HyperLook

Copyright notice

This manual was written using the NeWS version of FrameMaker 2.1.1.

The following are trademark or registered trademarks of their respective companies or organizations:

HyperLook, HyperLook logo / The Turing Institute Limited
Turing Institute Logo / The Turing Institute Limited
Sun Microsystems, NeWS / Sun Mircosytems Incorporated
OpenWindows, SunOS, Solaris / Sun Microsystems Incorporated
Adobe, PostScript / Adobe Systems Incorporated.
OPENLOOK, UNIX / UNIX system Laboratories Incorporated
FrameMaker / Frame Technology Corporation
SmallTalk / Xerox Corporation
Helvetica, Times / Linotype Corporation
ITC Zapf Dingbats / International Typeface Corporation
Lucida / Bigelow and Holmes
Puma / Staubli Unimation A.G.

The Turing Institute Limited

George House

North Hanover Street

Glasgow G1 2AD, Scotland

Tel: +44 41 552 6400

Fax: +44 41 552 2985

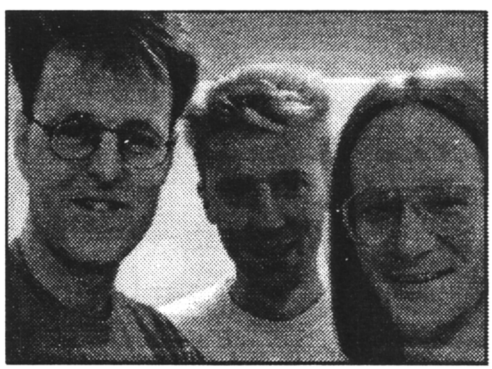
Acknowledgements

I would like to thank Don Hopkins for his hard work and late hours, Dug Scoular for his support, Tim Niblett for his expert advice, Rafael Bracho for his extremely useful feedback and Marleen for being so patient. I would also like to thank Jim Rudolf, Suhayya Abu-Hakima, and Danny Pearce for their work on earlier systems.

Many thanks to the hundreds of users who have provided the feedback which has been essential to the success of HyperLook.

Have fun,

Arthur van Hoff



Arthur, Dug and Don.

Table of contents

Chapter 1 Introduction

Chapter 2 Getting Started

Chapter 3 Using HyperLook

Chapter 4 Graphics Editing

Chapter 5 Editing Stacks

Chapter 6 Object Properties

Chapter 7 Scripting

Chapter 8 Client Programming

Appendix D Library functions

Is Index

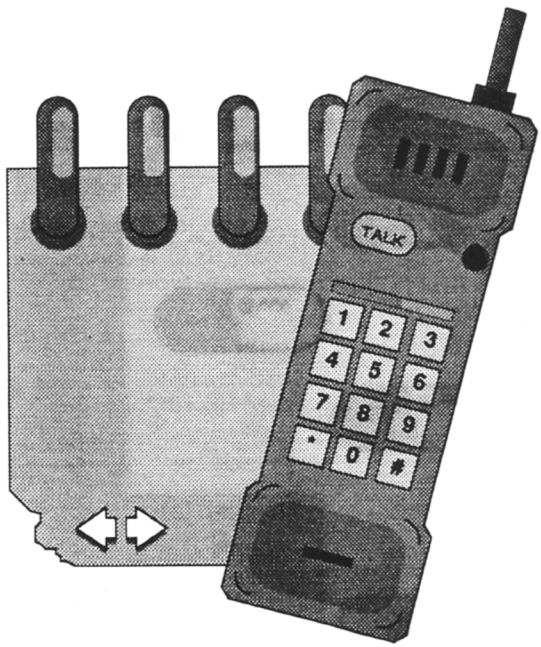
1 Introduction

Open windows with HyperLook

Welcome to the exciting world of HyperLook™ user interface design[1]. HyperLook is a powerful, easy to use development tool, and it's fun, too!

Figure 1

HyperLook is fun to use!



1. HyperLook is a trademark of The Turing Institute Limited.

Introducing HyperLook

This chapter gives you a whistle-stop tour of HyperLook. From it you'll get an idea of the things you can do with HyperLook and you'll be introduced to some of the terminology of the HyperLook world.

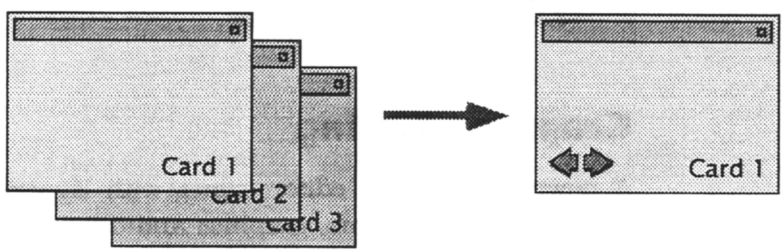
Don't worry if you don't understand every new term, they're all explained later. Now read on and find out what HyperLook can do...

Stacks and Cards

HyperLook uses the names **stacks** and **cards** instead of windows and window layouts. This widely used terminology was chosen to emphasize the fact that HyperLook stacks (windows) can have more than one card (layout).

Figure 2

Stacks can have multiple cards. You can get to them using the arrow buttons.



One card per window? Or all cards in one stack!

Each window layout is called a card, and may contain graphics and user interface components such as buttons, text and sliders. You can flip between cards like turning pages in a book.

Desktop Management

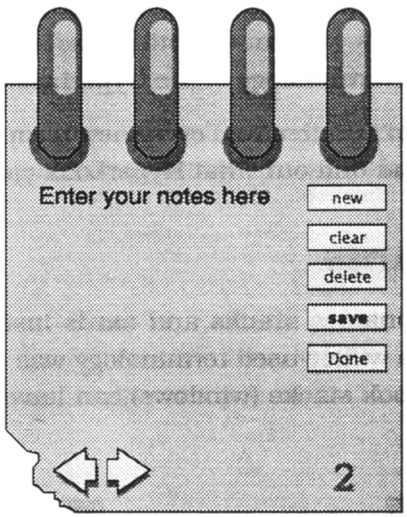
HyperLook is an easy to use environment. Any user of OpenWindows will find that it is easy to learn how to use HyperLook.

You are free to configure your working environment, and you can modify and extend the user interface.

HyperLook provides a comprehensive set of desktop tools including a clipboard, color pallet, notepad and many others.

Figure 3

The HyperLook Notepad stack.

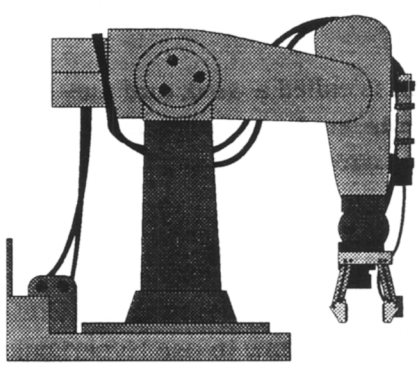


Graphics Editing

A powerful graphics editor comes with HyperLook. It lets you create sophisticated diagrams, graphics and other artwork, and it's simple to use.

Figure 4

An example of a drawing created with the HyperLook graphics editor.



Graphics created with HyperLook can be incorporated into other environments and used with other tools. For example, most of the artwork and diagrams in this manual were created using the PostScript® graphics editor and were then imported into FrameMaker™[1]. PostScript graphics can be imported into many desktop publishing tools.

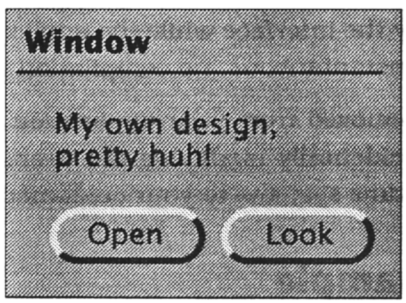
Rapid Application Development

HyperLook simplifies the task of building user interfaces. You can prototype user interfaces quickly by direct manipulation, without restriction of the creative process.

HyperLook is not restricted to a single look and feel. You can create a new look and feel for your application, or adapt an existing look and feel.

Figure 5

HyperLook can have any look and feel.



Interactive Design

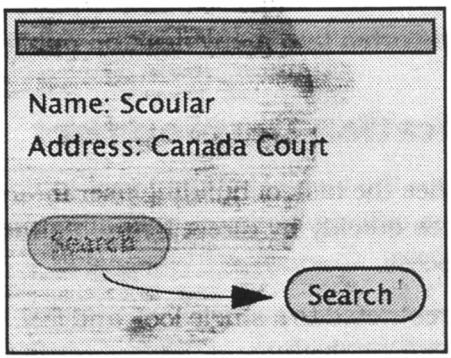
The interface designer can do much of the interface development by direct manipulation: by drawing, dragging and sizing objects with the mouse, without writing a line of code.

1. PostScript is a registered trademark of Adobe Systems Incorporated, FrameMaker is a trademark of Frame Technology Corporation.

One of the most time consuming aspects of creating interactive applications is the design of the interface. Even non-programmers can design applications using direct manipulation in HyperLook.

Figure 6

Design your interface by direct manipulation. Pick up a button, move it, and go!



You can even modify the interface while the application is alive and running! Changes are instantaneous, no compilation is needed!

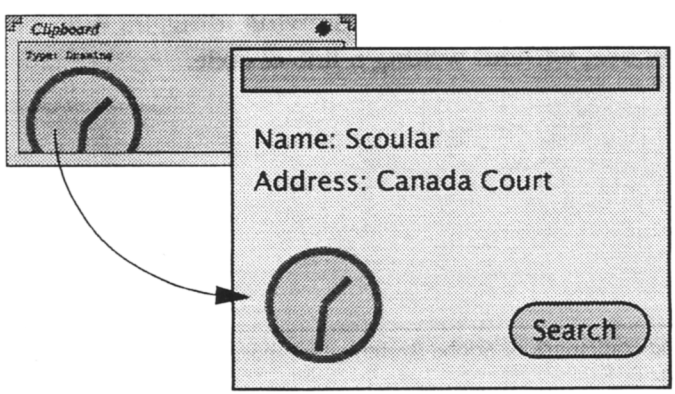
The editor can be removed from your application to stop users of your application from accidentally modifying the user interface. In this way you can deliver runtime systems to your customers.

Design by Example

Not only can you design your own user interface, but you can also use components designed by others. By copying and pasting components, you can insert them into your application, complete with their functionality.

Figure 7

Pasting a clock into your application.



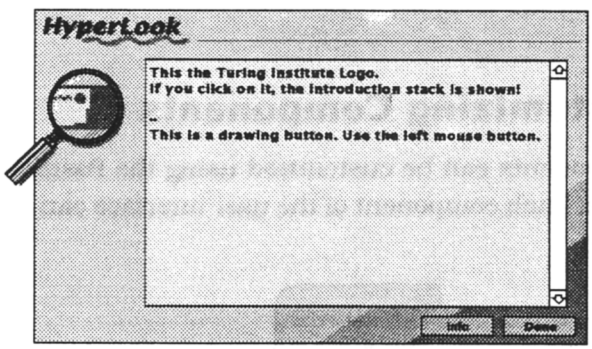
HyperLook provides example interfaces and components which you can plug right into your application.

Providing Help

Not only does HyperLook provide an on-line help system which helps you to navigate through the system, it also lets you define your own help for the users of your application. No programming is required!.

Figure 8

The HyperLook Help stack.



Help can be provided as text but you can also design your own help stack. It can use graphics and all the other functionality provided by the HyperLook system.

Extensibility

A comprehensive set of user interface components like buttons, scroll-bars and pull-down menus is provided. This set can be extended and tailored to the needs of any application.

HyperLook is not limited to the components it provides. You can modify existing components or add completely new ones. You can draw new components, customize existing ones, or create entirely new objects with their own interactive behaviour.

It is possible to design and implement a corporate style, or to design a new set of components for your environment. For example, you could make a set of components for use in a process control environment. Once tested they can be used to build industrial process control applications.

Figure 9

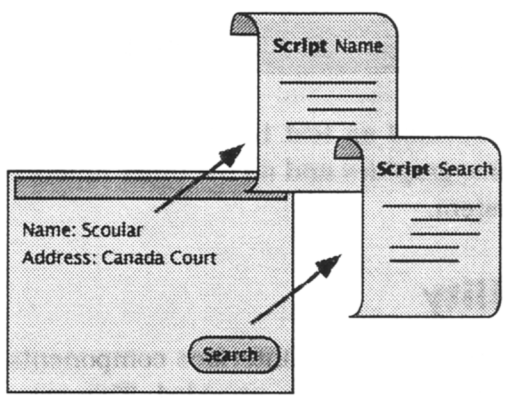
Designing a knife switch by drawing the on and off positions.

Customizing Components

Components can be customized using the PostScript programming language. Each component of the user interface can have a *Script* associated with it.

Figure 10

Every component can have a Script.



Scripting (programming user interface components) lets you to change the component's look and feel.

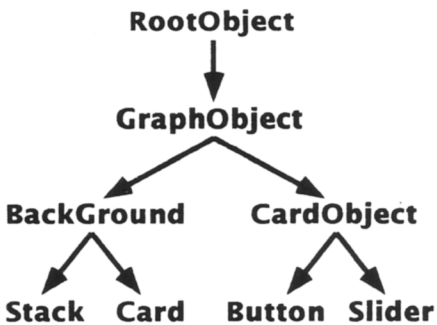
There is a C-like scripting language available for HyperLook, called PdB. PdB is the ideal scripting language if you are already familiar with C. You can write new component classes in PdB too!

Object Oriented

HyperLook is completely object oriented. All components are implemented as classes which are defined using a classing mechanism similar to that of SmallTalk™[1].

Figure 11

Functionality is Inherited using classes.



An easy to understand message passing mechanism is used for communicating between objects and communicating with client programs.

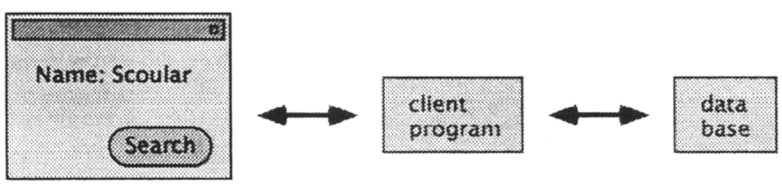
It is possible to add your own classes to the system. The classing mechanism provides multiple inheritance, persistent variables, dynamic loading, and other features useful for the creation of dynamic object oriented user interfaces.

Interfacing to Other Languages

HyperLook allows any UNIX® process to access it through an easy to use C programming interface[2]. Interfaces to other languages such as C++, Prolog, Lisp and Perl are available.

Figure 12

Linking a database client to HyperLook.



1. SmallTalk is a trademark of Xerox Corporation.
2. UNIX is a registered trademark of Unix system Laboratories Incorporated.

Any program that can link in a small C library is able to use the HyperLook interface. The C programming interface lets you create interfaces to complex applications such as database managers, expert systems, games, etc.

Platforms

HyperLook uses the OpenWindows™ system and runs on SunOS™ and Solaris® platforms[1]. It integrates into the existing Sun desktop. It makes use of the Sun desktop tools such as the file manager and the mail tool.

HyperLook provides complete access to the powerful PostScript capabilities of the OpenWindows environment.

1. OpenWindows, SunOS and Solaris are trademarks or registered trademarks of Sun Mircosystems Incorporated.

Background

HyperLook

The development of HyperLook began in October 1987 at The Turing Institute, and the initial system was completed in April 1988.

The programming and detailed design of HyperLook was done by Arthur van Hoff of The Turing Institute.

After 5 years of development and testing, worldwide distribution and 300 sites using the system, HyperLook was released as a commercial product in June 1992.

The Turing Institute

The Turing Institute is an independent company specializing in the research, development and commercial exploitation of the latest advances in computer software. The company was formed in 1983 in honor of Alan Turing, whose pioneering work helped provide a formal basis for present day computing and artificial intelligence.

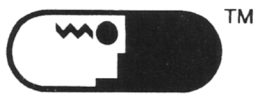
Figure 13

Alan M. Turing, 1912-1954

The Institute's activities are sharply focused on solutions which allow computers to interact with both humans and their environments. Such a strategy has led to a number of important developments and products in areas as diverse as user-interfaces, document image processing, decision support, fault diagnosis, machine learning, advanced robotics, computer vision and intelligent documentation.

Should you require any further information about the Turing Institute or about any of its technologies or products, please contact:

The Turing Institute Limited[1]



George House

North Hanover Street

Glasgow G1 2AD, Scotland

Tel: +44 41 552 6400

Fax: +44 41 552 2985

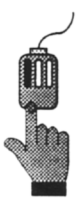
1. The Turing Institute logo is a trademark of The Turing Institute Limited.

Reading this Manual

This manual provides a complete overview of the HyperLook system.

- Using HyperLook
  - **Getting Started** explains how to start HyperLook and how to set up your environment.
  - **Using HyperLook** explains how to use HyperLook stacks and components. It explains how to use the tools which are provided by HyperLook.
  - **Graphics Editing** explains how to use the HyperLook graphics editor.
- Editing HyperLook stacks
  - **Editing Stacks** explains how you create your own stacks and cards and how you edit components.
  - **Object Properties** explains the properties of the HyperLook objects.
- Programming HyperLook
  - **Scripting** explains how to write scripts for objects.
  - **Client Programming** explains how to communicate with HyperLook from UNIX programs.

The Mouse



Where this manual says *click the mouse* or *press the mouse* you should use the **left** mouse button. HyperLook uses the left mouse button most of the time, that is why the mouse button specification is usually omitted.

If you are meant to use the **middle** or **right** mouse button, it will be specified explicitly.

2 Getting Started

This chapter explains how to start HyperLook and how to exit it. Once you know how to start HyperLook, you can move on to the next chapter, which explains how to use the HyperLook user interface.

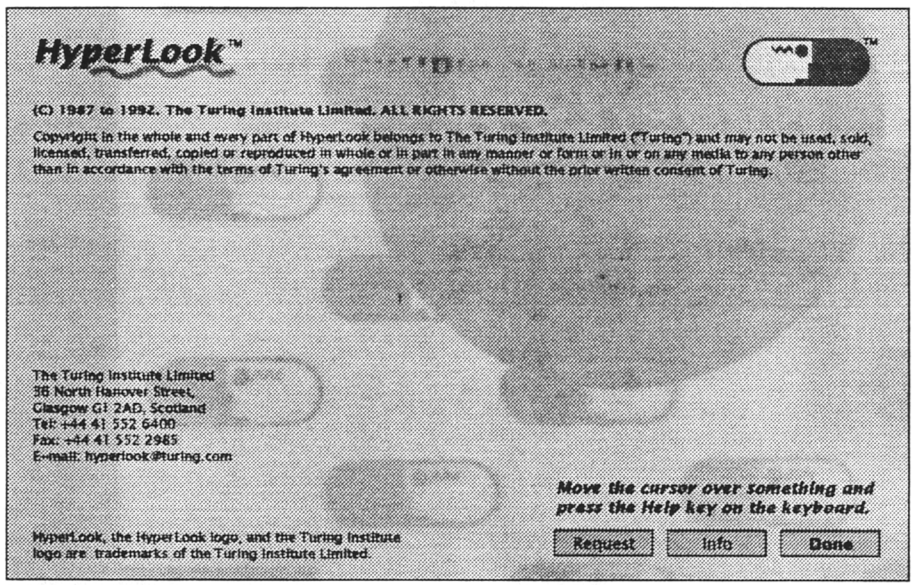


Figure 14

The introduction stack. It is shown when you start HyperLook.

Starting HyperLook

Before using HyperLook, it must be installed on your computer. Refer to the *Installation Guide* which you received with HyperLook for installation instructions[1].

Once HyperLook is installed, you must make sure that you have access to the directory where it lives. This directory is called the HyperLook home directory.

HyperLook is usually installed in the directory **/usr/src/hyperlook**. If this directory does not exist on your system, ask your system administrator if who will tell you where HyperLook is installed.

You start HyperLook with the **hyperlook** command. Assuming that HyperLook is installed in **/usr/src/hyperlook**, you can start HyperLook by typing the following command in the UNIX shell:

```
/usr/src/hyperlook/bin/hyperlook
```

Below is an example of the HyperLook startup procedure. The things that you type are in bold, what the computer types is in italics.

```
Where is OpenWindows installed?

Enter OPENWINHOME: /usr/src/openwin3

Where is HyperLook installed?

Enter HLHOME [/usr/src/hyperlook]:<Return>

Starting OpenWindows version 3.0
Using OPENWINOPTIONS = -cubesize large
Using OPENWINHOME = /usr/src/openwin3
Using HLHOME = /usr/src/hyperlook
```

If you have not started OpenWindows, HyperLook will ask you where OpenWindows is installed, so it can start the window system for you. If you don't know where it's installed, refer to your system administrator.

1. You can find more information on how to install HyperLook in the INSTALL file in the HyperLook home directory.

If all goes well, the Introduction Stack will appear in the center of your screen after 30 seconds or so. Figure 14 on page 2-2 shows a picture of this stack.

If you don't succeed in starting HyperLook you should read "Trouble Shooting" on page 2-8.

Read "Using HyperLook Every Day" on page 2-9 if you want to install HyperLook more permanently.

Now that you have loaded the Introduction stack, click the mouse button (the left one) on the **Done** button, this will remove the Introduction stack, and show the **system** stack and the **SystemStatus** stack.

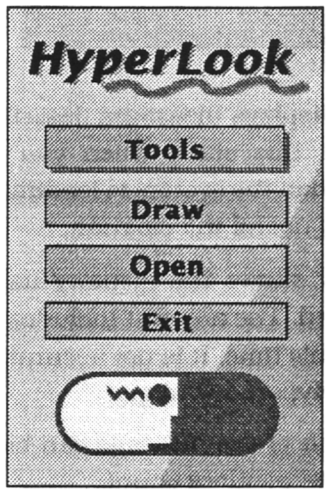
Press the **Request** button if you want to request more information by electronic mail. Press **Info** for more information on HyperLook.

HyperLook Basics

The system stack contains buttons which give you access to functions that you will use frequently.

Figure 15

The HyperLook system stack.



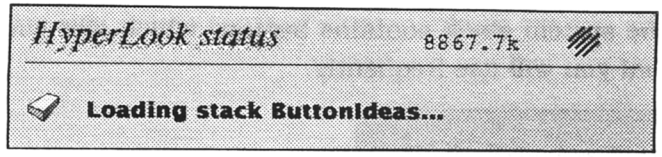
The buttons on the system stack provide the following functions:

- **Tools** Access frequently used tools (see Chapter 3, "Using HyperLook").
- **Draw** Open a HyperLook graphics editor (see Chapter 4, "Graphics Editing").
- **Open** Open or create a stack (see "Opening Stacks" on page 3-8).
- **Exit** Exit the HyperLook system (see "Exiting HyperLook" on page 2-6).

The SystemStatus Stack

Figure 16

The SystemStatus stack.



The **SystemStatus** stack displays messages describing what HyperLook is doing. Keep an eye on this stack when you are learning to use HyperLook. It shows you when the system is loading stacks, when a loading error occurs, and other useful information.

The top right hand corner shows the memory used by OpenWindows since HyperLook was started. The amount includes the memory used by other applications during this time. It is not accurate in determining how much memory is used by HyperLook alone!

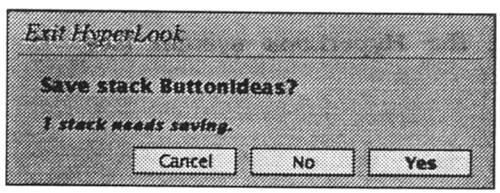
The eraser button to the left of the message can be clicked to erase the message. It also updates the memory count.

Exiting HyperLook

After using HyperLook you can exit by pressing the **Exit** button in the system stack. This will show the **ExitHyperLook** stack. If you have been editing stacks, you will be asked whether you want to save the stacks that you have changed before exiting.

Figure 17

Do you want to save changes before exiting?

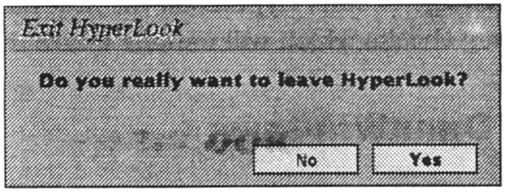


You can save a stack by pressing the **Yes** button, discard changes to the stack by pressing the **No** button, or decide not exit at all by pressing the **Cancel** button.

Once you are sure you want to exit HyperLook, press the **Yes** button. You can still decide not to exit HyperLook by pressing the **No** button.

Figure 18

Really exit HyperLook?



Trouble Shooting

If you don't manage to start HyperLook the first time: *don't panic*. There are several easy checks which will usually locate the problem.

Starting OpenWindows

1. Make sure you are using the correct version of OpenWindows. Refer to the installation instructions to find out which versions of OpenWindows are supported.
2. Conflicting initialization files in your home directory (some programs create these files without asking you), in particular files called **`.user.ps`**, **`.startup.ps`** and **`.init.ps`**[1], may set your environment up incorrectly. Try removing these files and then start HyperLook as described in "Running HyperLook" on page 2-8.

If you are using a version of OpenWindows that is supported by HyperLook and the above does not work then you should try to start OpenWindows from scratch. Refer to the OpenWindows installation guide if you have any problems doing so.

You may want to remove **`.xinitrc`** from your home directory if you are running a new version of OpenWindows for the first time. OpenWindows requires certain initializations which are not standard for X window systems.

Running HyperLook

Once OpenWindows has started, open a console window and try running HyperLook from that window as described in "Starting HyperLook" on page 2-3. Pay attention to error messages appearing in the console window.

If the HyperLook introduction stack does not appear, ask your system administrator to make sure that HyperLook is installed correctly.

1. To list files all files in your home directory use: **ls -a ~**

Using HyperLook Every Day

If you are using HyperLook a lot (and why shouldn't you?) you may want to take a few more steps so that it's really easy to start and stop HyperLook at any time.

Using the File Manager

The Sun desktop environment provides a file manager tool which provides an iconic representation of your file system. The Sun documentation tells you how to use the file manager. You can use the file manager to start HyperLook and to open stacks and drawings.

Before you can use the file manager with HyperLook you need to install the HyperLook icons. To install them execute the following command:

```
/usr/src/hyperlook/bin/install_hyperlook
```

Once you have done this you must exit the file manager and restart it. It will now recognize HyperLook files.

Figure 19

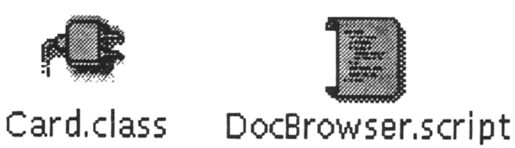
The HyperLook icon and a stack icon.

You can start HyperLook by double clicking the HyperLook icon, or you can open a stack by double clicking a stack icon.

The class icon represents a HyperLook user interface object, it contains code for the object. It cannot be opened by itself. A script contains code for a particular user interface object. It can be opened for editing.

Figure 20

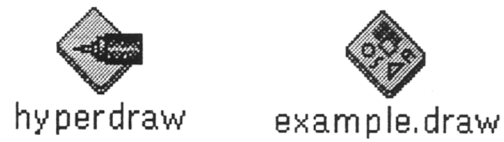
The HyperLook object class icon and the script icon.



Double clicking the hyperdraw icon starts the graphics editor. A drawing can be opened by double clicking a drawing icon (see Chapter 4, "Graphics Editing").

Figure 21.

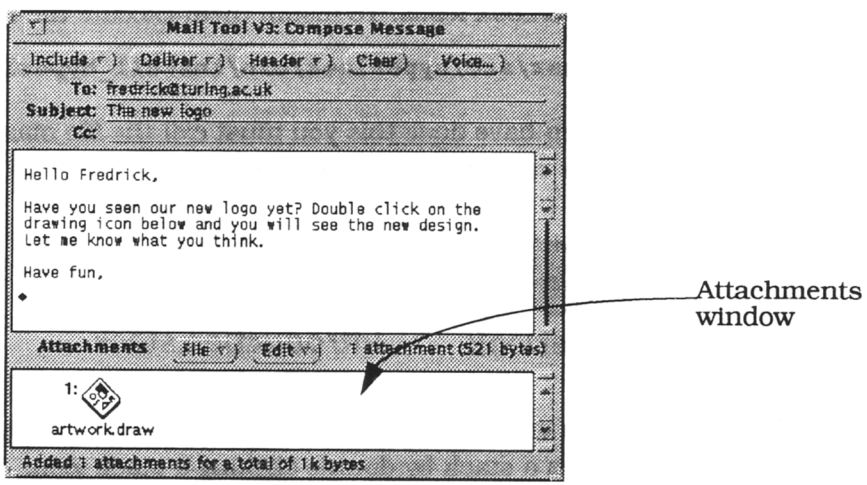
The HyperDraw icon and a drawing icon.



If you use the Sun desktop mail tool, you can drag stacks and drawings into the mail tools attachments window when composing a message to another user. This lets you to easily mail stacks and drawings to other users.

Figure 22

Using the mail tool to mail a drawing to another user.



Setting Up Your Environment

You may want to install HyperLook into your environment permanently. This involves changing your **`.login`** file and setting some environment variables.

You don't need to set your environment up to run HyperLook. HyperLook can set your environment up for you.

If you want to set up your environment, then you must edit your **.login** file to include the following lines:

```
# This is where OpenWindows lives
setenv OPENWINHOME /usr/openwin

# This is where HyperLook lives
setenv HLHOME /usr/src/hyperlook

# Make HyperLook executables accessable
set path=($HLHOME/bin $path)
```

Note that the above is only an example and that the directories may not be the same on your system.

Once you have changed your **.login** file, log out and log in again. You can then start HyperLook either from your login shell or from a window by typing **hyperlook**.

When starting HyperLook from your login shell it will start OpenWindows for you. You can specify parameters to be passed to OpenWindows. This is necessary if you want to run OpenWindows on multiple screens, or if you want to run OpenWindows with the **-nosunview** or **-includedemos** option.

You can specify options in the **OPENWINOPTIONS** variable. Add the following to your **.login** file:

```
setenv OPENWINOPTIONS "-nosunview -includedemos"
```

Refer to the OpenWindows documentation for other options that you can specify when starting OpenWindows.

HyperLook on Color Screens

To display colors correctly, HyperLook needs to run in an environment with a large color cube size. The color cube size is selected by the OpenWindows **-cubesize** large option. It is set as follows:

```
setenv OPENWINOPTIONS "-cubesize large"
```

HyperLook works fine if this option is not set but you may find that some of the colors are not correct. HyperLook supports both dynamic and static visual modes.

HyperLook and Multiple Screens

HyperLook uses the **DISPLAY** environment variable to connect to the OpenWindows server. This variable is automatically set when you startup OpenWindows.

After connecting to the server HyperLook determines whether the server has a color screen. If it has, then this screen will be used.

See the OpenWindows documentation how the **DISPLAY** environment variable can be used to run HyperLook remotely.

3

Using HyperLook

This chapter describes how to use the HyperLook user interface. HyperLook has its own user interface style, but it is just like any other application running on the OpenWindows desktop. That means that you can cut and paste from HyperLook to other desktop applications. The mouse is used is the same way, too!

This chapter explains how HyperLook is used in *browse* mode. This is the mode in which you normally use the system. The other mode is called *edit* mode. It is explained in Chapter 5, "Editing Stacks".

The best way to read this chapter is to get HyperLook started (see Chapter 2, "Getting Started") and try things out as you go through.

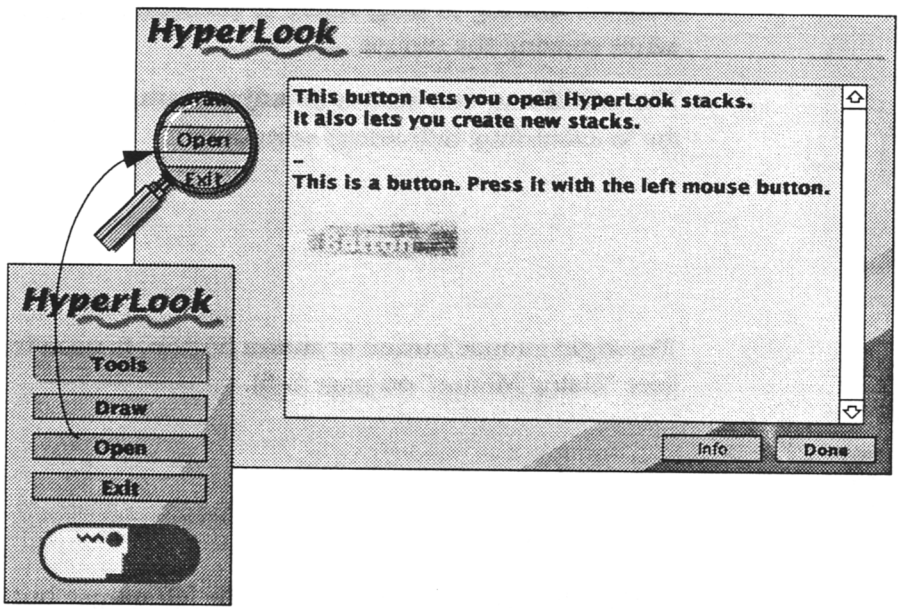
Using Help

HyperLook has a built in help system which lets you get help about the user interface. To get help on something, move the mouse over the object and press the **Help** key on the keyboard[1].

Most components of the HyperLook user interface provide a short explanation of their functionality this way.

Figure 23

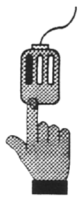
Getting help on the **Open** button of the system stack.



Remember to use Help if you get stuck. It provides useful information on most of the HyperLook components.

1. On most keyboards the **Help** key is on the bottom left.

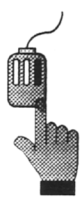
Use of the Mouse



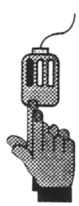
The **left** mouse button or **select** button is used in most cases. It is used to activate buttons, sliders, to select from pull down menus, to make selections etc.

Unless otherwise specified, a *"mouse click"* or *"press the mouse"* refers to the left mouse button. A *click* refers to pressing and releasing the button without moving. A *drag* refers to pressing and holding down the button while moving the mouse.

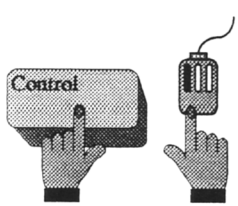
The **middle** mouse button or **adjust** button, is not used very often. It is for to extending (adjusting) selections.



The **right** mouse button or **menu** button, is used to show pop-up menus (see "Using Menus" on page 3-5).



To **double click** the mouse click the left mouse button twice in the same position.



You sometimes need to use the mouse while holding down a key on the keyboard. First hold down the key and then, while holding the key down, use the mouse.

Using Menus

Pop Up Menus

Holding down the right mouse button in any stack will pop up a menu. To select a menu item move the mouse over the desired item, while holding down the right mouse button.

Figure 24

The stack menu.

As you move over items, they highlight to indicate which one will be executed when the mouse button is released. Releasing the mouse over a selected item executes the item and the menu disappears.

If you don't want to select *any* item, move the mouse off the left edge of the menu until no item is selected and release the mouse button.

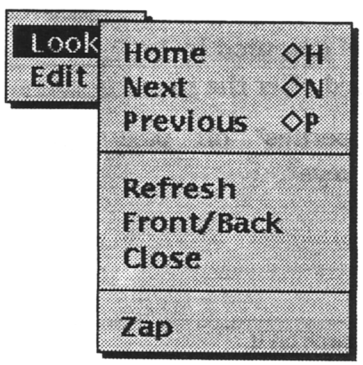
The HyperLook stack menu is explained in "The Stack Menu" on page 3-36.

Pull Right Menus

Some menu items have a ">>" sign on the right as in Figure 24. This means that they have a pull right sub menu. To show it, move the mouse over the ">>" sign.

Figure 25

Selecting a sub menu.



You can hide a pull right menu by moving the mouse over its left edge until it disappears.

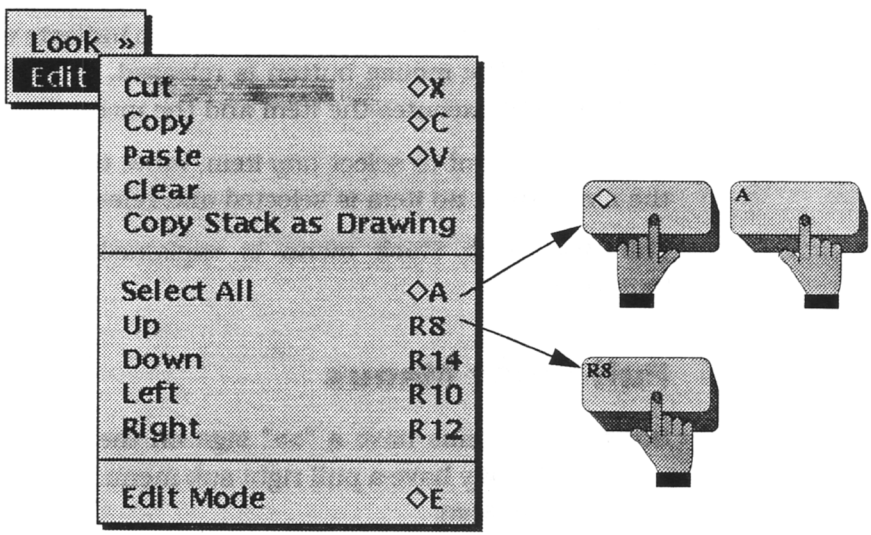
Menu Key Accelerators

Frequently used menu items have menu key accelerators, sometimes called keyboard accelerators. This lets you execute a menu item from the keyboard without showing the menu.

These menu items are displayed with a diamond and a letter. They are executed by holding down the meta key[1] and, while it is held down, typing the appropriate letter.

Figure 26

Menu key accelerators.



Some menu items can be executed by typing a function key[2]. The function key name is displayed after the menu item.

See "The System Properties" on page 3-31 on the use of the OpenWindows function keys[3].

1. The meta key on the Sun keyboard has a diamond shape on it.
2. Function keys are labelled F1 through F12 and R1 through R15.
3. The OpenWindows function keys are labelled Stop, Again, Props, etc.

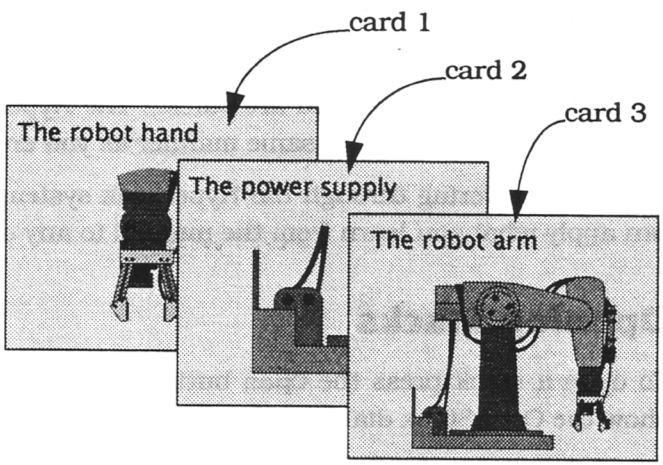
Using Stacks

A HyperLook window is called a *stack* (see also "Stacks and Cards" on page 1-3). The reason for this is that each stack may contain several cards just like a stack of cards. A card represents a layout of the stack.

Let's look at an example. A help system may provide help on several topics.This could be solved by designing a stack which contains a card for each topic. When help on a topic is needed the appropriate card of the help stack is shown.

Figure 27

The 3 cards of a Robot information system.



Each card can hold objects such as buttons but it can also hold graphics. You can change cards like turning pages of a book.

The Home Card

The first card of a stack is called the *home* card. When a stack is displayed for the first time, the home card is usually visible. The home card is often used as an index to the other cards in the stack.

Pressing the **Home** key on the keyboard takes you to the home card. The **End** key takes you to the last card of the stack.

Moving Between Cards of a Stack

Each stack can have more than one card. You can move to the next card of a stack by pressing the **PgDn** key on the keyboard. Pressing the **PgUp** key takes you to the previous card.

Going to the previous card from the home card takes you to the last card of the stack. The same is true in reverse, the next card after the last card in the stack is the home card.

Every Window is a Stack

While you learn to use HyperLook you will notice that all the HyperLook windows which are provided are stacks! The HyperLook system stacks were created in exactly the same manner as you create stacks.

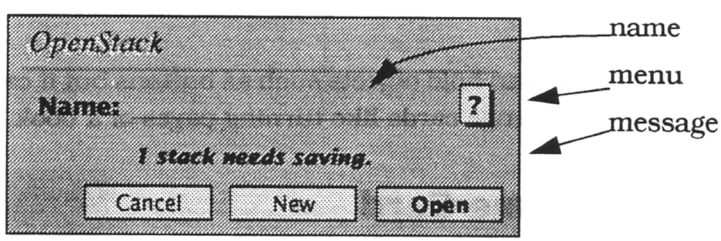
While maneuvering through the HyperLook system remember that you can apply what you learn from the manual to any of the system stacks.

Opening Stacks

To open a stack press the **Open** button in the system stack. This will show the OpenStack dialog.

Figure 28

The OpenStack dialog.



To open a stack, move the mouse over the OpenStack dialog and type the name of the stack and hit return. When the stack is loaded the dialog will disappear and the stack is shown. Try opening a stack called **Notepad**.

If the stack is not found, the message in the OpenStack dialog will say so. Refer to "The Resource Manager" on page 3-23 on how to open stacks from other Unix directories.