring the hosts so that actions can execute remote applications.
- "Using Actions to Access Remote Data" covers configuring the hosts so that actions can use remote data.

# To configure the local (importing) host

1. NFS-mount the file system of the database host to provide access to **/tmp** and **/usr/vue/config/export**. The mount point must be:

   **/nfs/**database_host

2. Create the directory **/usr/vue/config/import** if it does not yet exist.

3. Edit **/usr/vue/config/Xsession**. Add or modify the line that sets the VUEACTIONREMOTEHOSTS environment variable, which specifies the remote hosts whose actions will be imported:

   **VUEACTIONREMOTEHOSTS=**remote_host**,**remote_host . . .

4. Add the remote host to the Network Toolbox (see "To add a host to the Network Toolbox").

5. Log out and back in.

**See Also**
- "To NFS-mount a remote file system (HP-UX only)" explains how to set up NFS mounts.
- "To reread Login Manager configuration files" in Chapter 12 explains how to reread **Xsession**.

## To add a host to the Network Toolbox

1. Configure the local host to import the actions (see "To configure the local (importing) host").

2. Configure the remote host to export its actions (see "To configure the database (exporting) host").

3. Create a link between the Network Toolbox and the remote host (database host) toolbox:

   ```
   ln -s /nfs/database_host/usr/vue/config/export/tools \
          /usr/vue/config/import/database_host
   ```

   This creates the host icon in the top level of the Network Toolbox.

## Example: Exporting and Importing an Action

The application Interface Architect on host **appserver** is used by numerous systems in the network. It is desirable to create one action for the application, and export that action to all the systems that use it. Thus, **appserver** becomes both the application server and the database (action) host.

**On host 'appserver'**

1. Create the file **/usr/vue/config/export/architect.vf** and define the action:

   ```
   ACTION           Architect
      WINDOW-TYPE   NO-STDIO
      EXEC-STRING   /usr/architect/bin/architect
   END
   ```

2. Create an action icon by creating an executable file **/usr/vue/config/export/tools/Architect**.

3. Edit **/etc/export** to include all the hosts that will use the Architect action.

4. Edit **/usr/adm/inetd.sec** to include all the hosts that will use the Architect action.

**On each importing host**

1. NFS-mount the file system of **appserver**.

2. Execute the following command to add the imported actions to the Network Toolbox on the local hosts.

```
ln -s /nfs/appserver/usr/vue/config/export/tools /usr/vue/config/import/appserver
```

3. Edit **/usr/vue/config/Xsession** to add **appserver** to the local host's search path using the environment variable VUEACTIONREMOTEHOST:

**VUEACTIONREMOTEHOSTS=appserver,** *another_remote_host* ...

4. Log out and back in.

# Configuring Network Security

There are several security considerations in HP VUE, determining:

- Which users can unlock the display.
- Which systems can NFS-mount local files.
- Which systems can run applications using actions.
- Which systems can access the local display.

## To configure who can unlock the display

- Use the Session Manager **keys** resource to specify who can unlock the display:

      **vuesession*keys:** *user*[,*user* ... ]

The default is the login user and root. Regardless of the value of the resource, root can always unlock the display.

**Example**

This resource specifies the users who can unlock the display.

      vuesession*keys:    kreta,ellen,dex,anna

## To limit access to the local file system (NFS security)

- Edit **/etc/export** to provide NFS security. It contains a list of the remote systems that are permitted to NFS-mount local disk volumes:

      *file_system    remote_host*

**Example**

This entry in **/etc/export** permits remote host **hpthere** to mount volume **/doc**.

```
/doc    hpthere
```

# To limit execution access to the system

1. Edit **/etc/services**, if necessary. The **spc** service must be listed.

2. Edit **/usr/adm/inetd.sec** to include the hosts that should have access to the local SPCD. You should keep the access list as small as possible, since these services allow access to the machine by anyone connecting to the port. To add a complete network, include the network or subnet component of the address.

The Sub-Process Control Daemon (SPCD or softspcd) supports remote execution. When an action on a local host invokes an application on a remote host, the local HP VUE sends a message to the remote SPCD specifing the execution string for the application. For security reasons, the SPCD does not allow root to perform remote execution; root can't perform the file-based authentication over NFS.

By default, the **mserve** (Message Server) and **spc** (Sub-Process Control) services provided by HP VUE are restricted in **/usr/adm/inetd.sec** to:

• The host name of the system.

• The nodes in the cluster, if it is a cluster system. On a diskless cluster, **/usr/adm** is a context-dependent directory. Thus, **/usr/adm/inetd.sec** is not shared automatically within the cluster.

> **NOTE**
>
> If you change the host name or internet address of the machine, make sure that **inetd.sec** contains the new host name.

**Example**

The following lines in **/usr/adm/inetd.sec** specify the hosts permitted to access the local host's BMS: hostA, hostB, hostC, and all hosts on subnet 192.6.36.

```
mserve   allow   hostA hostB hostC 192.6.36.*
spc      allow   hostA hostB hostC 192.6.36.*
```

**See Also**

• The **inetd.sec(4)** man page contains additional information.

# To limit remote access to a system's display

The mechanism for restricting access to the local display depends on whether the display connection is requested by an action or by some other mechanism.

**Display access with actions**

When an action executes a remote application, the application server is automatically given permission to connect to the local display (the **xhost** command is executed automatically).

To turn off automatic authorization, use the resource

[*client*]*autoXhosting: false

**Display access by other mechanisms**

When a remote application is started in ways other than by actions, the remote host must have explicit permission to connect to the local display. There are two ways to provide (and limit) remote access to a local display:

• Authorization by host name. This is the default authorization system used by HP VUE. It involves maintaining a list of all the remote hosts that have permission to use the local server to display clients. Authorization by host name is implemented by the **/etc/X*.hosts** file, where * is the display (for example, **X0.hosts** for display 0) or by the **xhost** command.

• Authorization by user. This feature is provided by the R4 and R5 X servers.

# Configuring Access to Remote Data

In HP VUE, remote data files are accessed using NFS mounts.

Once the remote file systems are mounted, File Manager provides theses mechanisms for accessing remote files:

- Using the Remote Systems directory.
- Using fully-qualified file names.

## To add a host to the Remote Systems (nfs) directory

1. Configure the remote host to allow NFS-access by the local host (`/etc/export`).

2. NFS-mount the remote host's file system to the local host. The mount point for the remote file system must be **/nfs/***remote_hostname*.

The user can can browse files on the remote system by using the *hostname*:/*path* to specify a directory, or by choosing Remote Systems from the File Manager Directory menu. The Remote Systems directory is /nfs.

## To NFS-mount a remote file system (HP-UX only)

File Manager and actions use NFS mounts to provide easy access to remote data files. Usually, the remote system files are mounted under **/nfs/***hostname*.

> **NOTE**
>
> This procedure applies to HP-UX systems only.

**On the data host**  The data host is the system whose files are being accessed.

1. Log in as root.

2. Run SAM. (If the system is running HP VUE, you can run SAM using the action in the System_Admin subdirectory of the General Toolbox.)

3. In SAM, select:

   - For HP-UX 9.0:

     ```
     Networking/Communications ->
         Networked File Systems (NFS) ->
             Local File Systems Exported ->
     ```

   - For HP-UX 8.0:

     ```
     Networking/Communications ->
         LAN Hardware and Software (Cards and Services) ->
             NFS (Network File System) Configuration ->
     ```

4. Choose Add from the Actions menu.

5. Follow SAM's on-line help to provide permission for the local system to mount the file system on the data host. The local host *must* have permission to mount **/tmp**. It also must be able to mount directories containing the data files it needs to access.

**On the local system**

1. Log in as root.

2. Double-click Sam in the System_Admin subdirectory of the General Toolbox to run Sam.

3. In SAM, select:

   - For HP-UX 9.0:

```
Networking/Communications ->
    Networked File Systems (NFS) ->
        Remote File System Mounted ->
```

- For HP-UX 8.0:

```
Networking/Communications ->
    LAN Hardware and Software (Cards and Services) ->
        NFS (Network File System) Configuration ->
```

4. Select Add Remote Directory from the Actions menu.

5. Follow SAM's on-line help to mount the data host's file system under
   /nfs/*hostname*.

**See Also**

- The documentation on Networked File Systems contains additional
  information about NFS mounting.

# If you try to access data not correctly mounted

If the user tries to access a remote directory that is not correctly
NFS-mounted, an error message is displayed telling the user that
the attempt failed. In addition, the following message is written to
/*HomeDirectory*/.vue/errorlog:

```
vuefile: Can't connect to hostname.  No mount point.
```

# Changing the System Hostname

If your system uses **/usr/adm/inetd.sec** to enforce security by listing specific hosts that are allowed, then the entry for your local host must be changed to your new host name. For example, if you change the name of your host from **oldname** to **newname**, then you must alter lines of the form:

```
mserve   allow    oldname host2  ...
spc      allow    oldname host2  ...
```

by replacing **oldname** with **newname**.

# Providing Networked Sessions

A **networked session** is managed across multiple systems. The networked session files are located on one system, called the **session server**. The files are distributed to other systems using NFS mounts and symbolic links.

For example, suppose you commonly use one of three systems—systemA, systemB, and systemC. If you provide a networked session:

- All the systems share a single home directory.

- A session saved when you log out on systemA will be restored when you log into systemB or systemC.

- Personal customizations made on one system are distributed to other systems—for example, personal filetypes, actions, and Front Panel.

**See Also**
- "Customizing Sessions" in Chapter 15 describes other aspects of configuring sessions.

## To configure the session server

1. Provide a valid login and password for each user.

2. Edit **/etc/export** to list all the hosts using the networked session.

3. For each user:

   a. If a display-dependent session exists , remove or move it.

   b. Log in and out to create a display-independent session
      (/*HomeDirectory*/.vue/sessions).

A display-dependent session is a session stored in a subdirectory of /*HomeDirectory*/.vue/*display*; for example, /users/george/.vue/hpaaaa:0/current).

**See Also**

- "To limit access to the local file system (NFS security)" describes
  /etc/export.
- "To create a display-independent session" in Chapter 28.

## To configure the local systems

1. Provide NFS access to the home directory on the session server (see "To
   NFS-mount a remote file system (HP-UX only)").

2. Move the user's home directory to /*HomeDirectory*.old/

3. Create a new home directory that is a link to the home directory on the
   session server. Use the home directory location specified in /etc/passwd

4. Merge any customizations in /*HomeDirectory*/.old into the new linked
   home directory.

These steps create a single home directory for the user, regardless of which
system the user logs into.

## Example: Configuring networked sessions

System **hpaa** is the session server for two users.

| *User* | $HOME *on* hpaa | $HOME *on* hpbb | $HOME *on* hpcc |
|--------|-----------------|-----------------|-----------------|
| ellen  | /users2/ellen   | /users1/ellen   | /users/ellen    |
| kreta  | /users3/kreta   | /users1/kreta   | /users/kreta    |

**On session server ('hpaa')**

1. Provide a valid login and password for users `ellen` and `kreta`.

2. Edit `/etc/export` to include hosts `hpbb` and `hpcc`.

   ```
   /users2  hpbb hpcc
   /users3  hpbb hpcc
   ```

3. If present, remove or rename these display-dependent sessions:

   - `/users2/ellen/.vue/hpaa:0`.
   - `/users3/kreta/.vue/hpaa:0`.

4. Create these display-independent sessions by having each user log in and out.

   - `/users2/ellen/.vue/sessions`.
   - `/users3/kreta/.vue/sessions`.

**On system 'hpbb"**

1. NFS-mount the home directories of `hpaa` to create:

   - `/nfs/hpaa/users2`
   - `/nfs/hpaa/users3`

2. Move the home directories to a temporary location:

   ```
   mv /users1/ellen /users1/ellen.old
   mv /users1/kreta /users1/kreta.old
   ```

3. Create new home directories that are links to the home directories on on `hpaa`.

   ```
   ln -s /nfs/hpaa/users2/ellen /users1/ellen
   ln -s /nfs/hpaa/users3/kreta /users1/kreta
   ```

4. Copy the customizations you want to preserve from the subdirectories of `/users1/`*login_name*`.old` to the new directories.

**On system 'hpcc"**

1. NFS-mount the home directories of `hpaa` to create:

   - `/nfs/hpaa/users2`
   - `/nfs/hpaa/users3`

2. Move the home directories to a temporary location:

   ```
   mv /users/ellen /users1/ellen.old
   mv /users/kreta /users1/kreta.old
   ```

3. Create new home directories that are links to the home directories on `hpaa`:

```
ln -s /nfs/hpaa/users2/ellen /users/ellen
ln -s /nfs/hpaa/users3/kreta /users/kreta
```

4. Copy the customizations you want to preserve from the subdirectories of **/users/***login_name*.**old** to the new directories.

# Customizing HP VUE Lite

# Customizing HP VUE Lite

The major configuration differences beween regular HP VUE sessions and HP VUE Lite are:

- Session Manager uses different configuration files for HP VUE Lite, and session is started from an editable startup script.

- HP VUE Lite has a different Front Panel:

  □ The Front Panel terminal button has a configurable subpanel for starting other Terminal Emulators.

  □ The Tools control for HP VUE Lite can be configured to start the application of you choice. The Tools subpanel provides a place to put controls for starting othre applications.

  □ Since HP VUE Lite does not use File Manager, there are no drop zone controls (such as the Printer and Trash Can in regular HP VUE sessions).

# Administering HP VUE Lite Sessions

HP VUE Lite sessions are started from a session startup script, which is responsible for starting clients. Thus, the same clients are started at the beginning of each session (unlike regular HP VUE sessions, which restores clients running at the end of the previous session).

Resources are saved and restored from one HP VUE Lite session to another.

## To make HP VUE Lite the only session available

1. Log in as root.
2. Set the **vuelite** resource to true in **/usr/vue/config/Xconfig**:

    **Vuelogin*vuelite: true**

3. Reread the Login Manager configuration files.

When the **vuelite** resource is set to true, the user is unable to log into a regular HP VUE session on that system.

**See Also**
- "To reread Login Manager configuration files" in Chapter 12 covers how to reread **Xconfig**.

---

# To customize HP VUE Lite session startup

**System-wide**

1. Log in as root.

2. Edit **/usr/vue/config/sys.ses.lite** to contain a command line for each client you want to start. All the clients must be started in the background (each command line must end with **&**).

This file is used if the user has not created a personal startup script */HomeDirectory/*.vue/sessions/lite/vue.session.

**Personal**

1. If *HomeDirectory/*.vue/sessions/lite/vue.session does not already exist, create it by copying **/usr/vue/config/sys.ses.lite**. Give the new file write permission.

2. Edit **vue.session** to include a command line for each client you want to start. Run the clients in the background (each command line must end with **&**).

**Example**

The following lines start two Terminal Emulator windows: a console in a workspace named One, and an additional terminal in workspaces One and Two.

```
hpterm -C -ls -xrm '*workspaceList: One' &
hpterm -ls -name Terminal -xrm '*workspaceList: One Two' &
```

---

# To start clients in different workspaces

- If the client has the −**workspaceList** option, use this syntax for the command:

  *command* -**workspaceList** *workspace_name* [*workspace_name* ... ]

- If the client lacks the −**workspaceList** option but has the −**xrm** option, use this syntax for the command:

---

*command* -xrm '*workspaceList:* *workspace_name workspace_name* ... ]'

If workspaceList is not specified, the application window is placed in the workspace displayed at the end of the previous session.

**Example**

The following command line starts an xterm window in two workspaces named project1 and extra space.

```
xterm  -xrm '*workspaceList: project1 "extra space"' &
```

# To set resources for the first HP VUE Lite session

1. Before running the first HP VUE Lite session, log into No Windows mode or a fail-safe session.

2. Log in as root.

3. Edit /usr/vue/config/sys.res.lite

At logout, resources are saved to /*HomeDirectory*/sessions/lite/vue.resources. They are restored at the next session.

Resources are saved and restored separately for HP VUE lite sessions and regular HP VUE sessions. Therefore, changes you make to an HP VUE lite session have no effect on the next regular session.

**See Also**

• "Syntax of resource specifications" in Chapter 25

# To set resources during an HP VUE Lite session

Change the resources using:

• Style Manager.

• *Or*, manually setting the resources.

Resources changed during an HP VUE Lite session are saved at logout, and restored at the beginning of the next HP VUE Lite session.

**See Also**

- Chapter 25.

# How HP VUE Lite sessions are started

HP VUE Lite does not use the same Session Manager as regular HP VUE sessions. In most cases, logging in and out is faster with HP VUE Lite.

When the user has been validated, HP VUE Lite Session Manager executes the shell script **/usr/vue/bin/vuelitesession**. The script:

1. Loads resources from the appropriate resource file:

   - If **/***HomeDirectory***/.vue/sessions/lite/vue.resources** exists, it is used.

   - Otherwise, the system default **/usr/vue/config/sys.res.lite** is used.

2. Executes **vuehello**, which displays the greeting screen.

3. Starts the color server.

4. Starts the window manager (**vuewm**).

5. Executes the HP VUE lite session startup script to start applications:

   - If executable file **/***HomeDirectory***/.vue/sessions/lite/vue.sessions** exists, it is used.

   - Otherwise, the system default **/usr/vue/config/sys.ses.lite** is used.

When the user logs out:

1. The Window Manager halts.

2. Session resources are saved into **/***HomeDirectory***/.vue/sessions/lite/vue.resources**.

# Customizing HP VUE Lite Terminal Emulators

You can customize Terminal Emulators by:

- Switching to a different terminal emulator client.
- Customizing the Terminal button and Terminals subpanel.

## To use a different terminal emulator in HP VUE Lite

1. If /*HomeDirectory*/.vue/types/user-prefs.vf does not exist, create it by copying /usr/vue/types/user-prefs.vf. Give the new file write permission.

2. Remap the Terminal action to a different action.

3. Choose Restart Workspace Manager from the Workspace menu.

If the actions to which the remote terminals are mapped do not already exist, you will have to create them.

**Example**

The following lines in *HomeDirectory*/.vue/types/user-prefs.vf set the default terminal to **xterm**. (**Xterm** is the name of a built-in action.)

```
ACTION Terminal
        TYPE        MAP        Xterm
END
```

**See Also**

- "Creating a Simple Action" in Chapter 19 explains how to use Create Action to create new actions.

# To customize the remote terminal buttons

1. If */HomeDirectory/*.vue/types/user-prefs.vf does not exist, create it by copying /usr/vue/types/user-prefs.vf. Give the new file write permission.

2. Remap the the RemoteHpterm or RemoteXterm actions to different actions.

3. Choose Restart Workspace Manager from the Workspace Menu.

If the actions to which the remote terminals are mapped do not already exist, you will have to create them.

**See Also**

- "Creating a Simple Action" in Chapter 19 explains how to use Create Action to create new actions.

# To add a control to the Terminals subpanel

1. If the change is personal rather than system wide, create a personal **fp.terminal** file (see "To create a personal Terminals subpanel configuration file").

2. Open **fp.terminal** and add a control to the subpanel box.

```
BOX TerminalSubpanel
{
    TYPE       subpanel
      :
    CONTROL   new_control_name
}
```

3. Add the definition for the new control to the file.

4. Save the file.

5. Create the terminal action.

6. Choose Restart Workspace Manager from the Workspace menu.

- "Example: Adding a terminal to the Terminals subpanel" is a step-by-step example.
- "To create a personal subpanel configuration file" in Chapter 17 explains how to create a personal **fp.terminal** file.
- "To use Create Action in HP VUE Lite" covers creating actions with the Create Action utility.

## To create a personal Terminals subpanel configuration file

- Copy **/usr/vue/config/panels/fp.terminal** to the /*HomeDirectory*/.vue directory. Give the new file write permission.

- Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox. This opens your personal Workspace Manager configuration file (/*HomeDirectory*/.vue/vuewmrc) for editing.

- Edit **vuewmrc** to include the new file by replacing the line that includes the system subpanel file with a line that includes the new file.

```
INCLUDE
{
    "HomeDirectory/.vue/fp.terminal"
}
```

## Example: Adding a terminal to the Terminals subpanel

These steps create a personal Terminals subpanel with an additional control that starts an **hpterm** window with scrollbars. Assume your home directory is **/users/tim**.

**Create a personal
'fp.terminal' file**

1. Copy **/usr/vue/config/panels/fp.terminal** to
   **/users/tim/.vue/fp.terminal**. Give the new file write
   permission.

2. If **/users/tim/.vue/vuewmrc** doesn't already exist, create it by
   copying **/usr/vue/config/sys.vuewmrc** and giving the new file write
   permission.

3. In **vuewmrc** edit the line that includes **fp.terminal**:

   ```
   INCLUDE
   {
       :
     "/usr/vue/config/panels/fp.terminal"
   }
   ```

   to include the personal file instead. Save the file.

**Add the control to
'fp.terminal'.**

4. Open **/users/tim/.vue/fp.terminal** and add the new control to the
   subpanel box:

   ```
   BOX TerminalSubpanel
      {
      TYPE      subpanel
       :
      CONTROL   HpTermScroll
   }
   ```

5. Add the control to the bottom of the file.

   ```
   CONTROL HpTermscroll
   {
       TYPE          button
       LABEL         "Scrollable hpterm"
       IMAGE         termina
       PUSH_ACTION   f.action HptermSrollBar
       HELP_STRING   "This control opens an hpterm window with scrollbars."
   }
   ```

6. Save **fp.terminal**.

**Create a new action**

7. Choose Create Action in the Tools subpanel to open the Create Action
   dialog box.

8. Provide values for these fields:

   ```
   Name: HptermScrollBar
   Command Line: hpterm -sb
   ```

                  `Window Type: X Windows`

9. Choose Apply, then Close.

**Restart the workspace manager**    10. Choose Restart Workspace Manager from the Workspace menu.

# Adding Applications and Utilities to HP VUE Lite

The Tools control in the HP VUE Lite Front Panel provides easy access to applications and utilities. You can:

- Configure the HP VUE Lite Tools control in the top row of the Front Panel to start an application.

- Add applications to the Tools subpanel.

- Create your own actions, which can then be added to the Front Panel.

## To assign an application to the HP VUE Lite Tools control
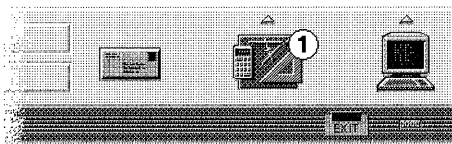


**You can configure the Tools control ① to start the application of your choice.**

- Choose Create Action in the Tools subpanel to display the Create Action dialog box.

- Fill in the required fields. Name the action `MainTool`

- If desired, fill in the optional fields.

- Choose Apply to save the action, then Close.

- Choose Restart Workspace Manager from the Workspace Menu.

The action name `MainTool` attaches the new action to the Tools control on the top row.

**See Also**

- "To use Create Action in HP VUE Lite" explains how to use Create Action.

## Example: Assigning an application to the HP VUE Lite Tools control

You frequently use the application IslandPaint$^{TM}$ and want to be able to start it by choosing the Tools control. The command to start the application is

> `IslandPaint` *filename*

**Define the action**

1. Choose Create Action in the Tools subpanel to open the Create Action dialog box.

2. Supply these values to the fields:

   ```
   Name:             MainTool
   Command Line:     IslandPaint
   Window Type:      X Windows
   Filename Prompt:  IslandPaint data file:
   ```

   If IslandPaint is not located along the PATH search path, you must use the absolute path to the executable file.

3. Choose Apply, then Close.

4. Choose Restart Workspace Manager from the Workspace Menu.

## To add an application to the Tools subpanel

1. If the change is personal rather than system wide, create a personal `fp.tool` file (see "To create a personal Tools subpanel file").

If the change is system-wide, use the system file
`/usr/vue/config/panels/fp.tool` (you must be superuser).

2. Open `fp.tool` and add a control to the subpanel box.

```
BOX ToolsSubpanel
{
    TYPE       subpanel
      ⋮
    CONTROL   new_control_name
}
```

3. Add the definition for the new control to the file.

4. Save the file.

5. Create the action for the application.

6. Choose Restart Workspace Manager from the Workspace menu.

**See Also**

- "Example: Adding an Action to the Tools subpanel" is a step-by-step example.
- "To create a personal subpanel configuration file" in Chapter 17 explains how to create a personal `fp.tool` file.
- "Creating a Simple Action" in Chapter 19 covers creating actions with the Create Action utility.

# To create a personal Tools subpanel file

- Copy the `/usr/vue/config/panels.fp.tool` to the
  /*HomeDirectory*/ `.vue` directory. Give the new file write permission.

- Edit the new file to add or remove controls.

- Double-click EditVuewmrc in the System_Admin subdirectory of the General Toolbox. This opens your personal Workspace Manager configuration file (/*HomeDirectory*/ `.vue/vuewmrc`) for editing.

- Edit `vuewmrc` to include the new file by replacing the line that includes the system subpanel file with a line that includes the new file.

```
INCLUDE
```

```
{
    /HomeDirectory/.vue/fp.tool
}
```

## Example: Adding an Action to the Tools subpanel

The action ChangePassword is built into HP VUE. You want to add it to the
Tools subpanel. Assume your home directory is **/users/tim**.

**Create a personal
'fp.tool' file**

1. Copy **/usr/vue/config/panels/fp.tool** to
   **/users/tim/.vue/fp.tool**. Give the new file write
   permission.

2. If **/users/tim/.vue/vuewmrc** doesn't already exist, create it by
   copying **/usr/vue/config/sys.vuewmrc** and giving the new file write
   permission.

3. In **vuewmrc** edit the line that includes **fp.tool**:

   ```
   INCLUDE
   {   .
       .
       .
       "/usr/vue/config/panels/fp.tool"
   ```

   to include the personal file instead. Save the file.

**Add the control to
'fp.tool'.**

4. Open **/users/tim/.vue/fp.tool** and add the new control to the
   subpanel box:

   ```
   BOX ToolsSubpanel
       {
       TYPE       subpanel
           .
           .
       CONTROL  Password
       }
   ```

5. Add the control to the bottom of the file.

```
CONTROL Password
{
    TYPE          button
```

```
LABEL        "Change Password"
PUSH_ACTION  f.action ChangePassword
HELP_STRING  "This control runs the utility for changing your password."
}
```

6. Save **fp.tool**.

**Restart the workspace manager**

7. Choose Restart Workspace Manager from the Workspace menu.

# To use Create Action in HP VUE Lite

**Start Create Action**

1. Choose Create Action in the Tools subpanel.

**Supply the required information**

2. In the Name field, type a unique name for the action.

3. In the Command Line field, type the command to start the application. Type the command exactly the way you would type it in a command line, except that, where you would type a file name, substitute $n$, where $n$ is an integer. Here are some example command lines:

```
emacs
bitmap $1
diff $1 $2
lp -oraw $1
```

4. Use the Window Type options menu to select how the action's output will be displayed:

| | |
|---|---|
| X Windows | The application creates it's own window. |
| No Output | The command has no display output. |
| Terminal | The application runs in a terminal emulator window. You want to keep the terminal emulator running until you explicitly close it. |
| Terminal (auto-close) | The application runs in a terminal emulator window. You want the terminal emulator window to close when you exit the application. |

**Supply a prompt for data**

5. If the command takes a data file (the Command Line contains $1), type the text of the prompt into the Filename Prompt field. Leave this field empty if the command does not take a data file.

**Supply optional information**

6. In the Description field, type the item help for the action icon. Press (Enter) to wrap text to the next line in the text box. The newline characters will be ignored.

7. If the action is to be used only in HP VUE Lite, leave the Large Icon and Small Icon fields empty.

**Store the action**

8. Choose Apply. A dialog box appears telling you the name of the database file created for the new action, and the system is busy momentarily as the database is reread.

9. If the action has been successfully created, choose Close.

**Make the new action available**

10. If the action is used in the Front Panel, choose Restart Workspace Manager from the Workspace menu.

A new action definition takes effect:

- When the application that uses it (for example, the Workspace Manager) is restarted.

- *Or*, when the user logs out and back in.

You can use Create Action to create actions that execute local applications. More complex actions are created manually in a database configuration file.

**See Also**

- "Creating Actions Manually" in Chapter 21 covers creating actions by editing a database file.

# To make a system-wide action in HP VUE Lite with Create Action

1. Create the action using the Create Action utility. When you apply the action, make note of the file to which the definition is saved.

2. Log in as root.

3. Copy the definition file to /usr/vue/config/types.

The action can now be used in a system-wide Front Panel control.

**See Also**

- "To use Create Action in HP VUE Lite"

# To run an action from the command line

- In a Terminal Emulator window, execute:

     vueaction *action_name*

The **vueaction** utility lets you run an action without adding it to the Tools subpanel.

**Example**

You can use the built-in EditResources action to change resources:

1. Execute:

     vueaction EditResources

   Your resources will be displayed using your chosen editor. (If your editor users its own window, two windows will be opened—one for your editor, and a Terminal Emulator window. You can ignore the terminal window.)

2. Edit your resources and save the file.

3. Close the window containing the editor. Your resources will automatically be written to your resource database.

# Customizing Workspace Manager

# Customizing Workspace Manager

Workspace Manager controls how items on the screen look and behave, and how they respond to input from the mouse or keyboard. You can extensively customize most features, including:

- The number and appearance of the workspaces.

- The appearance of windows and icons in the workspace.

- The appearance and contents of the Front Panel

- Mouse and keyboard actions.

- Workspace and menu button menus

Many of these can be changed with Style Manager. Style Manager is able to make often-used changes quickly, with little effort on your part. Other resources must be set manually.

The HP VUE Workspace Manager is **vuewm**. It is based on the OSF/Motif Window Manager.

**See Also**

- Refer to the **vuewm** man page for a complete list of resources.
- Chapter 9 explains using Style Manager
- Chapter 25 explains when and where to specify resources.

# Workspace Manager Configuration File

Workspace Manager gets information about the Front Panel, window menus, workspace menus, button bindings, and key bindings from a resource file.

sys.vuewmrc
> This file contains the system-wide default values for the window manager features. This file is shipped with HP VUE. Located in the /usr/vue/config/ directory.

vuewmrc
> A personal version of sys.vuewmrc. This file is used if you want to change anything in the sys.vuewmrc file. It is located in your *HomeDirectory*/.vue/ directory.

If your vuewmrc exists, HP VUE uses it, otherwise HP VUE uses sys.vuewmrc.

## To edit vuewmrc

1. Execute the EditVuewmrc action in the System_Admin subdirectory of the General Toolbox.

   (In HP VUE Lite, execute vueaction EditVuewmrc.)

   If you already have a personal vuewmrc file, it is loaded into the editor. If not, sys.vuewmrc is copied to vuewmrc, which is then loaded into the editor.

2. Edit the file.

3. Exit the editor. The file is saved as your personal vuewmrc, regardless of its original source.

# To include other files in vuewmrc

1. Use the EditVuewmrc action to open an editor containing your **vuewmrc** file.

2. Add `include` statements at the appropriate locations within the **vuewmrc** file.

3. Exit the editor.

4. Choose Restart Workspace Manager from the Workspace menu.

The syntax for `include` statements is:

```
include
{
file
  .
  .
  .
}
```

The included file must contain an entire panel, menu, box, or control. The `include` command cannot be imbedded in existing definitions.

Menus, button, boxes, or controls described in included files are used in the same manner as those items described within the **vuewmrc** file.

**Example**

A file named **mymenu** in the **users/ellen/** directory contains the following:

```
Menu MyLittleMenu
{
    "Little Menu"               f.title
    "Refresh Screen"            f.refresh
    "Restart Workspace Manager" f.restart
}
```

The **vuewmrc** file contains the following lines. They could be placed anywhere in the file, but locating them with the other menu descriptions is best.

```
  .
  .
include
{
/users/ellen/mymenu
}
  .
  .
```

The following button mapping, also in **vuewmrc**, displays **mymenu** when
button 3 is pressed and the default menu when button 1 is pressed.

```
<Btn1Down>     root    f.menu    VueRootMenu
<Btn3Down>     root    f.menu    MyLittleMenu
```
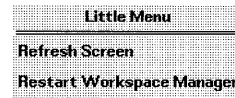


**This menu is displayed when button 3 is pressed.**

# Customizing Workspaces

The default HP VUE configuration provides six workspaces. The default HP VUE Lite configurations provides four workspaces. You can add additional or remove workspaces, or change the appearance of any workspace.

## To customize backdrops

1. Create a directory for your backdrop bitmaps.

2. Create a subdirectory named `Color` for backdrop pixmaps.

3. Create a symbolic link between your new bitmap directory and the HP VUE backdrops by executing:

       ln -s /usr/vue/icons/Vuebackdrops/* *yourDirectory*

4. Create a symbolic link between your new pixmap directory, `Color`, and the HP VUE backdrops by executing:

       ln -s /usr/vue/icons/Vuebackdrops/Color/* *yourDirectory*/Color

5. Use the EditResources action to change the `backdropDirectory` resource to use your directory.

       *backdropDirectory:     *yourDirectory*

6. Create your bitmap files in your directory.

7. Create your pixmap files in your new `Color` subdirectory.

8. Choose Restart Workspace Manager from the Workspace menu.

9. Restart Style Manager by clicking the Style Manager control.

The default HP VUE backdrop images are in the /usr/vue/icons/Vuebackdrops directory. Once you change the backdropDirectory resource, backdrops in the default directory will not be available unless you link them to your directory.
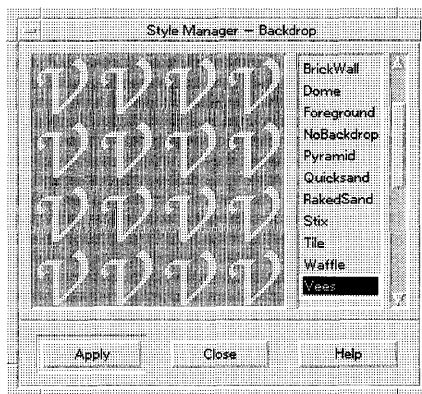
**The custom backdrop appears in the Backdrop Dialog.**

**See Also**

- "Where to put backdrop image files" in Chapter 16 explains Icon Editor and more about these directories.

---

## To make a "deep" backdrop

1. Issue the following command in the directory you use for custom backdrops:

   ln -s *backdrop* **deep***backdrop*

2. Restart Style Manager by clicking on the Style Manager control.

3. Choose Restart Workspace Manager from the Workspace menu.

The "deep" backdrop is a series of concentric borders that gives the impression of depth. You can combine this feaature with any other backdrop. The second file name in the link must begin with **deep**.

Both names appear in the list of backdrops. You can select the original version of the backdrop or the version with the deep border around it.

**Example**      The following line provides a border around a backdrop named "bigV":

```
ln -s bigV.bm deepbigV.bm
```
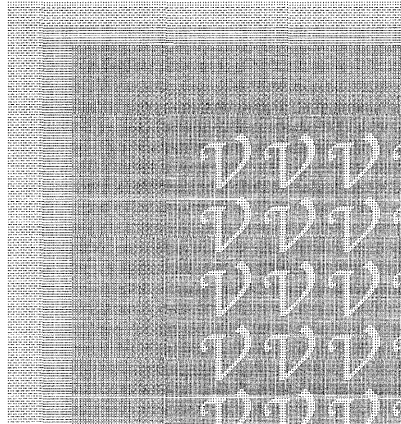


**The deep backdrop can be combined with other backdrops.**

## To display the root window in a workspace

- Select "NoBackdrop" in Style Manager's Backdrop dialog.

In HP VUE, the root window is the window behind the workspace backdrops. Ordinarily, the root window is completely obscured by the backdrops. "NoBackdrop" creates a transparent backdrop that allows you to see the root window.

**xsetroot**      The **xsetroot** client allows you to set colors and bitmaps for the root window, for instance:

```
xsetroot -bitmap /users/ellen/bitmaps/bigV
```

tiles the root window with the specified bitmap. This differs from setting a backdrop in that no resources are updated, nor does the root window color or bitmap show in Style Manager's Workspaces dialog.

**xsetrootgif**         If you have the **xsetrootgif** client, you can display `.gif` images in the root
                        window, effectively using the image as a backdrop:

```
xsetroot -gif myfamily.gif
```

**See Also**            • Refer to the **xsetroot** or **xsetrootgif** man pages for details about how
                        to use these clients.

# Customizing Window Components

**Frame Elements**

You can customize the appearance of the frame which surrounds your windows:

- Components included in the frame.
- Colors used in the frame.
- Placement of the window.

**Icon elements**

An icon is used to represent a window that has been minimized. You can customize icon:

- Appearance
- Location

---

**NOTE**

If you use resources to set frame or icon colors, those elements are not dynamic, that is they will not change when you make a new color selection with Style Manager.

---

**See Also**

- The **vuewm** man page has a complete list of the icon resources.
- Chapter 25 provides information about how and where to set resources.

# To specify window frame components

1. Use the EditResources action to set the value of the `clientDecorations` resource for the front panel to `all`:

       Vuewm*FrontPanel*clientDecorations:    all

2. Choose Restart Workspace Manager from the Workspace menu.

A window frame is composed of several elements.
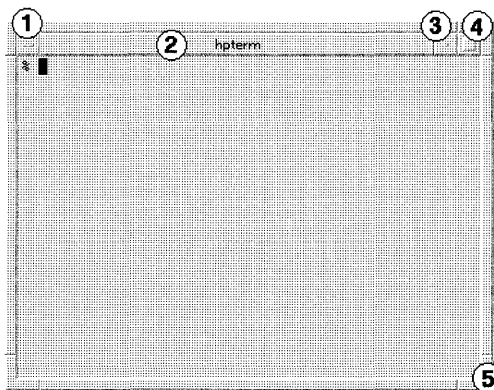


**There are several components in a window frame.**

where:

| | |
|---|---|
| ① | Window menu button. |
| ② | Title bar. |
| ③ | Minimize button. |
| ④ | Maximize button. |
| ⑤ | Resize handles. |

Two resources allow you to specify what components you want in your window frames:

| | |
|---|---|
| `clientDecoration` | Specify components for all windows. |
| `transientDecoration` | Specify components for short-lived windows, such as dialog boxes. |

You can specify the decoration for all clients, or specific ones. (Refer to "Reference: Workspace Manager" for information about the correct format for each style.)

These resources can take the following values. If the first item in the list starts with a plus (+), the workspace manager starts with no frame elements and adds the ones you specify. If the first item starts with a minus (-), the window manager starts with a complete frame, and removes the elements you specify.

| | |
|---|---|
| `all` | Include all frame elements (default value). |
| `none` | Include no window frame elements. |
| `±border` | Include or exclude border. |
| `±maximize` | Include or exclude the maximize button. The title bar is shown if this button is included. |
| `±minimize` | Include or exclude the minimize button. The title bar is shown if this button is included. |
| `±resizeh` | Include the resize border handles. The border is shown if this element is included. |
| `±menu` | Include the menu button. The title bar is shown if this button is included. |
| `±title` | Include the title bar. |

**Example**

To remove the maximize button and resize handles from all client windows:

```
Vuewm*clientDecoration:   -maximize-resize
```

To specify only a title bar and menu button for a client named `scribe`:

```
Vuewm*scribe*clientDecoration: +menu
```

**See Also**

- "Reference: Workspace Manager" has information the formats to use in specifying Workspace Manager resources.
- Chapter 25 explains where and how to specify resources.

# To set Front Panel decorations

1. Use the EditResources action to set the value of the `clientDecorations` resource for the front panel to `all`:

   ```
   Vuewm*FrontPanel*clientDecorations:    all
   ```

2. Choose Restart Workspace Manager from the Workspace menu.

By default, the decoration for the Front Panel is turned off. By setting the decoration to `all`, you allow all legal Front Panel components to be displayed. For instance, the Front Panel is not resizable, so no resize handles are shown.

**See Also**

- Chapter 25 explains where and how to specify resources.

# To set subpanel decorations

1. Use the EditResources action to set the value of the `subpanelDecorations` resource.

2. Choose Restart Workspace Manager from the Workspace menu.

The `subpanelDecorations` resource sets the default subpanel decoration. It can take the following values:

| | |
|---|---|
| `all` | Include all frame elements (default value). |
| `none` | Include no window frame elements. |
| `±border` | Include or exclude border. |
| `±maximize` | Include or exclude the maximize button. The title bar is shown if this button is included. |
| `±minimize` | Include or exclude the minimize button. The title bar is shown if this button is included. |
| `±resizeh` | Include the resize border handles. The border is shown if this element is included. |

±menu             Include the menu button. The title bar is shown if this
                  button is included.

±title            Include the title bar.

This resource sets the default subpanel decoration. You can specify
decorations for individual subpanels by allocating them directly:

        Vuewm*_subpanel_*clientDecoration:     _value_

**See Also**        • Chapter 25 explains where and how to specify resources.

---

# To manually set window frame color

Each palette shown in the Color Dialog box has up to eight color sets within
it. These resources specify which color set is to be used for active and
inactive windows. These resources can be set either manually or through the
Color Dialog Box.

activeColorSetId      Active window frames. The default color set is 1.

inactiveColorSetId    Inactive window frames, menus, dialog boxes, and
                      icons. The default color set is 2.

For example, the following lines set the active windows to the colors in color
set 3 and the inactive windows to the colors in color set 4.

        Vuewm*activeColorSetId:      3
        Vuewm*inactiveColorSetId:    4

**See Also**        • Chapter 25 for how and where to set resources.

# To change icon appearance

This section covers icon bitmaps and labels. For other icon customization, refer to the **vuewm** man page.
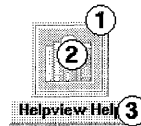


**An icon is composed of three parts.**

where:

①                Frame.

②                Image.

③                Label.

**Icon image**          To specify a bitmap to use within the icon, either:

1. Create the bitmap.

2. Use the EditResource action to set the `iconImage` resource.

3. Choose Restart Workspace Manager from the Workspace menu.

The `iconImage` bitmap is specified in relation to the `bitmapDirectory` resource. The syntax is:

    `Vuewm*`*client*`*iconImage:`     *path/bitmap*

where *client* is:

*client class*       The class of the client.

*client name*      The name of the client.

`default`          To be used when the client doesn't specify an icon.

The `bitmapDirectory` resource specifies the path to the directory
containing the bitmaps you want to use. The default directory is
`/usr/include/X11/bitmaps/`. If you use custom bitmaps, either

- Specify the complete path name of the bitmap in the `iconImage` resource.

  This method allows you to continue to use the default bitmaps.

- Change the `bitmapDirectory` resource to your bitmap directory.

  This method does not use the default bitmaps unless you copy them into
  your own directory.

**Example**

This example assignes a bitmap named "bigV" located in the
`/users/ellen/bitmaps` directory as the icon image to be used if the
application doesn't supply an image.

```
Vuewm*default*iconImage:      /users/ellen/bitmaps/bigV
```



**This custom icon replaces the default icon.**

**Icon label**

Use the `iconDecoration` resource to specify the part of an icon to be
displayed:

`label`                 Label only.

`image`                 Image only.

`label image`           Both label and image.

`label activelabel`     An untruncated active label.

For example, the following line specifies a default decoration of an image and
label, with an untruncated label when the icon is active.

```
Vuewm*icon*iconImage:    label activelabel image
```

**See Also**

- Chapter 25 explains how and where to specify resources.
- Chapter 16 explains how to change images within the icons.

# Workspace Manager Menus

Workspace Manager has two default menus. You can add others.

The workspace menu     Sometimes called the "root menu". The HP VUE
default button bindings display this menu when
mouse button 1 or 3 is pressed on the backdrop.
The menu is associated with those buttons through
button bindings.

The window menu     The menu displayed by pressing the menu button in
a window's frame, or mouse button 3 at any point
on the window frame. The menu is associated with
the button by the **windowMenu** resource.

**NOTE**

To keep window operation consistent between applications, modify only the workspace menu.

Workspace Manager menus are defined in **vuewmrc**.

# Workspace Manager menu syntax

**Syntax**
Workspace Manager menus have the syntax:

**Menu** *MenuName*
{
    *selection1 [mnemonic] [accelerator] function [argument]*
    *selection2 [mnemonic] [accelerator] function [argument]*
    · · ·
    *selectionn [mnemonic] [accelerator] function [argument]*
}

where:

| | |
|---|---|
| *selection* | The text or bitmap that appears on the menu. |
| *mnemonic* | A single character that acts as a keyboard shortcut to this selection when the menu is displayed. It is specified in the form: *_character*. |
| *accelerator* | A key or set of keys that can be pressed to make this selection even when the menu is not visible. |
| *function* | The function to be performed when this selection is made. Refer to the **vuewmrc** man page for a list of functions. |
| *arguments* | Function arguments if the function needs them. Refer to the **vuewmrc** man page for more details. |

**Selection**
The item which appears on the menu can be either text or a bitmap.

If text is used:

- Multiple word text must be enclosed in quotation marks if it contains spaces, for example "Raise window".

- Single word text does not have to be enclosed in quotation marks. Either "Raise" or Raise is valid.

- The underscore character can be used to replace spaces. Either "Raise_window" or Raise_window is valid.

If a bitmap is used:

- The path must be preceded by a "@" sign.

**Accelerator**

Accelerators are in the form:

*modifier*<**Key**>*Keyname*

where *modifier* is a **modifier key**:

Ctrl        The CTRL key.

Shift       The Caps key.

Meta        The Extend char key.

Lock        The Lock key.

For keys with letters or numbers, the key name is usually what is printed on the key. For instance the name of the "a" key is "a", and the "2" key is named "2". The "Tab" key is named "Tab". The "F3" key is named "F3".

For keys that do not have letters or numbers on them, the use of the key is spelled out, for instance, the "+" key is named "plus".

Special key combinations, keypad keys, and non-ASCII keys must be spelled out.

For a list of all possible key names, refer to the /usr/include/X11R5/X11/keysymdef.h file. Remove the "XK_" part to find the key name. For example, the ":" character shows in the file as "XK_color", but you would use it as "colon".

**Example**

The following menu item displays the word "Restore". When it is chosen, it normalizes the window. When the menu is displayed, typing "R" will also restore the window. Pressing (Extend char) (F5) will also restore the window.

```
Restore    _R   Meta<Key>F5   f.normalize
```

A bitmap can be used instead of words on a menu. The following menu item displays a terminal icon and starts an hpterm window when chosen.

```
@/users/ellen/bitmaps/termbit.bm  f.exec "hpterm &"
```

# To use a custom workspace menu

1. Use the EditVuewmrc action to to create a new menu with a unique name. Use **VueRootMenu** or **VueLiteRootMenu** as a guide.

2. Change the button bindings to call the new menu.

3. Choose Restart Workspace Manager from the Workspace menu.

The new menu is displayed whenever mouse button 1 or 3 is pressed when the pointer is in the workspace.

**Example**

This example defines a new workspace menu called **MyMenu** for a regular HP VUE session. These lines would be located in the **vuewmrc** file.

```
Menu MyMenu
{
    "My Menu"                        f.title
    "Restart Workspace Manager..." f.restart
    "Log out..."                     f.action EXIT_SESSION
    @/users/ellen/bitmaps/shuffle  f.circle_up
}
```

The following line in the **VueButtonBindings** or **VueLiteButtonBindings** section of **vuewmrc** call the usual root menu if button 1 is pressed, and the custom menu when button 3 is pressed:

```
<Btn1Down>      root      f.menu    VueRootMenu
<Btn3Down>      root      f.menu    MyRootMenu
```
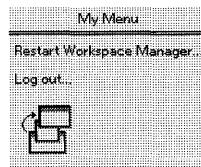


**This custom root menu is displayed when button 3 is pressed.**

> **NOTE**
>
> This menu replaces your existing workspace menu. If you want to keep certain menu items, copy them from **VueRootMenu** or **VueLiteRootMenu** into your new menu.

## To use a custom window menu

1. Use the EditVuewmrc action to create a new menu with a unique name. Use **SampleWindowMenu** as a guide.

2. Use the EditResources action to set the **windowMenu** resource to the menu name.

3. Choose Restart Workspace Manager from the Workspace menu.

**Example**

The following lines in **vuewmrc** ...

```
Menu MyWindowMenu
{
    "Restore"            _R        f.normalize
    "Minimize"           _n        f.minimize
    "Maximize"           _x        f.maximize
    "Close"              _C        f.kill
}
```

and the following line added to the resoures ...

```
Vuewm*xterm*windowMenu:    MyWindowMenu
```

cause the custom menu to appear when the window menu button is pressed.
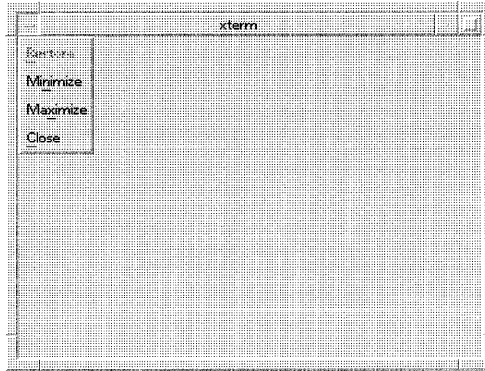
**This custom window menu is displayed when the window button is pressed.**

# Customizing Button Bindings

A **button binding** associates a mouse button operation and possible keyboard modifier key with a window manager function. Button bindings apply to all workspaces.

Button bindings contain a description of what keys are pressed, where the pointer is when the keys are pressed, and what happens when the keys are pressed.

**Default button bindings**  These default button bindings are shown in `vuewmrc` for reference only, and cannot be changed. These button bindings are described in the `DefaultButtonBindings` section of `vuewmrc`.

| Action | Result (location of pointer) |
|---|---|
| Press button 1 | Displays the window menu. (window menu button) |
| Double-click button 1 | Closes the window. (window menu button) |
| Click button 1 | Minimizes the window. (minimize button) |
| Click button 1 | Maximizes the window. (maximize button) |
| Dragging button 1 | Moves the window. (title bar) |
| Press button 1 | Gets keyboard focus. (window, icon) |
| Dragging button 1 | Resizes window. (resize border) |
| Click button 1 | Displays icon window menu. (icon) |
| Double-click button 1 | Normalizes the window. (icon) |
| Dragging button 1 | Moves the icon. (icon) |

These button bindings are in the `VueButtonBindings` section of `vuewmrc` and can be modified.

| Action | Result (location of pointer) |
|---|---|
| Pressing button 1 | Displays workspace menu. (root) |
| Pressing button 3 | Displays workspace menu. (root) |
| Pressing button 1 | Raises window to the top of the stack. (frame, icon) |
| Pressing button 3 | Displays window menu. (frame, icon) |

(Alt) (Extend char) and          Moves the window. (icon, window)
dragging button 1

(Atl) (Extend char) and          Minimizes the window. (window)
pressing button 3

# Button binding syntax

The syntax for button bindings is:

```
Buttons ButtonBindingSetName
{
       button context function [argument]
       button context function [argument]
       button context function [argument]
}
```

where:

*button*       A combination of a button name, possible modifier, and an action.

*context*      Indicates where the pointer must be for the binding to be effective. These can be concatenated together if the binding applies to more than one context. Multiple contexts are separated by the " | " character.

*function*     One of the window manager functions. Refer to the **vuewm** man page for a list of valid functions.

*argument*     Any window manager function argements that are required. Refer to the **vuewm** man page for details.

**Button names**   Btn1          The left mouse button.

Btn2          The middle mouse button on a 3-button mouse, or the right button on a 2-button mouse.

Btn3          The right button on a 3-button mouse, or both buttons on a 2-button mouse.

| | Btn4 | Buttons 1 and 2 together on a 3-button mouse. Not valid for a 2-button mouse. |
| --- | --- | --- |
| | Btn5 | Buttons 2 and 3 together on a 3-button mouse. Not valid for a 2-button mouse. |
| **Modifiers** | Ctrl | The CTRL key. |
| | Shift | The Caps key. |
| | Meta | The Extend char key. |
| | Lock | The Lock key. |
| **Operations:** | Down | Holding down a mouse button. |
| | Up | Releasing a mouse button. |
| | Click | Pressing and releasing a mouse button. |
| | Click2 | Pressing and releasing a mouse button twice in rapid succession (double-click). |
| | Drag | Pressing a mouse button and moving the mouse without releasing the mouse button. |
| **Contexts** | root | The workspace window. |
| | window | Client window or window frame. |
| | frame | Window frame, excluding the contents. |
| | icon | Icon. |
| | title | Title bar. |
| | app | Client window (excluding the frame). |

**Example**

The following line in the **vuewmrc** file causes the menu described in **VueRootMenu** to be displayed when mouse button 1 is pressed while the pointer is in the workspace window (but not within client windows.)

```
<Btn1Down>     root     f.menu     VueRootMenu
```

## To create a custom button binding set

1. Use the EditVuewmrc action to create a new button binding set with a unique name. Use the **VueButtonBindings** or **VueLiteButtonBindings** section as a guideline.

2. Use the EditResources action to set the **buttonBindings** resource to the new name.

3. Choose Restart Workspace Manager from the Workspace menu.

Do not bind the same button to different functions for the click and press operations; and do not bind one function to the same button and context.

---

**NOTE**

The new button bindings replace your existing button bindings. Copy any button bindings you want to keep from **VueButtonBindings** or **VueLiteButtonBindings** into your new set.

---

**Example**

This example adds a binding allowing a menu named "Graphics Project" to be displayed when mouse button 2 is pressed. The original button bindings are still in effect, and were copied directly from **VueButtonBindings**.

```
Buttons MyButtonBindings
{
    <Btn1Down>          root            f.menu      VueRootMenu
    <Btn3Down>          root            f.menu      VueRootMenu
    <Btn1Down>          frame|icon      f.raise
    <Btn3Down>          frame|icon      f.post_wmenu
    Meta<Btn1Down>      icon|window     f.move
    Meta<Btn3Down>      window          f.minimize
    <Btn2Down>          root            f.menu      "Graphics Project"
}
```

The following resource makes the new set active.

```
Vuewm*buttonBindings:  MyButtonBindings
```

**See Also**    • Chapter 25 has information about how and where to change resources.

# Customizing Key Bindings

A **keyboard binding**, also known as a key binding, associates combination of keys with workspace manager functions. Key bindings apply to all workspaces.

Key bindings describe the keys pressed, where the keyboard focus is when the keys are pressed, and what action takes place when the keys are pressed.

---

**NOTE**

Be careful about using an existing key combination as a keyboard binding. For example, (Shift) (A) normally puts the letter "A" into your current window. If you bound (Shift) (A) to a function, you would lose its normal usage and always perform that function when (Shift) (A) was entered.

---

**Default key bindings**  The HP VUE default key bindings are located in the **VueKeyBindings** section of the **vuewmrc** file.

| Keys | Action (keyboard focus) |
|---|---|
| (Alt) (Menu) | Toggles Workspace Manager between iconified and restored state. (icon, window, none) |
| (Shift) (Escape) | Displays window menu. (window, icon) |
| (Alt) (Space) | Displays window menu. (window, icon) |
| (Alt) (Tab) | Switches keyboard focus to the next window or icon. (window, icon, none) |
| (Alt) (Shift) (Tab) | Switches keyboard focus to the previous window or icon. (window, icon, none) |
| (Alt) (Escape) | Switches keyboard focus to the next window or icon. (window, icon, none) |
| (Alt) (Shift) (Escape) | Switches keyboard focus to the previous window or icon. (window, icon, none) |

| | |
|---|---|
| (Alt) (▼) | Puts the top window on the bottom of the stack. (icon, window, none) |
| (Alt) (▲) | Puts the bottom window on the top of the stack. (icon, window, none) |
| (Alt) (F6) | Switches keyboard focus to the next transient window in an application. (window) |
| (Alt) (Ctrl) (Shift) (!) | Restarts **vuewm** with the default behavior. (window, icon, none) |

# Keyboard binding syntax

The syntax for keyboard bindings is:

```
Keys KeyBindingSetName
{
        key  context  function  [argument]
        key  context  function  [argument]
        key  context  function  [argument]
}
```

where:

| | |
|---|---|
| *key* | The modifier and key to which the function is mapped, using the form "Modifier<Key>keyname". |
| *context* | The element which must have the keyboard focus for this action to be effective. These can be concatenated together if the binding applies to more than one context. Multiple contexts are separated by the " | " character. |
| *function* | A window manager function. Refer to the **vuewm** man page for a list of valid functions. |
| *argument* | Any window manager function arguments that are required. Refer to the **vuewm** man page for details. |

| **Modifiers** | `Ctrl` | The CTRL key. |
|---|---|---|
| | `Shift` | The Caps key. |
| | `Meta` | The Extend char key. |
| | `Lock` | The Lock key. |

**Key names**

For keys with letters or numbers, the key name is usually what is printed on the key. For instance the name of the "a" key is "a", and the "2" key is named "2". The "Tab" key is named "Tab". The "F3" key is named "F3".

For keys that do not have letters or numbers on them, the use of the key is spelled out, for instance, the "+" key is named "plus".

Special key combinations, keypad keys, and non-ASCII keys must be spelled out.

For a list of all possible key names, refer to the /usr/include/X11R5/X11/keysymdef.h file. Remove the "XK_" part to find the key name. For example, the ":" character shows in the file as "XK_color", but you would use it as "colon".

| **Contexts** | `root` | Workspace window. |
|---|---|---|
| | `window` | Client window. |
| | `icon` | Icon. |

**Example**

The following line in the **vuewmrc** file allows the user to switch the keyboard focus to the next transient window in an application by pressing (Alt) (F6).

```
Meta<Key>F6    window    f.next_key    transient
```

# To create a custom key binding set

1. Use the EditVuewmrc action to create a new key binding set with a unique *KeyBindingSetName*. Use the **VueKeyBindings** as a guide.

2. Use the EditResources action to set the **keyBindings** resource to the new set name.

3. Choose Restart Workspace Manager from the Workspace menu.

---

**NOTE**

The new key bindings replace your existing key bindings. Copy any key bindings you want to keep from **VueKeyBindings** into your new set.

---

**Example**   The following key binding set in **vuewmrc** changes the result of pressing
(Extend char) (Shift) (Tab) and (Extend char) (Tab), and removes the binding for
(Extend char) (space).

When (Extend char) (Shift) (Tab) is pressed, the active window is iconified. When
(Extend char) (Tab) is pressed, the active icon is normalized.

```
Keys MyKeyBindings
{
    Shift<Key>Escape           icon|window          f.post_wmenu
    Meta<Key>Escape            root|icon|window     f.next_key
    Meta Shift<Key>Escape      root|icon|window     f.prev_key
    Meta<Key>Down              root|icon|window     f.circle_down
    Meta<Key>Up                root|icon|window     f.circle_up
    Meta Ctrl Shift<Key>exclam root|icon|window     f.set_behavior
    Meta<Key>F6                window               f.next_key transient
    Meta Shift<Key>Tab         window               f.minimize
    Meta <Key>Tab              icon                 f.normalize
}
```

The following resource directs HP VUE to use the new key bindings instead of
the original ones.

```
    Vuewm*keyBindings:        MyKeyBindings
```

# Switching between default and custom behavior

1. Press (Alt) (Shift) (Ctrl) (!) to toggle between OSF/Motif 1.1 default and HP VUE window behavior.

2. Click on OK in the dialog.

Switching to default behavior removes the Front Panel and any custom key and button bindings.

**NOTE**

If yo log out and save the session, you must toggle back to HP VUE behavior next time you log in to get your Front Panel back.

# Reference: Workspace Manager

**Workspace Manager resources**

Vuewm*_resource_:      _value_

For example, the following line sets the active color set ID to 2, which means that all active windows use the colors in color set 2.

Vuewm*activeColorSetId:     2

**Workspace component**

Vuewm*_component[*subcomponent]*resource_:      _value_

where _component_ is:

| | |
|---|---|
| client | Client window frames. |
| icon | Windows that have been minimized. |
| feedback | Quit or Restart dialogs. |
| menu | Menus produced by Vuewm. |
| frontPanel | The Workspace Manager |
| workspaceController | The Rename Workspace dialog. |
| workspacePresence | The Occupy Workspace dialog. |

For example, the following line provides a 7-pixel border around the Front Panel, makes the Workspace Menu background skyblue, and colors the frames around clients maroon.

```
Vuewm*frontPanel*borderWidth:     7
Vuewm*menu*background:            skyblue
Vuewm*clients*background:         maroon
```

The components can have subcomponents within them.

For example, the following sets the background for the title bar in all the client windows to red.

Vuewm*client*title*background:     red

**Client**

_client[*subcomponent]*resource_:      _value_

where client can be:

|  |  |
|---|---|
| *clientname* | The name of the client. |
| *clientclass* | The class to which the client belongs. |
| `default` | Used for clients of unknown class. |

For example, the following line specifies that an `hpterm` window named `pronto` will have only a title bar and menu button in its border.

```
Vuewm*pronto*clientDecoration:     +menu
```

**Screen or workspace**      `Vuewm[`*`scr][`*`wksp][`*`component][`*`subcomponent][`*`resource`:      *value*

where:

|  |  |
|---|---|
| *scr* | Name of the screen for which the component is applied. |
| *wksp* | Name of the workspace for which the component is applied. |

For example, if you use Style Manager to change the backdrop in workspace Two, then use the EditResources action to look at your current resources, you will see a line like:

```
Vuewm*0*Two*backdropImage:      BrickWall
```

"0" is the screen number as shown in the DISPLAY environment variable. "Two" is the name of the workspace.

**Scope**      The most specifically defined resource overrides a generally defined resource. In this example, only the background for the "clock" client is set to yellow. Backgrounds for other clients are set to blue. All other backgrounds are set to red.

```
Vuewm*background:      red
Vuewm*client*background: blue
Vuewm*xclock*background: yellow
Vuewm*Date*format:               %b %.1d%n%a
```

**See Also**      • `vuewm` man page contains a complete list of Workspace Manager resources.
• Chapter 22 to run vuewm using multiple displays.

# 25

## Using Resources

# Using Resources

A **resource** is used to describe certain features of HP VUE. For example, the following resource setting makes all your **hpterm** windows have scrollbars.

    *hpterm*scrollBar:    True

Each resource has a default value, but for most, you can change that value. You need to specify resource values only if they differ from the default value.

HP VUE uses the RESOURCE_MANAGER property of the root window to describe all resources that are not set to their default values.

At the beginning of an HP VUE session, Session Manager loads resources from one of its files into the RESOURCE_MANAGER property. During the session, you can modify the RESOURCE_MANAGER property as described below. At the end of a session, Session Manager stores the current resources into one of its files, ready to be used for the next session. (Refer to "System resource files" for more details.)

Resources in the RESOURCE_MANAGER property are added or changed in the following ways:

- Using Style Manager.

  Style Manager provides a graphical way to update resources. It requires no special knowledge of resource syntax.

- Editing the resources manually.

  Resources can be edited directly, either with the EditResources action or the **xrdb** client. You can use either method to include a personal resource file.

---

**NOTE**

It's a good idea to make a backup file of your current resources in the RESOURCE_MANAGER property if you plan to make many changes to them.

---

- "System resource files" lists the different resource files and when they take effect.
- Client man pages list the resources available for that client.
- Chapter 9 gives information about how to use Style Manager to customize HP VUE.

# Setting Personal Resources for an Application

Resources can be changed or specified in several ways (listed in order of preference):

**Style Manager**

You can use Syle Manager to change resources interactively.

This is the easiest method provided the resource is covered by Style Manager.

**EditResources action**

The EditResources action lets you add, change, or delete current resources using a text editor, or make a backup file of the current resources.

**xrdb client**

You can use the **xrdb** client to load or merge new resources into the current resources.

You can add resources individually, or from a resource file.

This method allows you to easily add resources, but you must create an intermediate resource file if you want to change or delete a resource.

> **NOTE**
>
> Unless you are familiar with **xrdb** or need some special **xrdb** option, use the EditResources action instead.

**Command-line options**

This method is the most specific, and allows you to start an application with exactly the resources you want, without affecting other versions of that application you might be running. (Refer to "Specifying resources in a command line" for more information.)

# To change resources with Style Manager

1. Click on the Style Manager control in the front panel.

2. Click on the Style Manager component you want to change.

3. Make and save the changes as directed in the component's dialog box.

4. Close the Style Manager component window.

5. Exit Style Manager.

The Style Manager can be used to change resources if the resource in question is one that Style Manager handles. Not all features handled by Style Manager involve resources. Additionally, there is usually a larger range of resources for a particular feature than are managed with Style Manager (for instance, Fonts). However, Style Manager does handle the most popular customization features.

The `writeXrdbImmediate` resource determines when new values for color, fonts, and mouse double-click speed are available. The default value is True.

| Value | Description |
|-------|-------------|
| `True` | Changes you make using Style Manager take effect in HP VUE windows and cooperating clients immediately if possible, or the next time the client is started. This is the default value. |
| `False` | The changes take effect at the beginning of the next session. |

If the value is `False`, you do not see the font or click-speed changes you make during your current session, even if you start new applications during the session.

# To set resources using the EditResources action

1. Double-click EditResources in the System_Admin subdirectory of the General Toolbox.

(In HP VUE Lite, execute **vueaction EditResources**.)

A list of your resources is loaded into a text file.

The editor used is the one specified in your EDITOR enviroment variable. If you use an editor that opens its own window, you will see an extra terminal window. Ignore it or close it. It will disappear when you close the editor window.

2. Edit the file. You can add, edit, or remove resources from the file.

3. Store the edited file.

4. Close the editor window.

5. Restart Workspace Manager.

The EditResources action uses the **xrdb** client to store the current resources in a temporary file for editing, then to load that file back into the RESOURCE_MANAGER property.

If you want to change the editor used to edit the temporary file, either:

• Change the EDITOR environment variable. This is the preferred method.

• Change the EditResources action.

**See Also**

• Chapter 14 describes environment variables.
• Chapter 19 describes actions.

# To create a backup resource file

**EditResoureces action**

1. Use the EditResources action to display your current resources in an editor window.

2. Use the feature of your editor which allows you to save the contents under a different name.

**xrdb**

- Issue the following command to create a file containing the current resources:

  xrdb -edit *filename*

## To add resources using xrdb

**Interactively**

1. Execute

   xrdb -merge -nocpp

2. Type in resource names and values. Each resource must be on a separate line.

3. When you've entered all the resources, press (CTRL) (d) twice.

4. Choose Restart Workspace Manager from the Workspace menu.

**Resource file**

1. Create a temporary file containing your new resources.

2. Execute

   xrdb -merge -nocpp *filename*

3. Choose Restart Workspace Manager from the Workspace menu.

The **-merge** option indicates that the resources in the file will be added to the current resources.

**Example**

This example shows three resources commonly set by users. The first line indicates that scrollbars should appear on all **hpterm** windows. The second line sets "tty modes" to override the default terminal settings. The third line establishes four workspaces, named Home, Mail, Sheets, and Edit.

```
hpterm*scrollBar:     True
*ttyModes:            erase ^H intr ^C kill ^U start ^Q stop ^S swtch ^@
Vuewm*workspaceList:  Home Mail Sheets Edit
```

**See Also**

- "Syntax of resource specifications" has information about how to specify resources.
- *Using the X Window System* has information about the **xrdb** client.
- The **xrdb** man page explains all the options for **xrdb**.

# To replace current resources using xrdb

1. Create a temporary file containing the current resources:

    **xrdb -edit** *filename*

2. Edit *filename*.

3. Execute

    **xrdb -nocpp** *filename*

4. Choose Restart Workspace Manager from the Workspace menu.

The resource values in the file replace the existing current resources.

**xrdb -nocpp** *filename* is the same as **xrdb -load -nocpp** *filename*. **-load** is the default option, and directs **xrdb** to replace the current resources (**-merge** directs **xrdb** to add the new resources to the current ones).

**Example**

This example loads a backup file named **myresources**, overwriting the current resources.

    **xrdb -nocpp myresources**

**See Also**

- *Using the X Window System* has information about the **xrdb** client.
- The **xrdb** man page explains all the options.

# To delete resources

1. Use the EditResources action to remove the resource from the list.

2. Choose Restart Workspace Manager from the Workspace menu.

When a resource is "deleted", it is removed from the Resource_Manager property.

# Setting System-wide Resources

If you are administering a system, you can establish resources that will be in effect when a new user starts HP VUE for the first time. After the first time, the system-wide resources files are ignored in favor of session-specific files.

The file used depends on whether the session is HP VUE or HP VUE Lite.

## To set sys.resources for first session

1. Logon to the system as superuser.

2. Edit **/usr/vue/config/sys.resources**.

When a new HP VUE user starts the first session, resources are copied from the **sys.resources** file into the users's RESOURCE_MANAGER property. When the session is ended, the RESOURCE_MANAGER property is copied to *HomeDirectory*/**.vue/sessions/current/vue.resources**. Subsequent sessions use this **vue.resources** file.

> **NOTE**
>
> Changes to the **sys.resources** file may be overwritten by future HP VUE releases.

# To set sys.res.lite for VUE Lite

1. Logon to the system as superuser.
2. Edit **/usr/vue/config/sys.res.lite**.

When a user starts the first HP VUE Lite session, resources are copied from the **sys.res.lite** file into the user's RESOURCE_MANAGER property. When the session is ended, the RESOURCE_MANAGER property is copied to *HomeDirectory*/ **.vue/sessions/lite/vue.resources**. Subsequent HP VUE Lite sessions use this resource file.

> **NOTE**
>
> Changes to the **sys.res.lite** file may be overwritten by future HP VUE releases.

# Reference: Resources

## Syntax of resource specifications

The general syntax for defining resources for a client is:

> *client\*[client components\*]resource*:      *value*

where:

*client*                        can be either the actual client name or the class to which the client belongs.

*client components*      allow you to define the resource to specific parts of the client.

*resource*                  can be either the resource name or class.

Resource files, including the RESOURCE_MANAGER property are text files and must obey the following rules:

- Each resource specification must be on a separate line. Use the backslash character (\) to continue a resource onto the next line if necessary.

- Any text to the right of an exclamation mark (!) is treated as a comment. This is a way to comment partial lines.

- A pound sign (#) in the first column makes the entire line a comment. (Use the **xrdb** option **-nocpp** to avoid it being interpreted as a preprocessor directive.)

- The resource name is separated from the value by a colon (:) and optional spaces or tabs.

- Do not have extra spaces or tabs after the value.

**Example**

The following lines set the background for all **hpterm** windows to blue, but the background for the scrollbar part of the window to yellow.

```
hpterm*background:        blue
hpterm*scrollBar:         True
hpterm*scrollBar*background:  yellow
```

**See Also**

- Client man pages for specific client components.

# How clients get resources

The X11 Resource Manager is built into HP VUE clients. It automatically merges together resource specifications from a variety of sources. The following list shows possible sources of resources in order of precedence; that is, categories at the top take precedence over those lower down. For instance, if a resource is specified on both the command line and in an **app-defaults** file, the value from the command line is used.

command-line options     Many resources can be specified in the command used to start an application. A command-line option is equivalent to a *client.resource* line in a resource file.

The options apply to only that single instance of the application.

**Xdefaults**     This file in your home directory is not read with the default HP VUE configuration. It was used in earlier HP VUE releases and in non-HP VUE systems. Refer to "To integrate X11 resource files into HP VUE" in Chapter 28 for information on how to include **Xdefault** file resources into HP VUE.

This file applies to individual users.

RESOURCE_MANAGER property     The general client resources are managed by this root window property on the server.

This property applies to individual users.

user-specific files | Files on the local host describing resources for a specific client class. These can be personal versions of system-wide **app-defaults** files or other personal files. These files can be changed by the user.

These files apply to individual users.

**app-defaults** files | Files on the local host describing resources for a specific client class, usually by application. These files should not be changed by the user.

These files apply to all users on the system.

built-in defaults | Resource default values can be defined in the client code.

These values apply to all users of the application.

**NOTE**

Using normal HP VUE processes to change the current resources in the RESOURCE_MANAGER property is the preferred method for users to specify personal resources. Using an **Xdefaults** file or user-specific files directly requires setting environment variables.

**See Also**

- man pages for specific clients list the command line options and resources for that client.
- X Toolkit Intrinsics documentation contains detailed information about how these different sources for resources are used. Refer to the **XtDisplayInitialize** function.

# Scope of resources

You can specify the scope of a resource value, that is, how generally or specifically the resource is applied. Scope of the customization is determined by:

- Using names or classes of clients.

- Using names or classes of resources.

- Specifying particular areas of a client.

- Using wildcards in the resource string.

The more specific resources take precedence over the more general resources.

**Client names and classes**

Every client has a name and a class. The name defines the client, while the class categorizes it. The class is more general than the name.

Often, the class and the name are similar, differing only in capitalization. For example, the client named **vuestyle** belongs to class **Vuestyle**.

Resources specified by client name take precedence over those specified by client class.

You can assign a name to a particular instance of a client, allowing you to refer to the client by name, client name, or class.

For example, if you have the following as a current resource . . .

```
localTerminal*scrollBar:     True
```

and issued the following command to start an **hpterm** window . . .

```
hpterm -name localTerminal
```

then that **hpterm** window will have a scroll bar in it.

**Resource names and classes**

Resources have both a name and a class. An individual resource name begins with a lower-case letter, such as **foreground**. A resource class begins with an upper-case letter, such as **Foreground**. The man page for each client lists the individual resources for that client and the resource class to which each belongs.

For instance,

```
*foreground:     blue
```

sets all foregrounds to blue. However,

```
*Foreground:     blue
```

sets all foregrounds to blue *and* any other resources that have a class of foreground (such as `cursorColor`) are also set to blue.

**Using wildcards**
The asterisk (*) character is a wild card providing resource generality.

**Examples**
The first line is the most general, and directs that all `foreground` resources be white. The second line indicates that the foreground resource for `hpterm` windows only be set to yellow, with all the rest left white. The third line is the most specific and indicates that the foreground for softkeys within `hpterm` windows be set to red.

```
*foreground:              white
*hpterm*foreground:       yellow
*hpterm*softkey*foreground:  red
```

# System resource files

**NOTE**

These files should not be changed by the user.

**HP VUE Files**

Session Manager maintains a number of resource files.

- `/usr/vue/config/sys.resources`
- *HomeDirectory/*`.vue/sessions/current/vue.resources`
- *HomeDirectory/*`.vue/sessions/home/vue.resources`
- *HomeDirectory/*`.vue/`*host:display/*`current/vue.resources`
- *HomeDirectory/*`.vue/`*host:display/*`home/vue.resources`

The `sys.resources` file contains system-wide resources. The first time you login to an HP VUE session, this file is read into the RESOURCE_MANAGER property. When the session ends, the RESOURCE_MANAGER property is stored in `vue.resources` in the appropriate directory, including any resource changes you made during the session. When the next session is begun, Session Manager copies the appropriate `vue.resources` file to the RESOURCE_MANAGER property.

**HP VUE Lite Files**

The Lite Session Manager maintains the following HP VUE Lite resource files.

- `/usr/vue/config/sys.res.lite`
- *HomeDirectory/*`.vue/sessions/lite/vue.resources`

The `sys.res.lite` file contains system-wide resources. The first time you login to an HP VUE Lite session, this file is read into the RESOURCE_MANAGER property. When the session ends, the RESOURCE_MANAGER property is stored in the `vue.resources` file, including any resource changes you made during the session. When the next HP VUE Lite session is started, Session Manager copies the `vue.resources` file to the RESOURCE_MANAGER property.

If you want to change these HP VUE resource files:

- Refer to "Setting System-wide Resources" for information about setting `sys.resources`.
- Refer to "Setting Personal Resources for an Application" for information about changing resources during a session.

**app-default files**

Many clients (including HP VUE) provide files with resources to be used by just that client. These files should not be changed by the user.

- HP VUE default files are located in the **/usr/vue/app-defaults** directory.

- Non-HP VUE application default files are in the **/usr/lib/X11/app-defaults** directory.

The entries in an **app-defaults** file are equivalent to a *Class*\**resource* entry in the RESOURCE_MANAGER property.

# Personal resource files

You can have personal resource files, which apply only to your sessions, and which you can change.

> **NOTE**
>
> Generally, it is better to load these files into the RESOURCE_MANAGER property, and manage the resources through normal HP VUE processes than to force HP VUE to use them as separate files.

**Personal app-defaults files**

If you want to override values in the system **app-defaults** files, do one of the following:

1. Use the EditResources action to update the current resources with the resource to be overridden.

2. Create your personal **app-defaults** file:

   a. Copy the **app-defaults** file to your home directory, using the same name. For example, **/users/ellen/XClock**.

   b. Edit the file.

3. Choose Restart Workspace Manager from the Workspace menu.

The resource is now specified in two places: the the RESOURCE_MANAGER property and the **app-defaults** file. Current resources in the RESOURCE_MANAGER property take precedence, so it is the value used.

**Temporary or backup files**

When you are making many changes to your current resources, it is a good idea to save the original set of resources in a backup file in case the changes don't work the way you planned:

To store your current resources, execute:

    xrdb -edit *filename*

**Xdefaults**

**Xdefaults** is a resource file that is used extensively in the X Window System, but not by HP VUE. In the default HP VUE configuration, it is ignored.

If you have an **Xdefaults** file from a previous HP VUE installation or X Windows environment, and want to use those resource values in your current environment, load the contents of **Xdefaults** into the current resources. (Refer to "To integrate X11 resource files into HP VUE" in Chapter 28.)

The XENVIRONMENT environment variable determines if an **Xdefaults** file will be read. If you *must* use the **Xdefaults** file, refer to the *Using the X Window System* manual for more information about setting the XENVIRONMENT environment variable.

**Loading personal files automatically**

If you have personal resource files other than personal **app-defaults** files, you should use the normal HP VUE processes to load the resource values into the current resources.

HP VUE loads personal resources files at the start of each session, depending on the values of the XUSERFILESEARCHPATH environment variable and the XAPPLRESDIR environment variable. Refer to the *Using the X Window System* manual or an Xt Intrinsics programming manual for information about how to set and use these environment variables.

# The RESOURCE_MANAGER property

The RESOURCE_MANAGER is a property of the root window. HP VUE uses the RESOURCE_MANAGER property to describe all resources that are not set to their default values.

There are several ways that resources are loaded into the RESOURCE_MANAGER property:

- Session Manager automatically loads resources from its saved session files at the start of a new session, and stores resources at the end of a session.

- You may add to or modify the RESOURCE_MANAGER property using Style Manager during a session.

- You may add to or modify the RESOURCE_MANAGER property using the EditResources action during a session.

- You may add to or modify the RESOURCE_MANAGER property using the **xrdb** client during a session.

Changes made during a session using **xrdb** are not retroactively effective, but apply to any new clients or processes started after the change.

Changes made with Style Manager may be written to the RESOURCE_MANAGER property immediately or at logout depending on the value of the **Vuestyle*writeXrdbImmediate** resource.

# Specifying resources in a command line

Most clients allow you to specify resource values on the command line used to invoke the client. The value is good only for that one instance of the client. If you use the client often and you want the resource used each time, add it to your current resources.

A command line option is equivalent to a *client.resource* line in a resource file.

Not every command-line option has an associated resource. However, by using the **-xrm** option, any resource can be specified on the command line.

If you save a session while the client is still running, HP VUE saves the command line so that the client will appear in the next session with the correct options.

**Example**     The following line starts a single **hpterm** window with the foreground and background colors reversed:

    hpterm -reverse

The following line also starts a single **hpterm** window with colors reversed, but uses the **-xrm** option to set the resource directly:

    hpterm -xrm "*reverseVideo:    True"

The following line in a resource file means that every **hpterm** window will use reverse colors:

    hpterm*reverseVideo:     True

## Resources and cached clients

Resource changes made to applications do not take effect until the client is restarted.

However, certain HP VUE applications started from Workspace Manager are "cached", that is, they may be running but not mapped to the display. Closing these applications simply unmaps them from the display, but does not end the application. They are not ended until the user logs out or ends them with a **kill** command.

Examples of cached applications are Help Manager and File Manager. Terminal emulator windows started from the front panel or as part of the session start-up procedure are also cached.

Resource changes for cached applications do not take effect until the application is killed and restarted or the current session is ended and a new one begun.

**Reference: Resources**

# Font, Color, and
# Application Resources

# Font, Color, and Application Resources

HP VUE allows you a wide range of colors and fonts to use in your display. While the simplest method of changing these resources is to use Style Manager, there are additional font and color resources you can customize.

- Chapter 9 for information about how to use Style Manager.
- Chapter 25 for details about how and where to specify the resources mentioned here.
- *Using the X Window System* explains fonts and colors in more detail.

# Managing Fonts

Using Style Manager's Font dialog box, you can select the size font you want for all your applications.

You can specify fonts on the command line or using resources to:

- Set font resources for individual applications.

- Assign different fonts to be used by the Fonts dialog box.

A font is a type style in which text characters are printed. HP VUE includes a variety of fonts in different styles and sizes.

A bitmapped font is made from a matrix of dots. The font is completely contained in one file. Many files are needed to have a complete range of sizes, slants, and weights.

The scalable typefaces are defined by a file containing a mathematical outline used by the system to create a bitmapped font of a particular size, weight, or slant. Only one "master" file is needed, but it can provide fonts customized as much as you need.

Fonts are specified as values of resources and as parameters to commands. The X Logical Font Description (xlfd) name is the method by which a desired font is requested. The system finds the font that best matches the description it was given.

**See Also**
- "To select a font size" in Chapter 9 explains how to use the Fonts dialog box in Style Manager.
- Chapter 25 explains how and when to set resources.

# To set HP VUE font resources

The Style Manger Font Dialog lets you set fonts for text entry, labels, and so on. A selection of up to seven sizes is presented.

**Set by Font Dialog**    A selection in the Font Dialog chooses two fonts. Samples of each are shown in the Font Dialog.
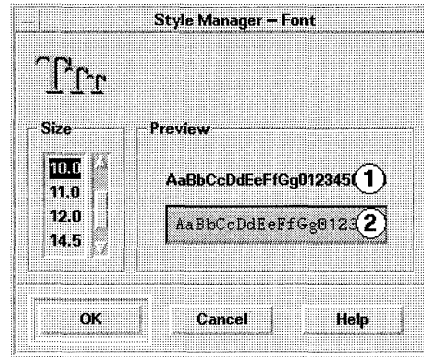


**The Font Dialog shows two fonts.**

① `SystemFont`    This font is used for system areas, such as menu bars, menu panes, pushbuttons, toggle buttons, and labels.

② `UserFont`    This font is used for text entered into windows.

When a font is selected, the following resources are written to the RESOURCE_MANAGER property:

- The following resources are set to `SystemFont`:

  `FontList`    Displayed in system areas of HP VUE clients and other clients created using the OSF/Motif toolkit.

- The following resources are set to `UserFont`:

  `Font`    General user font for clients.

  `XmText*FontList`    Displayed in text entry boxes.

  `XmTextField*FontList`    Displayed in text entry boxes.

**Used by Font dialog**    The fonts used for each selection in the Font Dialog Box are specified in the `/usr/vue/app-defaults/Vuestyle` resource file. Up to seven sizes can be specified.

`NumFonts`    The number of font sizes presented in the Font Dialog Box.

`SystemFont[1-7]`     Up to seven resources assigning a specific font to a
Font Dialog box selection for `SystemFont`.

`UserFont[1-7]`      Up to seven resources assigning a specific font to a
Font Dialog Box selection for `UserFont`.

---

**NOTE**

The default fonts for these resources have been carefully chosen to be easy to read on each type of display. Generally, if you want a specific font for an application, you should set it with an application font resource rather than changing these HP VUE fonts.

---

If you change one of these fonts, use all 14 field separators (-). You can use wild cards for the field itself, but do not combine fields with wild cards. For bitmap fonts, you need to provide a value for the pixel size field. For scalable typefaces, you need to provide a value for the point size field.

**See Also**         • Chapter 25 for information about how and where to specify resources.

---

## To list available fonts

• Execute

    xlsfonts [*-options*] [**-fn** *pattern*]

A list of xlfd names and alias names available on your system is produced. Bitmap fonts show values in all 14 xlfd fields. Scalable typefaces show zeros in the PizelSize, PointSize, ResolutionX, and ResolutionY fields.

If you have many fonts on your system, the list could be long. You can check for specific fonts by using the pattern-matching capability of **xlsfonts**. Use wild cards to replace the part of the pattern you are not trying to match.

If **xlsfonts** lists only three fonts, your *font path* is not including the HP VUE fonts. Type the following command to include the HP VUE fonts into your available fonts:

```
xset fp default
```

**Example**

This command . . .

```
xlsfonts -fn "*bitstream*"
```

. . . displayed this list of all fonts on the system with "bitstream" somewhere in their xlfd names.



**'xlsfonts' displays all fonts available to you.**

**See Also**

- **xlsfonts** man page for a list of the options available.
- *Using the X Window System* explains font alias names and the **xset** client.

# To specify fonts on the command line

- The **-fn** command line option allows you to specify a font for a specific client.

**Example**

```
xfd -fn "*bitstream-charter-medium-r-normal-8-88-75-75-p-45-iso8859-1"
```

# The X Logical Font Description (xlfd)

A font is specified by listing 14 different characteristics, separated by dashes
(-). This is called the X Logical Font Description (xlfd). In some cases, a
property in the list can be replaced by a "*" wild card, and a character
within a property can be replaced by a "?" wild card.

The form of the property string specification is:

*"FontNameRegistry-Foundry-FamilyName-WeightName-*
*Slant-SetwidwthName-AddStyleName-PixelSize-*
*PointSize-ResolutionX-ResolutionY-Spacing-*
*AverageWidth-CharSetRegistry-CharSetCoding"*

where:

*FontNameRegistry*    A string identifying the authority that registered the
                      font.

*Foundry*             A string identifying the font designer.

*FamilyName*          A string identifying the trademarked commercial
                      name of the font.

*WeightName*          A string giving the relative weight of the font, such
                      as bold.

*Slant*               A code describing the direction of slant.

                      R       Roman (no slant)

                      I       Italic (slant right)

                      O       Oblique (slant left)

                      RI      Reverse Italic (slant left)

                      RO      Reverse Oblique (slant right)

| | |
|---|---|
| *SetwidthName* | A string describing the width, such as compressed or expanded. |
| *AddStyleName* | A string providing any additional information needed to uniquely identify the font. |
| *PixelSize* | An integer giving the size of an em-square in pixels. |
| *PointSize* | An integer giving the size of an em-square in decipoints. |
| *ResolutionX* | An integer giving the horizontal resolution in pixels. |
| *ResolutionY* | An integer giving the vertical resolution in pixels. |
| *Spacing* | A code specifying the spacing between units. |

| | |
|---|---|
| M | Monospace (fixed pitch). |
| P | Proportional space (variable pitch). |
| C | Character cell. |

| | |
|---|---|
| *AverageWidth* | An integer giving the average width in 1/10th pixels. |
| *CharSetRegistry* | A string identifying the registration authority that has registered the font encoding. |
| *CharSetEncoding* | A string identifying the character set within the specified registry. |

**Example**

The following xlfd name:

```
*-bitstream-charter-medium-r-normal--8-80-75-75-p-45-iso8859-1
```

Describes a font named "charter" made by Bitstream and registered with ISO8859-1 standard. It is medium weight, no special slanting, and normal width. The font is proportional, with an em-square box of 8 pixels (or 8.8 points). The horizontal and vertical resolution are both 75 pixels. The average width of a character is 45 1/10ths pixels.

Parts of this string can be replaced by wild cards. The system uses the first font it finds that matches the parts you have specified.

If all you want is an 8-pixel "charter" font, you could use:

```
*-charter-*-*-*-*-8-*
```

# Managing Colors

You can color your display through two mechanisms:

- Style Manger color allocation.

  You can dynamically change color for HP VUE applications and other cooperating applications.

  The **foreground** and **background** colors set by Style Manager are available to non-cooperating applications.

- Directly setting a color resource for an application.

  You can set color resources through resource files, **xrdb**, or command line arguments.

  Use this method when a specific color is needed for an element of an application, regardless of the current palette.

HP VUE provides palettes of pre-defined colors. The number of colors in a palette depends on the type of display you have.

For a client to respond to Style Manager color changes, the following must be true:

- The client must be an HP VUE client or an HP OSF/Motif 1.1 client.

  Clients written with other toolkits cannot change color dynamically in response to Style Manager changes. Color changes do not take effect until the client is restarted.

- There must be no other specific color resources applied for the client. This includes user-specified resources, app-defaults, and resources built into the application.

  Any colors specifically given in resources are not changed in response to Style Manager changes.

Clients can specify special resources to use certain colors within an HP VUE palette.

**See Also**
- "Setting Personal Resources for an Application" in Chapter 25 for information about checking and setting resources.
- Chapter 9 covers how to change color or add palettes using Style Manager.

# To set palettes

- Use the Style Manager Color dialog to add, delete, or change color palettes.

A palette consists of up to eight color sets. The number of color sets is determined by the `colorUse` resource. (Refer to "To set the colorUse resource".)

The Style Manager's Color Palette dialog box shows the name of the selected palette and a button showing the background color for each color set within the palette. You can use this dialog box to add palettes or change color sets within a palette.
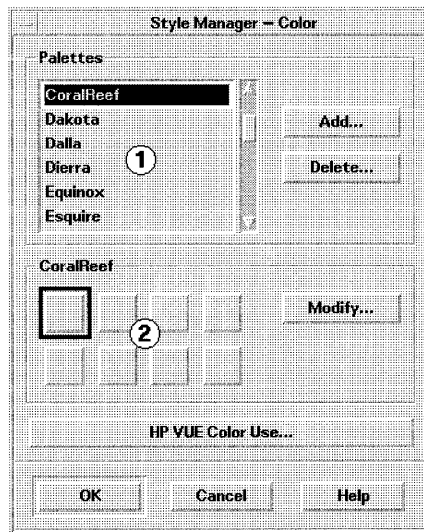


**Use the Color Dialog to select and display palettes.**

where:

①                Palette names.

②                Background color of the colorsets within the chosen palette.

A file exists for each palette. The `paletteDirectories` resource specifies the directories containing palette files. By default, this resource contains:

- **/usr/vue/palettes** contains files for the palettes shipped with HP VUE.

- /*HomeDirectory*/ **.vue/palettes** contains palettes you create or modify.

When you select a palette with Style Manager, a resource is written to your current resources in the RESOURCE_MANAGER property. Session Manager uses this resource to restore the current palette at the beginning of the next session. If the **colorUse** resource value is **B_W**, the palette resource is **monochromePalette**, otherwise the palette resource is **colorPalette**.

**See Also**
- "Customizing Screen Appearance" in Chapter 9 for information about adding or changing palettes and color sets.

## Color sets

A color set is composed of up to five colors. Each color set is represented by a color button in the Style Manager Color dialog (showing the background color). These five colors represent the following display component resources:

| | |
|---|---|
| **foreground** | The foreground of an application window or window frame. It is always black or white. This is generally used for text within windows and titles. |
| **background** | The background of the application controls and the window frame. |
| **topShadowColor** | The color of the top and left bevels of application controls (such as push buttons) and window frames. |
| **bottomShadowColor** | The color of the bottom and right bevels of application controls and window frames. |
| **selectColor** | The color of certain controls, such as depressed buttons, editable areas, and scroll bars. |

When a palette is selected, the **foreground** and **background** resources are written to the RESOURCE_MANAGER property. Non-HP VUE applications can access the new value the next time they are started.

# To set the colorUse resource

1. Click the Style Manager control on the front panel.
2. Click the Color Icon.
3. Click the HP VUE Color Use button.

The `colorUse` resource configures how many colors HP VUE uses.
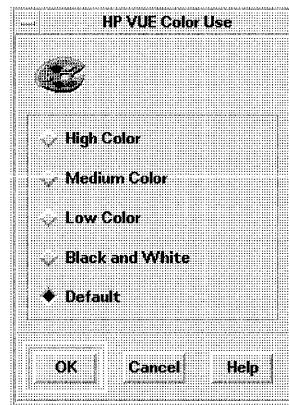


**Control the number of colors HP VUE uses with the Color Use Dialog.**

The "default" value of the resource is determined based on the type of display you have. The value of this resource affects the number of colorsets in a palette. Other resources affect the number of colors used to make shadows.

The value of the `colorUse` resource also affects whether you can use multi-color icons.

| Value | Description |
|-------|-------------|
| B_W   | monochrome displays (1-3 color planes) |
|       | number of color sets: 2 |
|       | maximum number of colors: 2 |
|       | default number of colors: 2 |
|       | no multi-color icons |

LOW_COLOR
low-resolution displays (4-5 color planes)
number of color sets: 2
maximum number of colors: 12
default number of colors: 8
no multi-color icons

MEDIUM_COLOR
medium-resolution displays (6 color planes)
number of color sets: 4
maximum number of colors: 22
default number of colors: 22
no multi-color icons

HIGH_COLOR
high-resolution displays (7 or more color planes)
number of color sets: 8
maximum number of colors: 42
default number of colors: 42
multi-color icons allowed

The "default" choice allows the most display-independence, that is, you need not reset the **colorUse** resource when you run on other displays. HP VUE chooses the correct value for that display.

Multi-color icons use additonal colors.

**See Also**

- "Getting Started with Icon Editor" in Chapter 16 for information about creating and using multi-color icons.

## To set pixmap shadows

1. Use EditResources action to set the **shadowPixmaps** resource.

2. Choose Restart Workspace Manager from the Workspace menu.

Shadow pixmaps create shades of colors without adding additional colors to the color map. Shadow pixmaps can be used as a tile that provide a visual texture by "mixing" the background and foreground (black or white) colors into a color pattern.

The **shadowPixmaps** resource directs HP VUE to replace the two shadow colors with pixmaps. Pixmaps use combinations of the foreground and background colors, thereby reducing the number of needed colors by two.

| Value | Description |
|---|---|
| True | HP VUE creates a **topShadowPixmap** and **bottomShadowPixmap** to use instead of the shadow colors. |
| False | **topShadowColor** and **bottomShadowColor** from the palette are used. |

The default value for **shadowPixmaps** depends on the **colorUse** resource you have.

| Display | Default value |
|---|---|
| B_W | True |
| LOW_COLOR | True |
| MEDIUM_COLOR | False |
| HIGH_COLOR | False |

**See Also**

- Chapter 25 for information about how and where to set resources.

---

# To set the foreground color

- Use the EditResources action to set the **foregroundColor** resource.

The **foregroundColor** resource specifies how the **foreground** resource is configured in a palette.

| White | **foreground** is set to white. |
|---|---|
| Black | **foreground** is set to black. |
| Dynamic | **foreground** is set to either black or white, depending on the value of **background**. For instance, white letters on a yellow background would be hard to read, so the system chooses black. |

If **foregroundColor** is set to either Black or White, the number of colors in the color set is reduced by 1, and the foreground will not change in response to changes in the background color.

The default value for **foregroundColor** is **Dynamic**, except where the value of **colorUse** is B_W.

**See Also**

- Chapter 25 for information about how and where to set resources.

---

# To use dynamic color

1. Use the EditResources action to set the **dynamicColor** resource to **True**.

2. Choose Restart Workspace Manager from the Workspace menu.

The **dynamicColor** resource controls whether applications change color dynamically, that is, the clients change color when you switch palettes.

| Value | Description |
|-------|-------------|
| **True** | Clients change color when a new palette is selected. This is the default value. |
| **False** | Clients do not change color. Any new client started after the new palette uses the new colors, but already- running clients do not change. |

Clients that cannot change colors dynamically (for instance, if a color was specified in the command line) allocate different cells in the color map than clients that can change colors dynamically, even if you see the same color. Setting **dynamicColor** to **False** can reduce the number of colors your display must manage, since all clients can share the same color cells.

A client willing to dynamically change color can either specify no color resources or map specific color sets to special resources.

**See Also**

● Chapter 25 for information about how and where to set resources.

---

# To compute number of colors a palette uses

1. Multiply the number of color sets in the palette by the number of colors within each color set.

2. Add 2 (for black and white).

The minimum number of colors used by Style Manager is the number of colors used by a palette. This number depends on the values for these resources:

● `colorUse` (Refer to "To set the colorUse resource".)

● `shadowPixmaps` (Refer to "To set pixmap shadows".)

● `foregroundColor` (Refer to "To set the foreground color".)

● `dynamicColor` (Refer to "To use dynamic color".)

The following table shows the maximum number of colors for each type of display:

B_W                 2 (**foreground** and **background** only)

LOW_COLOR           12 (2 color sets times 5 colors plus black and white).

MEDIUM_COLOR        22 (4 color sets times 5 colors plus black and white).

HIGH_COLOR          42 (8 color sets times 5 colors plus black and white).

However, with the following configuration . . .

```
*colorUse:          MEDIUM_COLOR
*shadowPixmaps:     True
*foregroundColor:   White
```

then you would have only 10 colors in your palette: 4 color sets times 2 colors in each set (**background** and **selectColor**) plus black and white.

If you had the following configuration . . .

```
*colorUse:          MEDIUM_COLOR
*shadowPixmaps:     False
```

---

```
*foregroundColor:        Dynamic
```

and used the same color for the two shadow colors, you would have 18 colors in your palette (4 color sets times 4 colors in each set plus black and white).

# To limit the number of colors

To limit the number of colors HP VUE uses, you can do one of the following:

- Let Style Manager handle color resources as much as possible. Limit client color resource specifications in resource files.

- Change the `colorUse` resource to a lower-resolution value. (Refer to "To set the colorUse resource".)

- Replace the top and bottom shadow colors with pixmaps using the `shadowPixmaps` resource. (Refer to "To set pixmap shadows".)

- Set the foreground color to White or Black using the `foregroundColor` resource, rather than computing it dynamically. (Refer to "To set the foreground color".)

- Turn off dynamic color allocation for applications using the `dynamicColor` resource. (Refer to "To use dynamic color".)

# To map color to screen elements

1. Use the EditResources action to set the color set ID resources.

2. Choose Restart Workspace Manager from the Workspace menu.
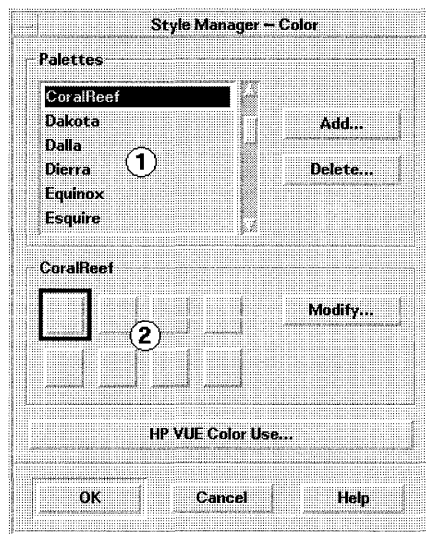
The Color Dialog shows color sets.

①      shows the palette selected.

②      shows the color sets within that palette, sets 1 through 4 on the top line and sets 5 through 8 on the bottom line.

Color sets are mapped to various display elements through resources. HP VUE makes the following assignments:

| Resource | Display element (default color set) |
|---|---|
| `activeColorSetId` | active frame color (1) |
| `inactiveFrameColorId` | inactive frame color (2) |
| `primaryColorSetId` | application's main background areas (3) |
| `secondaryColorSetId` | application's menu bar, menus, and dialog boxes (4) |

Applications are color-coded through the `primaryColorSetId` and `secondaryColorSetId` values, so that it is easy to tell which window a dialog box is associated with, for instance. HP VUE uses the same `primaryColorSetId` for its clients, but provides client-specific `secondaryColorSetId` values.

| HP VUE client | Secondary Color Set ID |
|---|---|
| File Manager | 5 |
| Style Manager | 6 |
| Vueicon | 7 |
| Vuepad | 8 |

Coloring display elements with color set ID's allows the element to change to the new color scheme when a new palette is selected with Style Manager.

**Example**

The following line shows how you would group all `hpterm` windows by using color set 8 for their primary color.

```
hpterm*primaryColorSetId:    8
```

**See Also**

• Chapter 25 for information about how and where to specify resources.

# How colors are specified

Colors can be directly specified in either of two ways:

• Name

• rgb value

Style Manager uses the rgb value of colors it specifies. If you set a color resource directly, you can use either method.

**Color name**

The file `/usr/lib/X11/rgb.txt` lists all the named colors. These names can be used whenever a color is specified. There are many more possible colors than have names, however.

For example,

```
Vuewm*hpterm*iconImageBackground:    DarkSlateGrey
```

**Rgb value**          The syntax for rgb numbers is:

#*RedGreenBlue*

*Red*, *Green*, and *Blue* are hexadecimal numbers, each 1 to 4 digits long, which indicate the amount of that primary color used. There must be the same number of digits for each of the primary colors. Thus, valid color values consist of 3, 6, 9, or 12 hexadecimal digits.

Style Manager uses rgb values when writing color information to its palette files.

For example, black can be specified in any of the following ways:

```
#fff
#ffffff
#fffffffff
#ffffffffffff
```

# Geometry Resources

The geometry of a window is its size and location. A window has a default size (set by the application) and position (set by **vuewm**). You can directly specify the size or location for an application's windows by specifying a **geometry** resource for the application.

The **geometry** resource has the following syntax:

*[width×height][±column][±row]*

where:

| | |
|---|---|
| *width* | The width in characters (for terminal windows) or pixels (for other applications). |
| *height* | The height in lines (for terminal windows) or pixels (for other applications). |
| *±column±row* | The column and row location in pixels. Plus values indicate distance from the upper left corner. Minus values indicate distance from the lower right corner. |

These two options can be combined, as shown in the example.

**Example**

This resource specifies that all **hpterm** windows will be created 80 characters wide and 24 characters high, and be placed at the upper right corner of the display.

```
hpterm*geometry:     80x24-1+1
```

**See Also**

• Chapter 25 for information about how and where to change resources.

# Style Manager Resources

The `componentList` resource describes which components appear in the Style Manager window. All the possible choices appear in the default HP VUE Style Manager Window.

- `Color`
- `Font`
- `Backdrop`
- `Keyboard`
- `Mouse`
- `Audio`
- `Screen`
- `Vuewm`
- `Startup`

The default `componentList` resource for HP VUE Lite lists the following elements:

- `Color`
- `Background`
- `Font`
- `Window`

**To set componentList resource**

1. Use the EditResources action to change the `componentList` resource.
2. Restart vuestyle.

**Example**

This list displays only the Color, Background, and Font components of Style Manager:

```
componentList:     Color Background Font
```

**See Also**

- Chapter 25 for information about checking and modifying resources.

# File Manager Resources

The `rereadTime` resource sets how often File Manager rereads the open directory and displays the results in the File Manager window.

The default value is 2, meaning the directory is read every 2 seconds.

You may prefer to have the File Manager window updated less often if your system is running close to capacity, or if you are doing some activity in another window which causes numerous temporary changes in the active directory.

If you want it to be read less often, update the resource with the time in seconds.

If the value is set to zero, File Manager does not reread the directories. You must manually reread the directory using the Directory menu if you want an updated version.

**To set rereadTime resource**

1. Use the EditResources action to specify the `rereadTime` resource value.

2. Choose Restart Workspace Manager from the Workspace menu.

**See Also**

• Chapter 25 for information about checking and modifying resources.

Font, Color, and Application Resources
**File Manager Resources**

# Non-English HP VUE Sessions

# Non-English HP VUE Sessions

HP VUE offers a number of capabilities in non-English languages. At the highest level, you can obtain HP-UX for certain languages with HP VUE localized as the included user interface.

Since HP VUE is designed with internationalization support as a primary objective, the user interface can be customized in 8-bit and 16-bit languages. Various elements, such as screens, default languages, and icons can be changed to non-English languages. In addition, the menus, online help, and error messages are localizable and available in some languages.

# Fileset Requirements for Asian Lanuages

The following filesets are required to run HP VUE in Asian languages:

| Language | Filesets |
|---|---|
| chinese-s | NLX11-CHS |
| | CHINESES-X11 |
| | NLIO-CHS |
| chinese-t | NLX11-CHT |
| | CHINESET-X11 |
| | NLIO-CHT |
| japanese | NLX11-JPN |
| | JAPANESE-X11 |
| | NLIO-JPN |
| korean | NLX11-KOR |
| | KOREAN-X11 |
| | NLIO-KOR |

# Non-English HP VUE Login

Non-English login through HP VUE is easy. However, for the effect to be noticeable, several key elements should be present. Specific hardware requirements such as keyboards and printers vary by language, character set, and country. Software and fonts further increase the effective localization of your system. To login in a specific language:

1. Use the Options menu on the login screen to select a language.

2. Login normally with your name and password.

# Using a Non-English Terminal Emulator

Successful emulation in any language requires that the specified font be available. Fonts should reside in the **/usr/lib/X11/fonts** directory.

To determine if a given language is supported, read the file **/usr/lib/nls/config** or check for the language in the Options menu at login.

With 16-bit languages such as Japanese, Chinese, and Korean, Native Language Input/Output (NL I/O) must also be installed. Refer to the NL I/O documentation for information and operating instructions.

There are two **hpterm** terminal emulators available for use in HP VUE:

- The **hpterm** that is part of X11 Window System R4.

- The **hpterm** that is part of X11 Window System R5. This new **hpterm** incorporates the Native Language I/O servers.

**See Also**
- The *Using the X Window System* manual contains additional information for localizing terminal emulators and other X Window applications.
- The NL I/O documentation contains additional information about 16-bit language capabilities.

## To Start a Japanese Terminal Emulator

Since the xterm terminal emulator has very limited language capabilities, this example uses hpterm.

The assumptions are that the default language is not Japanese, that you are using the Korn Shell, and that the specified font (jpn.8x18) is in a subdirectory of **/usr/lib/X11/fonts**.

1. From a command line in a Korn Shell terminal window type:

        LANG=japanese hpterm -fn jpn.8x18

In other shells, using the appropriate command, set the value of the LANG environment variable to **japanese** for the HP15 character set. Other code sets use different names.

2. After the new terminal window appears, press the System key to display the softkey labels.

3. Press F8 (Config Keys) and F5 (Terminal Config).

4. Slowly press key F6 until the label for the softkey changes to Japanese.

5. After the cursor changes back to normal, type **date** to verify that the language is Japanese.

## To Start a German Terminal Emulator

Since the German language uses 8-bit characters, **xterm** or **hpterm** may be used to emulate a terminal.

1. From a command line in a Korn Shell terminal window type:

```
LANG=german xterm
```

In other shells set the value of the LANG environment variable to **german** using the appropriate command.

2. If you are using **hpterm**:

   a. After the new terminal window appears, press the System key to display the softkey labels.

   b. Press F8 (Config Keys) and F5 (Terminal Config).

   c. Press key F6 repeatedly until the label for the softkey changes to "Deutsch."

   d. After the cursor changes back to normal, type **date** to verify that the language is German.

# To use the X11 R5 hpterm

**To use the X11 R5 hpterm**

- Run the script /usr/bin/X11/hptermR4toR5.

**To switch back to the X11 R4 hpterm**

- Run the script /usr/bin/X11/hptermR5toR4.

You should switch to the X11 R5 **hpterm** only if you require the Native Language I/O servers and the X11 R5 input method protocols (Xsi, Ximp) for native language input it uses. Using the X11 R5 hpterm reduces system performance.

**CAUTION**

Always use the scripts to switch between the X11 R4 and the X11 R5 **hpterm**s. Manually switching the executables can cause errors that prevent HP VUE from running properly.

# Using Non-English Data

You can create, read, and print non-English files. You can also give files non-English names. However, you should use file names containing only ASCII characters for system administration.

## To Create or Edit a File with Non-English Characters

If you have logged in to HP VUE in a specific language, the HP VUE Text Editor will also be set to that language, so the steps below can be skipped.

If your default language for HP VUE is English and you want to create a file with a non-English character set, you can still use Text Editor, but you must invoke a new instance in your target language.

1. In a Terminal Emulator window, set the LANG and KBD_LANG environment variables to the desired language.

2. In the same Terminal Emulator window, execute:

   `/usr/vue/bin/vuepad -xrm "Vuepad*fontList:` *font*`" -standAlone`

**Example**    The following commands start Text Editor with the Korean language, using a Terminal Emulator with Korn Shell.

```
export LANG=korean
export KBD_LANG=korean
/usr/vue/bin/vuepad -xrm "Vuepad*fontList: kor.8x18" -standAlone
```

You can now enter Korean characters (using NL I/O) if the specified Korean font (kor.8x18) is properly installed. You can also use the Text Editor session to edit a previously created Korean file. The **-standAlone** parameter is necessary to ensure that the editor does not default to the language set at HP VUE login.

## To Print a File with Non-English Characters

There are many variables affecting the successful printing of non-English data. The HP-UX operating system supports printing of all 8-bit character sets. Printing 16-bit characters requires the NL I/O software and a supported printer. Consult the NL I/O documentation for specifics. NL I/O supports printing on various Asian-specific devices, and allows Asian printing on HP LaserJet printers (models II and newer).

# Non-English HP VUE Customization

HP VUE users can modify certain files to achieve a more customized appearance for their culture or language.

## To Localize the Front Panel

If you are running an HP VUE version that has been localized for your language, the date and several other elements of the Front Panel should already appear in that language. To rename a workspace with characters of your language, simply select the Rename Workspace Control and type the new name (using NL I/O if necessary).

If you select a language at login for which HP VUE is not fully localized or installed, the necessary fonts may need to be installed in order to rename the workspaces with the correct characters.

## To Localize Icons

If you want to localize an icon, you will need to edit the icon with the icon editor and then put the result in the correct directory.

HP VUE uses the following search path to locate localized icons, where:

- *language* is the value of LANG.

- *applications* is the application class name.

    /*HomeDirectory*/bitmaps
    /*HomeDirectory*
    /usr/vue/icons/*language*/*application*
    /usr/vue/icons/*application*
    /usr/vue/icons/Common

```
/usr/lib/X11/language/bitmaps/application
/usr/lib/X11/bitmaps/application
/usr/lib/X11/language/bitmaps
/usr/lib/X11/bitmaps
/usr/include/X11/bitmaps
```

**See Also**
- Chapter 16 explains how to create your own icons.

## Localizing actions and filetypes

The search path for action and filetype definition files includes
language-dependent subdirectories of:

- /*HomeDirectory*/.vue/types, the personal database.

- /usr/vue/config/types, the general database.

- /usr/vue/types, the built-in general database. These files may be
  overwritten by subsequent updates.

# System Administration for NLS

This section has information for system administrators. If you are a typical user, you should seldom need to use it.

These topics are discussed:

- How HP VUE sets the LANG environment variable and other environment variables.

- Accessing language-dependent message catalogs and resource files.

- Remote execution in systems on which National Language Support (NLS) is available.

## Setting the LANG Environment Variable

The LANG environment variable must be set in order for HP VUE to use national language support. Setting LANG causes HP VUE to use the operating system's language-sensitive routines for character handling.

You can set LANG to any value supported by the operating system. The NLS directory **/usr/lib/nls** contains an entry for each supported value. The HP VUE message catalogs are located in subdirectories of the NLS directory. Supported languages include simplified and traditional Chinese, French, C-French, German, Japanese, and Korean.

There are three ways to set LANG for HP VUE:

- By a **vuelogin** resource in the **Xconfig** file. The login screen is localized, and LANG is set for all users. This is the only way to change LANG for all displays in multiple-display systems.

- From the Options menu in the login screen. The login screen is localized and LANG is set for the user.

- In the **.vueprofile** file. The login screen is not localized, and LANG is set for the user.

**Using a Vuelogin Resource**

Setting the language by means of a **vuelogin** resource has these effects:

- The **vuelogin** client reads the appropriate message catalog for that language and brings up the localized login screen.

- The LANG environment variable is set to that language for HP VUE sessions for all users. (LANG can be set for individual users by an entry in **.vueprofile**.)

- The resource should set the NLSPATH environment variable appropriately for the chosen language. If this is not the case, or if you want to set NLSPATH yourself, see "Message Catalogs—the NLSPATH Environment Variable" later in this chapter.

The resource is set by placing the following line in **/usr/vue/config/Xconfig**:

> Vuelogin.*host_display*.**language:** *language*

For example, the following line sets LANG to Swedish on display hpcvxpae:0.

> Vuelogin.hpcvxpae_0.language: swedish

**Setting LANG from the Options Menu**

Setting the lanugage from the login screen Options menu causes HP VUE to use that value of LANG for the next session. LANG returns to its default value at the conclusion of the session.

**Setting LANG in .vueprofile**

User-specific environment variables for HP VUE sessions can be placed in the file **$HOME/.vueprofile**.

- If you use **sh** or **ksh**:

  > LANG=*language*
  > export LANG

- If you use **csh**:

  > setenv LANG *language*

# To Set Other NLS Environment Variables

This section covers other NLS environment variables. These variables are not affected by the **vuelogin** language resource nor by the login screen option menu.

- Set them in **/usr/vue/config/Xsession** to set system-wide variables.

- Set them in **$HOME/.vueprofile** for user-specific variables.

In the following examples, **%L** is translated by the system into the value of the LANG variable; **%N** is translated into the value of the application's class name.

**Message Catalogs—The NLSPATH Environment Variable**

The NLSPATH environment variable determines the paths applications search for NLS message catalogs. HP VUE clients place NLS message catalogs in the NLS directories **/usr/lib/nls/$LANG** (non-Domain/OS systems) or **/usr/nlslib** (Domain/OS). Both LANG and NLSPATH must be set in order to use those message catalogs.

The proper value of NLSPATH depends on whether message catalogs exist for the current value of LANG.

- To use the message catalogs for the language to which LANG is set, set NLSPATH to:

    **/usr/lib/nls/%L/%N.cat:/usr/lib/nls/C/%N.cat:$NLSPATH**

- If no message catalogs exist for the language to which LANG is set, set NLSPATH to:

    **/usr/lib/nls/C/%N.cat:$NLSPATH**

**Setting Language-Dependent App-Defaults**

The default location for the **app-defaults** for HP VUE and X clients is **/usr/vue/app-defaults/%L**, where **%L** is the the value of LANG. For example, if LANG is set to **swedish**, then applications will look for their **app-defaults** in **/usr/vue/app-defaults/swedish**. If LANG is not set, **%L** is ignored, and applications look for their **app-defaults** in **/usr/vue/app-defaults**.

The **XUSERFILESEARCHPATH** environment variables provides the ability to change the location of **app-defaults**. For example, if it were desirable to move **app-defaults** to **/users**, then **XUSERFILESEARCHPATH** could be set to **/users/app-defaults/%L/%N**.

If you set XUSERFILESEARCHPATH in $HOME/.vueprofile, the value
applies to all HP VUE and X clients you run. Non-clients will not find their
resource files unless you link or copy them into the directory specified by
XUSERFILESEARCHPATH.

For additional information about localized resources, refer to the discussions
of the resource database in the Xt Intrinsics documentation.

**Setting the KBD_LANG**
**Environment Variable**

> **NOTE**
>
> The ability to support native language keyboards depends on the operating system. Other operating
> systems may not support character sets for Asian languages.

HP VUE allows you to override the physical keyboard attached to the HP-HIL.
For example, a USASCII keyboard with KBD_LANG set to czech will behave
like a Czech keyboard.

KBD_LANG can be set after the server has started. A convenient place for
setting the variable system-wide is in **/usr/vue/config/Xsession**. The
following lines set KBD_LANG appropriately for languages that require the
keyboard to be remapped. Insert theses lines in **Xsessions** after the line
**export EDITOR LOGNAME MAIL TERM**:

```
if [ ! -z "$LANG" ]
then
    case $LANG in

            bulgarian | czech  | hebrew | hungarian | \
            japanese  | korean | polish | rumanian  | \
            russian   | serbocroatian)
                    KBD_LANG=$LANG
                    export KBD_LANG;;
            chinese-t) KBD_LANG=t_chinese;;
                    export KBD_LANG;;
            chinese-s) KBD_LANG=s_chinese
                    export BKD_LANG;;
            *);;
    esac
fi
```

The NLIO processes for Asian users start only when either the physical
keyboard is Asian or KBD_LANG is set to an Asian language.

**Language-Dependent Bitmaps—the XBMLANGPATH Variable**

The **XBMLANGPATH** variable specifies the search path for language-dependent
bitmaps. It lists the paths for bitmaps in this order:

1. User-specific bitmaps.

2. System bitmaps listed in the **XmpGetIconFileName(3X)** man page.

3. Append:

   **/usr/vue/icons/**

   This ensures that you will get the non-localized bitmaps, where necessary.

## Other Language-Dependent Resource Files

When LANG is set, HP VUE uses the following language-dependent default resource files:

- `/usr/vue/config/%L/sys.session`

- `/usr/vue/config/%L/sys.resources`

- `/usr/vue/config/%L/sys.vuewmrc`

## Editing in HP VUE

All characters that can be entered from an HP keyboard into a terminal emulator window can also be entered into HP VUE text entry areas.

## Remote Execution and NLS

You can invoke localized HP VUE applications on any remote execution host that has a similarly localized HP VUE installation. The values of the NLS-related environment variables on the host that is invoking the application are passed to the remote host when the application is started. However, the environment variables do not contain any host information. Thus, the message catalogs and application resource files must be in the same locations on both systems unless the `$HOME/.softenv` file on the remote host specifies the location of the files on the remote system.

# Reference

This sections covers the supported langauges, and the location of localized icon and font files.

## Supported Languages

Read the file **/usr/lib/nls/config** to list system supported languages.

## Localized Icon Search Path

HP VUE uses the following search path to locate localized icons, where:

*language* is the value of LANG.

*applications* is the application class name.

/*HomeDirectory*/**.vue/icons/***application*/**Color**
/*HomeDirectory*/**.vue/icons/***application*
/*HomeDirectory*/**.vue/icons/Color**
/*HomeDirectory*/**.vue/icons**
/usr/vue/icons/*language*/*application*/**Color**
/usr/vue/icons/*language*/*application*
/usr/vue/icons/*language*/**Color**
/usr/vue/icons/*language*

**See Also**
- "Where to put icon files" in Chapter 16

## Font Locations

Fonts are located in the **/usr/lib/X11/fonts** directory.

Non-English HP VUE Sessions
**Reference**
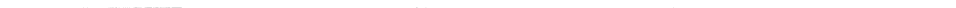
# For Previous HP VUE 2.01 or X11 Users

# For Previous HP VUE 3.0 or X11 Users

If you've previously used HP VUE 2.01, there are several configuration issues you may encounter when moving to HP VUE 3.0:

- New locations for configuration files.

- Differences in session management.

- New configuration syntax and features for the Front Panel.

- Toolboxes replace the applications directory.

- New syntax and locations for action and filetype definitions.

- Some backdrops have been replaced.

If you are switching to HP VUE 3.0 from X11 Window System, you will need to know how HP VUE Session Management replaces certain X11 startup files, and how environment variables are obtained in HP VUE.

# New Locations for HP VUE Configuration Files

System-wide HP VUE files are now located in subdirectories of `/usr/vue`.

| Directory | Contents |
|---|---|
| `/usr/vue/bin` | HP VUE executables. |
| `/usr/vue/config` | Configuration files for Login Manager, Session Manager, and Workspace Manager. |
| `/usr/vue/types` | Built-in actions and filetypes. |
| `/usr/vue/app-defaults` | Application defaults for HP VUE clients. |
| `/usr/vue/icons` | HP VUE system-wide icons. |
| `/usr/vue/palettes` | Palettes. |
| `/usr/vue/VUE2.0config` | Saved configuration files for HP VUE 2.01. |
| `/usr/vue/examples` | Printer configuration examples, old backdrops. |

**Login Manager Configuration Files**

During a system update, HP VUE 2.01 Login Manager configuration files are handled as follows:

- Customizations made in the **Auto Customization** section of the file `/usr/lib/X11/vue/Vuelogin/Xsession` (HP VUE 2.01) are copied into `/usr/vue/config/Xsession` (HP VUE 3.0). In addition, a symbolic link is created between the HP VUE 2.01 **Xsession** file and the HP VUE 3.0 **Xsession** file.

- The other HP VUE 2.01 Login Manager configuration files are moved to `/usr/vue/VUE2.0config`. Customizations in these files must be manually copied to the HP VUE 3.0 Login Manager files in `/usr/vue/config`.

# Session Management (HP VUE 2.01 to HP VUE 3.0)

HP VUE 3.0 sessions are saved in a different location and with different syntax than HP VUE 2.01 sessions. This allows a HP VUE 3.0 session to run on displays with different resolutions and color capabilities.

## Display-dependent and display-independent sessions

HP VUE 2.01 stored sessions in a display-dependent fashion—the session information included information about size and color of displayed objects that varied for displays of different resolution and color capacity.

If a user has previously logged into HP VUE 2.01, either or both of these sessions may exist:

1. Current session: */HomeDirectory/*.vue/*display:host/*current

2. Home session: */HomeDirectory/*.vue/*display:host/*home

If these directories exist, HP VUE 3.0 will continue to use them, rather than the HP VUE 3.0 */HomeDirectory/*vue/sessions directory. However, the HP VUE 3.0 sessions will be saved using display-independent syntax.

## The first HP VUE 3.0 session (HP VUE 2.01 to HP VUE 3.0)

Since display-dependent sessions have precedence over display-independent sessions, the HP VUE 2.01 session is restored when the user logs into HP VUE 3.0 the first time.

When the user logs out, the session is saved to the same display-dependent directory:

- The session data saved is saved using display-independent syntax.
- The file *CONVERTED* is created in the directory to indicate that the session has been converted to HP VUE 3.0 syntax.

## To create a display-independent session

1. Log in and out of an HP VUE 3.0 session. This creates session files in the display-dependent (HP VUE 2.01) directory that use display-independent (HP VUE 3.0) syntax. The file *CONVERTED* is created to let you know the conversion has taken place.

2. Copy the */HomeDirectory/*.vue/*display* directory to the */HomeDirectory/*.vue/sessions directory.

3. Remove or rename the */HomeDirectory/*.vue/*display* directory.

Creating a display-independent session allows you to use distributed session management, allowing you to obtain the same session regardless of which system you use to log in.

**See Also**

- "Providing Networked Sessions" in Chapter 22

# Moving an HP VUE 2.01 Front Panel to HP VUE 3.0

HP VUE 3.0 provides major enhancements to the Front Panel

- The new syntax is display-independent. You can use the same Front Panel for different display resolutions and color capabilities.

- The new syntax makes it easier to modify the Front Panel. Geometries are handled automatically, and definitions of rows and controls are easier to read and edit.

- Subpanels extend the functionality of the top-row controls.

- The Toolbox control and Toolboxes subpanel provide easy access to the new HP VUE 3.0 toolboxes.

You can use the ConvertVuewmrc action in the System_Admin subdirectory of the General Toolbox to convert a personal HP VUE 2.01 Front Panel to HP VUE 3.0 syntax.

**See Also**

- "To convert the syntax of your HP VUE 2.01 Front Panel to HP VUE 3.0" explains how to use ConvertVuewmrc.

# What happens to your HP VUE 2.01 Front Panel when you log into HP VUE 3.0

**NOTE**

If you want to use your HP VUE 2.01 Front Panel, you should convert it to the new syntax using the ConvertVuewmrc utility. Using the HP VUE 2.01 syntax will have a negative effect on system performance.

HP VUE 3.0 will continue to read your old Workspace Manager configuration file */HomeDirectory/*`.vue/vuewmrc` if it exists. Therefore, if you make no manual changes to configuration files after updating, your HP VUE 3.0 session will use your HP VUE 2.01 Front Panel.

Displayed in HP VUE 3.0, your old Front Panel will contain the same rows and controls as it did in HP VUE 2.01, but will look different. Like the HP VUE 3.0 Front Panel, the control icons will be embedded into a flat surface (no separators between controls). Controls built into HP VUE 2.01 will use the new HP VUE 3.0 icons.

# To switch from the HP VUE 2.01 Front Panel to the HP VUE 3.0 Front Panel

1. Rename */HomeDirectory/*`.vue/vuewmrc` a different file name—for example `vuewmrc.201`.

2. Choose Restart Workspace Manager from the Workspace Menu.

You can now modify the HP VUE 3.0 Front Panel.

**See Also**

- Chapter 17 covers modifying your Front Panel.

# To convert the syntax of your HP VUE 2.01 Front Panel to HP VUE 3.0

> **NOTE**
>
> Without extensive editing, a converted HP VUE 2.01 Front Panel will not take advantage of Front Panel enhancements such as subpanels, built-in pixmaps, and the HP VUE 3.0 toolboxes. If you want to use these enhancements, you should switch to the HP VUE 3.0 front panel.

- Double-click ConvertVuewmrc in the System_Admin subdirectory of the General Toolbox.

ConvertVuewmrc runs a utility **/usr/vue/bin/vuefp2to3**. The utility first copies your /*HomeDirectory*/.**vue/vuewmrc** to **v20_vuewmrc**, and then converts **vuewmrc** to the new syntax.

Converting the syntax of your HP VUE 2.01 Front Panel enhances system performance, and makes it easier to modify your Front Panel later.

# Accessing applications (HP VUE 2.01 to HP VUE 3.0)

In HP VUE 3.0, three separate toolboxes—Personal, General, and Networked—replace the applications directory.

The effect of this change depends on whether or not you switch to the HP VUE 3.0 Front Panel.

There are also differences in the may HP VUE 3.0 provides permission for remote applications to access the local display.

# How HP VUE 3.0 provides display access for remote applications

In order for a remote applicaton to run on a local system, remote system must have permissin to use the local system's display. This permission is called "xhost" permission.

HP VUE 3.0 does not use the same xhost permission mechanism as HP VUE 2.01. Style Manager no longer maintains a personal xhost list; instead, actions running remote applications automatically provide xhost permission to the application host.

If the application is run by a command, rather than an action, you must explicitly provide permission for the remote host to access the local display:

- For system-wide permission: add the remote system to `/etc/X0.hosts`.

- For personal permission: add an **xhost** command to
  `/HomeDirectory/.vue/sessions/sessionetc`.

# To display HP VUE 3.0 tools from the HP VUE 2.01 Front Panel

If you continue to use your HP VUE 2.01 Front Panel, you will be able to access the contents of the HP VUE 3.0 toolboxes through your applications directory.

- Choose the applications directory control on the top row of the Front Panel.

- Double-click:

  □ GeneralTools to display the contents of the HP VUE 3.0 General Toolbox.

  □ NetworkTools to display the contents of the HP VUE 3.0 Network Toolbox.

The personal_apps directory is equivalent to the Personal Toolbox.

# To display HP VUE 2.01 tools from the HP VUE 3.0 toolboxes

If you switch to the HP VUE 3.0 Front Panel, use this procedure to access your HP VUE 2.01 applications directory:

1. Open the Personal Toolbox by choosing the Toolbox control on the top row. Your Personal Toolbox will contain the tools that previously were in personal_apps.

2. To see the rest of your HP VUE 2.01 tools, double-click VUE2.0Tools.

To provide consistent locations for tools, You may want to reconfigure your HP VUE 2.01 system-wide and remote actions to use the General and Network toolbox.

**See Also**
- Chapter 6 covers using the HP VUE 3.0 toolboxes.

# To move the HP VUE 2.01 local_apps to the General Toolbox

1. Log in as superuser.

2. Move the action and filetype definition (`*.ad` and `*.ft`) from `/usr/local/lib/X11/vue/types` to `/usr/vue/config/types`.

3. Use **vueconvert** to convert the database files to the HP VUE 3.0 syntax (see "To convert HP VUE 2.01 database files to HP VUE 3.0 syntax").

4. Move the action icon files from `/usr/local/lib/X11/vue/types/tools` to `/usr/vue/config/types/tools`.

If your system contains system-wide actions that were added to `/usr/lib/X11/vue/types`, you can move those to the General Toolbox too.

# To move HP VUE 2.01 remote actions to the Network Toolbox

1. Configure the local and remote systems properly to use remote actions.

   See "To configure the database (exporting) host" in Chapter 22 and "To configure the local (importing) host" in Chapter 22 for information on how to do this.

2. On the exporting system (the system containing the action definition):

   a. Move the action and filetype definition (`*.ad` and `*.ft`) files to `/usr/vue/config/export`.

    b. Use **vueconvert** to convert the database files to the HP VUE 3.0 syntax (see "To convert HP VUE 2.01 database files to HP VUE 3.0 syntax").

    c. Move the action icons (the executable files with the same name as the action, located in **tools** subdirectories) to **/usr/vue/config/export/tools**.

3. On the importing systems:

    a. Create a directory in the Network Toolbox that is a link to the exported action icons:

```
ln -s /nfs/remote_host/usr/vue/config/export/tools \
        /usr/vue/config/import/remote_host
```

    b. Remove the host from the Vue.TypesDir search path, which is usually set in **/usr/lib/X11/app-defaults/Vue**.

# Filetypes and Actions (HP VUE 2.01 to HP VUE 3.0)

There are several major differences between HP VUE 3.0 and HP VUE 2.01 actions and filetypes:

- Update replaces the actions and filetypes built into HP VUE 2.01.

- HP VUE 3.0 uses a different file name convention and different syntax for definitions. Although HP VUE 3.0 can continue to use the old syntax, you should convert the actions and filetypes to the new syntax using the **vueconvert** utility. Using the old syntax has a negative effect on system performance.

- Distributed actions (actions defined on one system that are invoked on other systems) are configured and accessed differently.

## What happens to HP VUE 2.01 actions

> **NOTE**
>
> HP VUE 3.0 preserves your HP VUE 2.01 personal actions/filetypes and the system-wide actions/filetypes that were defined by the system administrator. We strongly recommend that you convert these to HP VUE 3.0 syntax and then remove the files containing the old syntax. Maintaining the old syntax has a negative effect on system performance.

**Built-in actions and filetypes**

HP VUE 2.01 built-in actions (**system_apps**) and filetypes are removed. The new built-in actions and filetypes are in **/usr/vue/types**.

If you added action or filetype definitions to **/usr/lib/X11/vue/types**, those files are preserved.

**System-wide actions and filetypes**

System-wide actions/filetypes defined by the system administrator in `/usr/local/lib/X11/vue/types` remain in place but are not autmoatically converted to the new HP VUE 3.0 syntax.

This directory is along the database search path, so the definitions will be found.

**Personal actions**

HP VUE 3.0 uses the same location for personal action/filetype definitions: /*HomeDirectory*/ `.vue/types`. HP VUE 2.01 actions defined there remain in place but are not automatically converted to the HP VUE 3.0 syntax.

# To convert HP VUE 2.01 database files to HP VUE 3.0 syntax

**NOTE**

HP VUE 3.0 will continue to recognize personal and local HP VUE 2.01 database files. However, you should convert them to enhance system performance and make it easier to maintain your database.

1. Switch to the directory containing the old database (`.ad` or `.vf`) file(s).
2. Execute:

   **vueconvert** *filename* [*filename* ... ]

   A new database file will be created in the same directory with a `*.vf` file name.
3. Remove the old file or rename it so that it does not use a filename `*.ad`, `*.ft`, or `*.vf`.

**See Also**

- Comments in the `vueconvert` script (`/usr/vue/bin/vueconvert`) contain additional information about running and modifying the script.

# Certain HP VUE 2.01 Backdrops are Replaced

HP VUE 3.0 provides a new set of backdrop icons. This new set contains some, but not all, of the old backdrops, The old backdrops are moved to `/usr/vue/examples/vuebackd.sh.z`.

To use an old backdrop:

1. Create a temporary directory /*HomeDirectory*/`bdrops`.

2. Copy `/usr/vue/examples/vuebackd.sh.z` to the new directory.

3. Execute:

   ```
   uncompress vuebackd.sh.z
   ```

   to uncompress the file.

4. Execute:

   ```
   unshar vuebackd.sh
   ```

   to unpack the set of backdrops into separate files.

5. Set up a personal backdrop directory and copy the old backdrop files you want to that directory.

**See Also**

- "To customize backdrops" in Chapter 24 explains how to create a personal backdrop directory

# Moving from X11 to HP VUE 3.0

If you previously used X11, you will need to know about these differences:

- How HP VUE obtains environment variables.
- How HP VUE otains and uses resources, and how to integrate your X11 resources into HP VUE.
- How HP VUE replaces your X11 startup script.
- How to integrate your X11 startup script into HP VUE Lite.

## To use your X11 environment variables

HP VUE does not automatically read `.profile` or `.login`.

See "To use a shell environment file with '.vueprofile'" in Chapter 14 for information on how to integrate your X11 environment into HP VUE.

## To integrate X11 resource files into HP VUE

1. Copy your X11 resource file to a another file.
2. Edit the new file to:
   - Remove color resources, unless the specific color is absolutely necessary for the application.
   - Remove font resources, unless the specific font is absolutely necessary for the application.
3. Log into the type of session in which you want the resouces—either HP VUE or HP VUE Lite.

4. Merge the edited X11 resources into the current session resources using:

   - The EditResources action in the System_Admin subdirectory of the General Toolbox.

   - *Or*, execute the command:

     `xrdb -merge -nocpp` *filename*

5. New resources take effect the next time the application is started. You can log out and back in to restart all the applications at once.

# To replace your X11 startup script for HP VUE sessions

**NOTE**

This information does not apply to HP VUE Lite (see "To replace your X11 startup script for HP VUE Lite sessions").

1. Log into an HP VUE session.

2. Use a Terminal Emulator to start the applications that were previously started from your X11 startup script. Do not start a window manager.

3. You now have two choices:

   - You can proceed with your session, opening and closing applications as necessary. Your next session will start the applications running at the end of your previous session. This is the default behavior.

   - You can save and use a "home" session. This allows you to see identical sessions each time you log in.

**See Also**

- "Using a Home Session" in Chapter 2 explains how to save and use a home session.

## To replace your X11 startup script for HP VUE Lite sessions

1. If *HomeDirectory*/.vue/sessions/lite/vue.session does not already exist, create it by copying /usr/vue/config/sys.ses.lite. Give the new file write permission.

2. Copy lines that start applications from your X11 startup script to vue.session. Do not copy the line that starts your window manager. Run the applications in the background (each command line must end with &).

For Previous HP VUE 3.0 or X11 Users
**Moving from X11 to HP VUE 3.0**

# 29

Troubleshooting

# Troubleshooting

There are several ways to troubleshoot HP VUE problems:

- Read the error messages recorded in the error-monitoring files. There are different files for errors generated during update, Login Manager startup, and HP VUE sessions.

- Several utilities help you diagnose configuration and database errors.

# If HP VUE (Login Manager) Doesn't Start

If your system is properly configured to automatically run HP VUE, you will see the HP VUE login screen when your system is booted.

If your system is not configured properly, you may instead see a console login prompt, or, possibly, nothing at all.

If this happens:

- Log in to your system console, if present. If you have no console, you must log in from a remote system.

- Execute the utility:

      /usr/contrib/bin/X11/dr_vue | more

* Examine the displayed output from dr_vue. It detects errors in various configuration files.

There are several reasons that HP VUE might fail to start:

- The fileset containing HP VUE is not installed properly.

- The file that controls system startup, /etc/inittab, may not be configured properly.

- The X server has not started properly. The X server is specified in the file /usr/vue/config/Xservers.

**See Also**
- Chapter 11 covers how to enable and disable HP VUE.

# Files that monitor errors

When HP VUE fails to behave as expected, you should open the appropriate error-monitoring file:

- **/tmp/update.log**: Lists errors encountered during the update process.

- **/usr/vue/config/Xerrors**: The system-wide errorlog. This file records errors that occur during login.

- **/***HomeDirectory***/.vue/errorlog**: The user's error log. A new errorlog is created each time a user logs in. Errors from the previous two sessions are saved in **errorlog.old** and **errorlog.older**.

- **/tmp/errorlog.***login_name*: This file is created if the system cannot write errors to **/***HomeDirectory***/.vue/errorlog**.

# Troubleshooting utilities

Utilities shipped with HP VUE let you:

- Constantly monitor session errors.
- Diagnose common configuration problems.
- Diagnose and examine the actions and filetypes database.

## Monitoring errors

The WatchErrors action opens a terminal window that monitors
/*HomeDirectory*/.vue/errorlog. The window can be minimized, and will
automatically normalize itself whenever anything is written to the errorlog.

WatchErrors can be run by:

- Double-clicking WatchErrors in the Utilities subdirectory of the General
  toolbox.
- Executing the following command in a Terminal Emulator command line:

      vueaction WatchErrors

## Diagnosing common errors

The dr_vue utility checks many common sources of problems, particularly
problems with network configuration. The utility can be started by:

- Executing /usr/contrib/bin/X11/dr_vue from a Terminal Emulator
  command line.
- Double-clicking the Dr_Vue action in the Unsupported subdirectory of the
  General Toolbox.

# Action and filetype errors

The **vuetypes** utility locates errors in the database of actions and filetypes. The utility can be started by

- Executing **/usr/vue/bin/vuetypes** from a Terminal Emulator command line.
- Double-clicking ListVueTypes in the Unsupported subdirectory of the General Toolbox.

**See Also**

- "Troubleshooting Actions and Filetypes" in Chapter 19.

# If the Broadcast Message Server Fails to Start

If the Broadcast Message Server (BMS) fails to start, Session Manager cannot start a session. Instead, the system displays an error message and exits back to the login screen.

If this happens:

1. Log in using No Windows mode or a failsafe session.

2. Make sure the line that starts the server in **/usr/vue/config/Xservers** uses the actual host name of the system. (The * in the first field is also allowed.)

3. If you previously used Softbench and have a personal */HomeDirectory/*.**softinit** file, you may need to remove the file or edit it to include the HP VUE tools (see the instructions below)

4. Check to see that **/usr/softbench/bin/softmsgsrv** exisits and is executable.

**To edit an existing personal .softinit file**

If you had a customized */HomeDirectory/*.**softinit** file before you installed HP VUE, you should:

1. Rename your */HomeDirectory/*.**softinit** file (for example, **.softinit.prevue**.

2. Copy **/usr/softbench/config/softinit** to */HomeDirectory/*.**softinit**. Give the new file write permission.

3. Copy any personal customizations from **.softinit.prevue** to the new */HomeDirectory/*.**softinit** file.

**See Also**

- "Moving to HP VUE from Softbench" in Chapter 13

# Glossary

# Glossary

**accelerator**
A key or combination of keys provided as a shortcut for choosing a command. Accelerators are most commonly used with menu commands, but may also be used for other frequent actions.

**action**
An HP VUE construct used to provide a user interface for applications and other commands. Actions are defined in a database of files located along the database search path.

**action icon**
An icon representing an HP VUE action in a File Manager or Toolbox window. An action icon is displayed by creating an empty executable file and giving it the same name as the action it represents.

**action server**
A host computer that provides access to a collection of actions.

**application**
See software application.

**application server**
A host computer that provides access to application software.

**application window**
See window.

**bitmap**
An image stored in a raster format. Usually the term *bitmap* implies that the image is limited to exactly two colors (a foreground and a background color). An image that uses more than two colors is usually called a *pixmap*.

**bitmapped font**
A bitmapped font is made from a matrix of dots. See **font**.

**button**

A control that executes a command, starts an action, or selects an option. There are three main kinds of buttons: push buttons, toggle buttons, and radio buttons.

**button binding**

Association of a mouse button operation with a particular behavior.

**click**

To press and release a mouse button. The term comes from the fact that pressing and releasing the buttons of most mice makes a clicking sound.

**client**

See software application.

**command line prompt**

A command line prompt shows that the computer is ready to accept your commands. Each terminal emulation window has a command line prompt that acts just like the command line prompt that would be shown if your computer was not running HP VUE. Usually the command line prompt is %, >, or $. You can find the command line prompt by pressing (Return) in a terminal emulation window.

**control**

A graphical user interface component that enables you to manipulate an object, select choices, or type information. Menus, text fields, and Front Panel push buttons and indicators are examples of controls.

**current session**

A session that was saved when logging out of the previous session. Returning to the saved session allows you to continue working where you left off. (Some applications may not participate in "session management," so you have have to manually start them after logging in.)

**data host**

A host computer where the data for an action is located.

**database host**

A host computer where an action is defined.

**Desktop**

A place to put files, directories, actions, and applications for quick access. A Desktop is available in each HP VUE workspace. To put an object on

the Desktop, drag its icon from a File Manager or Toolbox window and drop it on the workspace backdrop. An object on the desktop operates just like its original icon. The HP VUE Desktop is *not* available in HP VUE Lite.

**dialog**

A secondary window displayed by an application window. The Color dialog, belonging to Style Manager's main window is an example of a dialog.

**display-dependent session**

A session that can be restored on only a particular display.

**display-independent session**

A session that can be restored on any display, regardless of screen resolution or color capability.

**double-click**

Pressing and releasing a mouse button twice in rapid succession.

**drag**

Pressing and holding down a mouse button while moving the mouse, which moves the pointer on the screen.

**drop**

Releasing an object, such as a file icon, that has been dragged to a new position. To drop the icon, release the mouse button.

**drop zone**

An area of the display that accepts a dropped file. Drop zones include the Trash, Printer, Mailer, and Personal Toolbox control and icons in the toolboxes that use file arguments. You can also drop objects on the Desktop for quick access. There are no drop zones in HP VUE Lite.

**execution host**

A host computer where an application invoked by the action runs. The execution host may be the same computer where the action resides, or it may be another computer on the network.

**File Annotator**

An HP VUE application for adding text or audio annotations to files. Annotations do not alter the contents of the file or directory being

annotated. (Audio annotations are available only on systems capable of recording and playing audio.)

**File Manager**

An HP VUE application for managing the files and directories on your system. File Manager is not available in HP VUE Lite.

**Front Panel**

A centrally-located window containing a variety of controls for accessing applications and utilities on your computer. The Front Panel also contains the workspace switch. The Front Panel occupies all workspaces.

**file server**

A host computer used to store data files used by applications.

**filetype**

An HP VUE mechanism used to associate particular data files with the appropriate applications and actions. Filetypes can determine the type of a file based on file naming conventions, such as a particular extension name, or by the contents of the file.

**font**

A complete set of characters (letters, digits, and special characters) of one size and one typeface. "Ten-point, Helvetica, bold" is an example of a font.

**General Toolbox**

A container for system-wide actions built into HP VUE or put there by your system administrator. To display the General Toolbox, choose General in the Toolboxes subpanel.

**home directory**

A personal directory where you keep files and additional subdirectories that belong to you. By default, File Manager and Terminal Emulator windows are set to your home directory when you first open them.

*/HomeDirectory/*

Symbolizes your home directory. For example, if your home directory is `/users/anna/`, then */HomeDirectory/*`bitmaps/smile.bm` represents `/users/anna/bitmaps/smile.bm`.

**home session**

A session you explicitly save without logging out.

**hyperlink**

A connection from one help topic to another related topic. A hyperlink can be a word, a phrase, or a graphic image. Any underlined text within a help window is a hyperlink.

Most hyperlinks cause a "jump" to a related help topic. However, hyperlinks can also be used to perform other behavior, such as executing commands or invoking actions.

**ITE**

An Internal Terminal Emulator. The ITE lets you use your bitmapped display as a terminal (via No Windows mode from the Login screen).

**keyboard binding**

Association of a key stroke with a particular behavior.

**list**

A control, also called selection list, that contains scrollable choices from which you can select.

**local host**

A host computer from which an action is invoked. This host *must* be running HP VUE.

**mapping**

A mapped action is an action that invokes another action, rather than containing its own EXEC-STRING. The file **/usr/vue/types/user-prefs.vf** contains the built-in mapped actions. For example, the built-in Mailer action used by the Front Panel is mapped to another action named Elm. You can change the mailer by remapping.

**maximize**

To enlarge a window to fill a workspace. The push button that maximizes a window is located in the upper right corner of a window frame.

**menu**

A list of commands that is displayed from a window or software application. The window menu enables you to control the size, shape, and position of that window. An application menu enables you to operate the application.

**menu bar**

> The bar directly beneath a window's title bar that contains the titles of the pull down menus for the software application.

**minimize**

> To turn a window into an icon. The push button that minimizes a window is located near the upper right corner of the window frame.

**mnemonic**

> A single, underlined character in a command. The mnemonic initiates that the command can be executed by typing that letter. Mnemonics are used most commonly in menu commands, however, other buttons may also have mnemonics.

**modifier key**

> A key that, when pressed and held, along with another key, changes the meaning of the second key. (Ctrl), (Alt), and (Shift) are examples.

**newline character**

> An unseen character that marks the end of a line of text in a document. It tells a printer or screen to break a line and start a new one.

**Network Toolbox**

> A container for actions available locally that are located on other host computers on your network. To display the Network Toolbox, choose the Network control in the Toolboxes subpanel.

**networked session**

> A session managed across multiple systems. Using a networked session allows you to see the same session, regardless of which system you use to log in. It also provides a single home directory across multiple systems.

**object**

> Any logical piece of data that has associated behavior. For example, in File Manager, files, directories, actions, and applications are all considered objects. Each type of object has specific associated actions. Typically, each object is represented as an icon.

**Personal Toolbox**

> A container for personal applications and utilities. To display the Personal Toolbox, choose the Tools control in the Front Panel.

**pixmap**

An image stored in a raster format. Usually the term *pixmap* implies that the image may have more than two colors. An image that is limited to two colors is usually called a *bitmap*.

**push button**

A control that immediately starts an action as soon as it is chosen. OK, Cancel, and Help are examples of push buttons commonly found in dialogs.

**radio button**

A control that—combined with other radio buttons—offers you a fixed set of choices that are mutually exclusive.

**resource**

A mechanism of the X Window System for specifying an attribute (appearance or behavior) of a window or application. Resources are usually named after the elements they control.

**scalable typefaces**

A mathematical outline for a typeface that can be used to create a bitmapped font for a particular size, slant, or weight.

**scroll bar**

A window component for controlling the contents of a scrollable area. Scroll bars are used when information is too large to display within a window.

**session server**

A system that provides networked sessions. The session files reside on the session server. These files are used whenever the user logs into a system on the network.

**slider**

A control that "slides" back and forth across a range of measurements or values. Style Manager's Color dialog controls that choose the red, green, and blue color values are examples of sliders.

**software application**

A computer program that provides you with tools to do your work. Style Manager, Text Editor, and File Manager are examples of software applications.

**Style Manager**

The HP VUE application with which you easily customize some visual elements and system device behaviors of your environment, including colors and fonts, and keyboard, mouse, window, and session startup behaviors.

**subpanel**

A component of the HP VUE Front Panel that provides additional controls. Subpanels usually contain groups of related controls. For instance, the Printers subpanel provides access to multiple printers.

**superuser**

A login that allows special permissions for modifying system files that most users do not have permission to modify. Superuser is also called "the root user" or simply "root" since the user ID for superuser is `root`. On most computer systems, only a few users have permission to become superuser.

**terminal emulator**

A window that emulates a particular terminal type for running non-window programs. The most common use of terminal emulator windows is interacting with your computer's operating system by typing commands. The two most common terminal emulators in HP VUE are `hpterm` and `xterm`.

**Text Editor**

The HP VUE application with which you create and edit documents.

**text field**

A rectangular area in a window where you type information. Text fields with keyboard focus have a blinking text insertion cursor.

**tile**

A rectangular area used to cover a surface with a pattern or visual texture. Workspace Manager supports tiling, enabling users with limited color availability to create new color tiles blended from existing colors.

**title bar**

A rectangular area of a window frame (at the top of the window) that identifies the contents of the window. Usually, a window's title is the name of the application—for example, Style Manager.

**toolbox**

A container for action icons. There are three toolboxes: Personal Toolbox, General Toolbox, and Network Toolbox.

**window**

A rectangular area on your display. Software applications typically have one "main" window from which "secondary" windows, called dialogs, can be opened. A window frame has controls with which you can manipulate the window.

**window frame**

The visible part of a window that surrounds a software application. A window frame can contain five controls: title bar, resize borders, minimize button, maximize button, and the window menu button. (Some windows may not display all of these controls.)

**window icon**

A small, graphic representation of a window that takes up little area in a workspace.

**window menu**

The menu displayed when you press mouse button 1 on the window menu button in the upper left corner of the window frame, or press (Alt) (Space).

**workspace**

A single screen of windows. The HP VUE Workspace Manager creates multiple workspaces. Each workspace is like a separate screen, plus you can have windows "occupy" any combination of workspaces. The Front Panel contains the workspace switch you use to move from workspace to workspace.

**Workspace Manager**

The program that controls the size, placement, and operation of windows within multiple workspaces. The HP VUE Workspace Manager includes the Front Panel, the window frames that surround each application, and window and workspace menus.

**Workspace Menu**

The menu displayed by pressing mouse button 3 on the workspace backdrop. This menu is sometimes called the "root" menu.

# Index

HEWLETT
PACKARD

Reorder No.
B1171-90061

**Manufacturing
Part No.
B1171-90061**