ORPHEUS

An Integrated Development Suite

For Hypertext Authoring

- User's Guide -

Orpheus is a suite of programs dedicated to the creation and distribution of hypertext. The main program, OH.EXE, provides an integrated platform for hypertext authoring. The runtime reader, OHREAD.EXE, is distributed to end-users with works in finished forms. Among the other components, OH1to2.EXE is a conversion program for use by authors wishing to upgrade from Orpheus 1.

Orpheus makes you a master of hypertext; it gives you the freedom to explore the multiple dimensions of your subject while bringing them within easy access of your readers; it gives you the flexibility to experiment, to revise and restructure at will; it gives you the power to compose spontaneously in an electronic galaxy, then to release the worlds you have created so that others may experience them.

Orpheus is SHAREWARE. This means that you are welcome to evaluate the system, but you may not distribute anything you might create with it without purchasing a registered copy. For pricing and a handy order form, please see the file REGISTER.TXT.

ACKNOWLEDGEMENTS:

CONVENTIONS FOLLOWED IN THIS MANUAL:

Keystrokes are indicated by <> brackets; for example, <F1> means press and release the F1 key, <H> means press and release the "h" key. Combination keystrokes are hyphenated; for example, <Alt-F> means press and hold down the Alt key, then press the "f" key, then release both keys.

The term "Orpheus" is used both in reference to the system as a whole and to its main component, OH.EXE (the authoring program). Other components are referred to with names related to their function, such as: the Reader (OHREAD.EXE), Help (OHHELP.EXE), the Simulator (OHSIM.EXE), the Indexer (OHINDEX.EXE), and the Converter (OH1to2.EXE).

TRADEMARKS

Trademarks are the property of their respective holders.

T A B L E   O F   C O N T E N T S

....

CHAPTER 1 - GETTING STARTED

••

Topics in this chapter:

Quickstart
How To Use This Manual
System Requirements
Installation
Upgrading from Orpheus 1
Getting Help

Quickstart

If you're the intrepid explorer type who never reads manuals, go ahead and fire up the authoring program (OH.EXE). Everything you need to know is in the menus and online Help. Browse them both, experiment, have fun. For an explanation of any menu item, click on it with the right mouse button or place the selection bar on it and press <F1>. To make sure you get off on the right foot, I suggest you read Chapter 3, the tutorial on making hypertext with Orpheus.

How To Use This Manual

This manual is a supplement to OH.EXE's online Help. Use it to find out what Orpheus is all about, to get set up, to take your first steps on the path of hypertext authoring. Bear in mind however that all of the detailed information on Orpheus IS in online Help, which you can open   at any time in OH.EXE by pressing <F1>. You can also run Orpheus Help as a standalone program: give the command OHHELP at the DOS prompt.

System Requirements

The authoring program (OH.EXE) requires 512K of RAM, DOS 3.2 or higher, and a hard disk. There are no particular video requirements unless you wish to display graphics. For best performance with large projects, XMS memory is recommended but not required.

The Orpheus Reader (OHREAD.EXE) requires 340K or more of RAM, depending on the work being displayed, and DOS 3.2 or higher; a hard disk is NOT required. Video requirements depend on the graphical demands, if any, of the work being displayed.

All Orpheus programs should run on any IBM-compatible computer, from the late-model XT through AT, 386, 486, and Pentium systems. Orpheus will not run on the PCjr or early XT's. All Orpheus programs run happily under Windows or OS/2. A Microsoft-compatible mouse is required for drawing hotspots in graphics.

Installation

Orpheus should be installed in its own subdirectory on your hard disk, such as "C:\OH2". You will need at least 1.4 megabytes of free disk space. If installing from disks received from Hyperion Softword, follow the instructions that came with the disks. If installing from compressed files downloaded from the Internet or a BBS, create an Orpheus directory and uncompress them into that directory.

Orpheus 1 and Interim Version users: If you have used Orpheus 1 or the Interim Version, be sure to install Orpheus 2 in a completely different directory. The only contact between Orpheus 2 and earlier versions should be via the Converter, OH1TO2.EXE (see below, "Upgrading from Orpheus 1").

Orpheus requires that the DOS "Files" variable be set to at least 15. If your system runs Windows or OS/2 this is guaranteed to be the case. If not, look in your CONFIG.SYS file (in the root directory of your hard disk) for the line beginning "FILES=". Make sure that the number given is at least 15; if you change it, reboot the computer to make the new setting come into effect.

Once Orpheus is installed you can work with it in other drives or directories, if you wish, provided you either specify a path on the command line or include the Orpheus directory in the PATH environment string given in your AUTOEXEC.BAT file. If you don't have an AUTOEXEC.BAT   file, or don't know how to use the PATH command, consult your DOS manual.

Upgrading from Orpheus 1

If you have projects developed with Orpheus 1.nn or the Interim Version, you can use the Converter (OH1TO2.EXE) to upgrade them to work with Orpheus 2. See OH1TO2.TXT for details.

Getting Help

If you get stuck or need something explained, the FIRST thing to do is to explore online Help. In virtually all situations, pressing <F1> calls up context-sensitive Help with an explanation of what you need to do or know. When menus are open, context-sensitive Help on any item can be had by clicking on it with the right mouse button; or move the selection bar to that item and press <F1>. You can also run Orpheus Help as a separate program, by giving the command OHHELP at the DOS prompt.

If you're still stuck (and have already absorbed this manual), you are welcome to contact me (Rod Willmot) at Hyperion Softword, by one of the following methods:

> 1. Email (willmot@interlinx.qc.ca) -- this is the best way to get help, report
> problems, or discuss improvements you'd like to see.
> 2. Call or fax 1-819-566-6296, preferably between 11:00 a.m. and 11:30 p.m.,
> New York time (generally 7 days a week). I can help best if your computer is up
> and running with Orpheus loaded.
> 3. Write to me at Hyperion Softword, 535 Irene-Couture, Sherbrooke, QC,
> Canada J1L 1Y8. This is the worst way to get help because the turnaround time
> for me to receive and reply to your letter is simply too long.

CHAPTER 2 - THE ORPHEUS ENVIRONMENT

...

Topics in this chapter:

Orpheus (OH.EXE) is an Integrated Development Environment for creating hypertext documents. Everything you need to compose, revise, and enhance such documents is right at your fingertips. In this chapter we'll take a look at that environment and how you can expect to use it.

The Editor

When you start up OH.EXE the first time, you will see an empty screen with a statusbar at the top and the cursor in the upper left corner. What you are looking at is a full-screen window in the hypertext editor that is at the heart of Orpheus. You can open 9 such windows. As shown by the "1" on the statusbar (beside "Esc:Menu"), you start off in Edit window 1.

In Orpheus 2, uncompiled hypertext cards are simply text files, combining the words you want your readers to see with codes for hypertext links. Since the codes themselves are text, you could use just about any editor to compose an Orpheus project. The advantage of OH.EXE's editor is that, except when you actually want to see the codes, it interprets and hides them, letting you see your work the way your readers will see it, and navigate through it as hypertext even while you are still writing it.

As you work on a project in Orpheus, you will typically have from half to all 9 possible Edit windows open, some containing various parts of your project, others perhaps containing external text files that you are using as resources. (Besides composing hypertext, the Edit windows can be used to view or edit text files not belonging to a project.) When you exit OH.EXE or switch to a different project, Orpheus saves a record of the window contents for the project you were working on. The next time you open that project, Orpheus will restore the windows to what they contained before you quit.

For more about the Edit windows in Orpheus, check out the Window Menu and read online Help.

Statusbar

   The Statusbar displays a variety of messages and symbols that keep you informed of where you are and what you are doing; some of them can be used with the mouse to switch windows, toggle edit modes, or navigate hypertext. The illustration below represents all of the items that might   appear. (See online Help for more thorough discussions of each.)

```
Esc:Menu |*1 | Push Marking IniExiPrvNxtRptMor Wrap | _ |[   1]| "FILENAME.EXT"
1      2   4   5      6 7 8 9 10 11 12   13 14  16 17
         ^3                              ^15
```

(1) Press <Esc> to open the menu system, or click on "Esc:Menu". To open a specific menu in one step, press <Alt> with the first letter of the name of the menu, e.g. <Alt-F> for the File Menu.

(2) A hilited asterisk indicates that the file in the current window has been changed since it was last saved to disk.

(3) Item 3 is the number of the current Edit window being displayed. If you click on this number Orpheus switches to the next non-hidden window if any. For other window-related commands, open the Window Menu.

(4) The current editing mode is indicated here: "Push" for Pushright means that typing a character pushes any existing text at the cursor to the right; the opposite is "Over" for Overwrite. Press <Ins> to toggle from one mode to the other, or use the Edit Menu.

(5) Next comes a message area, which is normally blank. The illustration displays the "Marking" message, which would appear if you were marking a block of text. If you click on this area (whether blank or not), the Mouse Blocks dialog opens to facilitate block operations with the mouse.

(6-11) The central area of the Statusbar is reserved for flags that indicate whether the current card has a certain type of built-in link. These are: "Ini" for an Init link, "Exi" for an Exit link, "Prv" for a Prev link, "Nxt" for a Next link, "Rpt" for a Report link, and "Mor" for a More link. To navigate such a link you can click on the flag or go use the Link Menu. Note that the Reader interface to such links is completely different; for example, Init and Exit links are events that   the Reader performs automatically when the user enters or leaves a card with such links.

(12) Another edit-mode toggle concerns wordwrap and autoreformat. "Wrap" means that wordwrap is active; "Rfmt" means that autoreformat is active. You can toggle these states either by clicking on the flag, or through the Edit Menu. In ascii mode (see next) wordwrap and autoreformat are ignored, so the flags are dimmed.

(13) The final edit-mode toggle concerns hyper mode and ascii mode. In hyper mode, symbolized by a "_" character, Orpheus hides all linking codes and displays your text link anchors in whatever colors you have assigned to them. In ascii mode, symbolized by a "_" character, Orpheus displays the real contents of your cards, letting you edit the codes manually if   you wish. You can toggle from one to other by clicking on the symbol, or through the Edit Menu.

(14,16) Square brackets on either side of the line number indicate that you have set a temporary left or right margin. Pressing <Ctrl-[> sets a temporary left margin at the cursor position; the same keystroke cancels the tempory margin.

(15) Indicates the line on which the cursor is positioned, from 1 to 9999.

(17) The final item on the Statusbar is the name of the current file. If the file has been changed since you last saved it to disk (as shown by an asterisk beside the window number), you save it to disk by clicking on the filename, or through the File Menu.

Using Menus

Orpheus (OH.EXE) is a feature-rich program that new users may find a little complex. Fortunately, the menu system makes it easy to find your way around and quickly become an expert. The best way to learn Orpheus is to browse the menus while taking advantage of online Help. For an explanation of any menu item, place the selection bar on it and press <F1>, or click on it with the right mouse button.

Pressing <Esc> opens the menu system and preselects the name of the last menu opened. To select a menu, click on its name or press the key corresponding to its first letter, or press <Enter> after using the arrow keys if necessary to move the selection bar. To accomplish all that with a single keystroke, press <Alt> with the first letter of the desired menu, e.g. <Alt-L> for the Link Menu.

Once the menu system is open, you can change menus with the left/right arrow keys; for example, if the File Menu is open, press the right arrow to open the Edit Menu. To select an item on a menu, move the selection bar to it (using the up/down arrow keys) and press <Enter>. Alternately, just press that item's hilited letter, or click on it with the left mouse button. A very few menu items, such as the first three on the Edit Menu, allow you to change a setting without closing the menu. For example, if you click on "Edit mode" on the Edit Menu, or press <Enter> or <Space> with that command selected, the setting changes but the menu remains open. To close the menu, press <Esc> or click anywhere outside the menu.

While the menu system makes Orpheus easy to learn, keyboard accelerators let you bypass the menu as you quickly become an expert. Accelerators for menu commands are shown right on the menus, while others (for cursor control and so on) are listed in online Help under "Keyboard   Shortcuts". If you redefine the keyboard (see the Setup dialog on the Other Menu), you'll notice that the menus reflect any related changes you make, though Help does not.

Using Help

As mentioned above, Help is available on any menu item. It is also available in any dialog -- indeed, any situation whatever -- by pressing <F1>. A particularly useful facility of Help is that you can return to the previous help-topic (after leaving Help) by pressing <Alt-F1>. This makes it easier to learn new procedures: you can read the instructions for what you wish to do, exit Help and do as much as you can, then press <Alt-F1> to return to exactly the instructions you were reading before.

Virtually all Help topics include hypertext links connected to other topics. The presence of such a link is indicated by a colored word or phrase, called a link anchor. To jump through the link, click on the anchor, or move the selection bar to it and press <Enter>. Use the <Up> and <Down> arrow keys to move the selection bar, or to scroll as the case may be. Pressing <Right> is the same as <Enter>, to jump through the selected link anchor if any. Pressing <Left> is the same as <Backspace>, to retrace your steps after a series of jumps. You can also move the selection bar by pressing the first letter of any visible link anchor.

You may be interested to know that every feature of the Help interface, including the screen design, the command buttons along the bottom of the screen, and even the keyboard interface

described above, was decided by me using exactly the same authoring procedures that you will use in working with OH.EXE. Orpheus Help was in fact written as a project using OH.EXE, and the Help interface is encoded in that project, NOT in any program code. The program that performs Help, OHHELP.EXE, is a limited subset of the Orpheus Reader. Though normally it is called from within OH.EXE, you can also run OHHELP.EXE as a standalone program.

Using Dialogs

Most of what you will do in Orpheus, apart from editing text, is accomplished through dialogs. If you are familiar with virtually any contemporary software, you already know how to use the dialogs in Orpheus. The results of your dialog selections may be applied to immediate acts   like making a hypertext link, or may be optionally stored in (for example) a configuration file. All dialogs can be cancelled, and all dialogs have complete context-sensitive Help.

In many dialogs, related items are grouped together in boxes. Before the commands or actions in a box can be accessed, the box must be hilited. To hilite a box, click on it with the left mouse button, or move the selection bar with the <Tab>, <Shift-Tab>, or <Arrow> keys. If you are already in a box and wish to move out of it (to another box or to a non-boxed command), either click on the desired location or use the <Tab> or <Shift-Tab> keys. (Within a box, <Up> and <Down> move the box's selection bar, while <Left> and <Right> do nothing.)

To select or activate a command in a dialog, click on it with the mouse or move the selection bar to it and press <Enter>. Some commands have hilited action-keys that you can press, while others use non-hilited named keys, like <F1:Help>. The hilited keys in a box can only be used when you're in that box; similarly, when you're in a box the only hilited keys you can use are those in the box. However, named keys like <F1:Help> can be used at any time, whether you're in a box or not.

While most dialogs are pretty standard -- select an option and move on -- a small number have a specialized interface appropriate to their special purpose. An example is the dialog for changing colors (see the Other Menu->Setup Orpheus->Change Colors). Just remember that you are   never more than a click or a keypress away from online Help. Every dialog's Help topic begins with an illustration of that dialog, with each element and procedure thoroughly explained.

Customizing Orpheus

Many parts of the Orpheus environment can be customized to suit your preferences, needs and work habits. (You can also customize the appearance and behavior of the Orpheus Reader when it loads your finished project, but that is another topic.) As an overview, this section is designed to encourage you to try to get the most out of the authoring environment -- the most utility and the most enjoyment. Consult online Help for more information when do begin customizing.

All customization begins through the Setup Orpheus dialog, which is accessed through the Other menu. Before examining the various items in this dialog, notice that there are two ways to *exit* the dialog: by pressing <F10:Save CFG>, and by pressing <Esc:Cancel>. With the first method, your new settings are saved to a configuration file named OH.CFG. The next time you start up OH.EXE, Orpheus will read this file to determine how you want things to work. With the second method however, your new settings are not saved to disk -- but they do remain active for the current session. This lets you experiment with various options without committing to them (and of course you can always reconfigure even after saving OH.CFG).

The sub-dialogs accessed through Setup Orpheus, such as the Color dialog, all have <Esc:Cancel> buttons that really do cancel changes, restoring whatever settings were in effect before you entered that dialog.

CHANGING COLORS

The Color dialog lets you set up color schemes, not only for your work environment in Orpheus, but for the finished product as displayed by the Orpheus Reader. If you wish, you can store up to 9 different color schemes, called palettes, whether to suit the changing demands of your eyes or to create a different mood for every project. Each time you assemble a finished copy of your work (the version used with the Reader), Orpheus builds into it a copy of your *current* color scheme. Normally the end-user will therefore see your work in the same colors as those in which   you created it. This is a very good reason to make sure that you give the user access to the Reader's Color dialog; the colors that you find relaxing or stimulating may be unbearable (or uninteresting) to your intended readers.

All Orpheus programs provide color data appropriate to the user's video display hardware, which may be monochrome, grey-scale, or color. Suppose you as an author have a color display: users reading your finished work in color will see the color scheme that you designed, but users with mono or grey-scale displays will see the appropriate *default* schemes. (However, if you have access to all three types of video display you can design and store color schemes for all of them; see Help for details.)

PRINT OPTIONS

Printing support in Orpheus is intentionally pretty basic. You can make a standard printout, or export a card or Help topic to an external disk file (print to disk). If you need more refined printing, the solution is indirect but simple: use the print to disk feature, then import the resulting text file into your favourite word processor and go on from there.

When you use the Print command from the File menu, the Print dialog lets you make temporary changes to your print options before you actually print. To make changes that can be saved to OH.CFG, use Print Options from the Setup Orpheus dialog.

VIRTUAL MEMORY

Virtual memory is the technique of extending conventional memory resources by means of swapping: copying the contents of an area of memory to another location, such as XMS memory or a disk file, in order to reuse that area for some other purpose. Orpheus uses virtual memory during normal editing and project building, or rather it does so when necessary. If you develop large projects with long text cards, Orpheus will use virtual memory a great deal. In this case you should configure Orpheus to give you the best performance possible on your system.

If you have XMS memory available, Orpheus will show this in the lower right quadrant of the Virtual Memory dialog. For best performance, tell Orpheus to use XMS. If you do not have XMS but do have EMS memory, you may be able to achieve performance similar to that with XMS. The trick is to use some or all of that EMS as a RAMdrive; consult MS-DOS Help on RAMDRIVE.SYS (or your DOS manual) if you don't know how to do this. With a RAMdrive set up, return to the Virtual Memory dialog and set the swapfile path to the drive letter assigned to the RAMdrive.

The Virtual Memory dialog can also be used to check out how much conventional DOS memory is free for Orpheus to operate with. Ideally, before starting up OH.EXE you should have at least 512K free RAM; you can find this out by running the MS-DOS "mem" command. Returning to   the Virtual Memory dialog, check the "low RAM free" report. If it is lower than 48K you may need to free up additional DOS memory if you develop large projects.

DEFINE KEYBOARD

The Define Keyboard dialog lets you customize OH.EXE's response to the keyboard. This is especially handy for common editing tasks, since you can adapt Orpheus to behave the same as your favourite word processor. Orpheus comes with a small number of built-in keyboard assignments, most of which are accelerator keys for tasks listed on the menus. You can redefine the response to any keypress except <F1> and <Esc>.

A complete list of the default keyboard assignments is given in Help under "Keyboard Shortcuts". This list does not reflect any changes you might make, so for an accurate account you must browse the list in the Define Keyboard dialog. You'll notice that changes you make *are*   reflected on the menus, where keyboard accelerators are hilited; however, Define Keyboard gives you access to over 60 functions that are not on any menu and in some cases do not even have a default keypress.

Note that this dialog only affects the keyboard response of OH.EXE, in other words your working environment as an author. For project-specific control over how the Orpheus Reader responds to the keyboard, see Help under "Customizing a Project".

EXTERNAL EDITOR

Despite the customizations described above and in the next few paragraphs, you may prefer to use a more powerful (or more familiar) program for serious text editing. Provided you choose a DOS text editor that handles files in plain text format, you can do this quite easily, returning to Orpheus only for tasks related to hypertext. Programs recommended for this purpose include Multi-Edit for DOS from American Cybernetics, TSE from SemWare, and PC-Write.

To prepare for use of an external editor, start by going through Setup to the External Editor dialog, where you can set options to be saved in OH.CFG. To actually run the program you will go through the Edit menu and select "External Editor" from there; as with printing, you can then make temporary changes to your options that will not be saved to OH.CFG.

Note that the External Editor commands can be applied to *any* compatible DOS program, such as a spell-checker. After running an external program Orpheus checks each Edit window to see whether the file it contains has changed, and if so, loads in the newer copy from disk.

CURSORS

Use the "Cursors" box to set the appearance (size) of the cursor in Pushright and Overwrite modes. For example, if you are using a laptop and do not have a utility to make the cursor more visible in all programs, you can use this dialog to make the cursor a block. Most users like to use the standard line cursor in Pushright mode and a block cursor in Overwrite mode.

TABS

Use the "Tabs" box to set the number of spaces in a tab, and optionally to make <Tab> insert spaces instead of just moving the cursor.

MISCELLANEOUS

There are three options you can set in the Miscellaneous box. The first, ";Text-card comments", refers to how you want Orpheus to interpret lines beginning with a ";" character. If you select this option, Orpheus ignores such lines during compilation; you can use this feature to make notes to yourself that will not go into your finished work. The second option, "Swap for DOS shell", tells Orpheus whether or not to swap itself out if and when you shell to DOS from the File menu. The swapping code for this function will use XMS, EMS, or a disk file, in that order of preference, depending on availability. If neither of the first two is available and you have lots of conventional memory left over, you may want to leave this option off. The final Miscellaneous option, "Quit restores blink", tells Orpheus whether to restore blink or maintain high-intensity backgrounds when you exit the program.

CHAPTER 3 - TUTORIAL : A HYPERTEXT PROJECT

....

Topics in this chapter:

What is a Project?
How Do You Develop a Project?
Starting a New Project
Expanding a Project
Importing Text
Navigating Hypertext in OH.EXE
What's in a Card?
Finishing a Project

What is a Project?

In one sense you already know what a project is: it's that electronic guide, manual, book or infobase that you hope to create and distribute using Orpheus. But in terms of actual files on disk, what form does a project take during development, and what will it look like when finished?

Here is a schematic of a typical project, showing directories in lowercase letters, filenames in uppercase. The illustration assumes that you have installed Orpheus on drive c: in the \oh2 directory, and that the project is named "TEST", but they can be whatever you like. Note that only the first item, the working directory, is created by you; everything else is generated automatically by Orpheus when needed.

```
c:\oh2\              ...the "working directory" (you make this)
c:\oh2\TEST.PRJ      ...the "project file" for TEST
c:\oh2\test\         ...the "project directory" for TEST
c:\oh2\test\dir1\    ...its first "storage directory"
c:\oh2\test\........ ...more storage directories as needed
c:\oh2\test\dir9999\ ...up to 9999 storage directories
```

The place where you begin work on a project (your current directory when you start up OH.EXE) is called the working directory. This is the only directory you set up yourself. You can use the same working directory as the starting point for any number of projects; it can be wherever you installed Orpheus 2.

Except for the project file, everything in your work in progress is stored in the project directory and its numbered subdirectories. The parts you will edit yourself -- text you compose or import, and scripts to display graphics or perform various actions -- go into special text files called "cards", which are named and stored as shown below. These really are text files; you can open them with any text editor or file viewer. They are special because of their filenames, and because each begins with a one-line "header" with information used by Orpheus.

```
c:\oh2\test\dir1\1-001.TES  ...a "card" file
c:\oh2\test\dir1\.........   ...more cards generated as needed
c:\oh2\test\dir1\1-256.TES  ...up to 256 cards per storage directory
```

Card filenames begin with a two-part number separated by a hyphen, and end with a three-letter file extension. The first part of the number corresponds to the directory containing it: in

dir1, filenames start with "1-"; in dir2, filenames start with "2-". The second part is a 3-digit number from 001 to 256, corresponding to the fact that a single storage directory can hold up to 256 cards. (In dialogs and linking codes you can leave out the extra 0's; for example, 1-1 is the same as 1-001 for all purposes except filenames.) Finally, the three-letter file extension can   be whatever you wish, provided it is the same for all of the cards in the same project; by default, it is drawn from the name of your project.

Since there can be up to 9999 storage directories per project, with up to 256 cards per directory, you can have up to 2,559,744 cards -- each of unlimited length -- in a single project. (For comparison, the OH project here at Hyperion Softword, provided for your use as online Help, handles over half a megabyte of text in a mere 120 or so cards.)

Naturally, the FINISHED version of your project is much more compact, consisting of one or more files generated through the final assembly stage of the Build process (discussed later in this chapter). Compiled, compressed, and encrypted, these files cannot be edited or changed by other users. However, building a finished product has no effect on the unfinished version of your work; therefore, you can expand and edit a project indefinitely while generating new versions of the finished product whenever you wish.

As shown below, your finished work is deposited in the working directory as one or more files named after the project itself. Often there is just one file, normally with "HTX" as the file extension:

          c:\oh2\TEST.HTX   ...distribute this with the Orpheus Reader

With large projects there may be additional numbered files:

          c:\oh2\TEST.HTX   ...distribute this with the Orpheus Reader, plus:
          c:\oh2\TEST.001
          c:\oh2\TEST.002   ...up to 999 numbered files if needed

If you use the Orpheus Professional option called "Executable HTX", the file that would normally end with an HTX extension is merged with the Orpheus Reader to become a self-opening book:

          c:\oh2\TEST.EXE   ...distribute this only (no separate Reader)
          c:\oh2\TEST.001   ...plus any additional numbered files if created

These illustrations do not show any external files referred to in your work, such as graphics or other programs launched from a script. You would have to keep track of them separately.

How Do You Develop a Project?

From start to finish, developing a project consists of three phases: starting it, expanding it, and generating a finished version of it to distribute to end-users.

(1) The first phase, starting, is very brief. Using the "New Project" command on OH.EXE's Project menu, you tell Orpheus the basics of what to name your project and where to store it. Orpheus then produces certain startup components (files and directories), comprising the seed from which to grow your work in whatever direction you wish. A step-by-step guide through this phase is given in the next section, "Starting a New Project".

(2) The second phase, expanding (along with revision and so on), continues indefinitely for as long as you have an unfinished copy of your work. Expanding a project is basically a matter of

adding new card files and putting something in them. Normally, this is done through one of several commands on the Link menu that simultaneously generate a new card and add linking codes that point to it from your current location. A step-by-step guide to the most common procedure is given in a later section of this chapter, "Expanding a Project".

(3) The third phase, finishing, is as brief as the first -- but can be repeated whenever and as often as you wish. Imagine that you have published a beautiful book, complete with full-color graphics and leather binding. The morning after your launch party you find some typos, or get inspired to write a new chapter, or come across some groovier graphics. With a printed book you'd be stuck; but with hypertext you can just do what you want, tell Orpheus to build a fresh version of the finished product, then have another launch party. All aspects of building a finished version are handled through the "Build Project" command on OH.EXE's Project menu. A step-by-step guide to the basic procedure is given in the final section of this chapter, "Finishing a Project".

Starting a New Project

In this section we'll go step by step through the process of starting a new hypertext project in Orpheus. Our examples will assume that your working directory is "c:\oh2" and that the test project will be named "TEST".

Start up OH.EXE at the DOS prompt using the command "oh"; if using Windows or OS/2 do the same as you would to start up any other DOS program (such as File->Run in the Windows Program Manager). When Orpheus finishes loading you will see its statusbar on the top line, with the cursor under the top left corner, and "?Unnamed?" in the upper right. Now press <Esc> to open the menu system, followed by <P> to pull down the Project Menu. (Pressing <Alt-P> would do the same in one step.) The selection bar is already on "New Project", so press <Enter> to select it. This initiates a sequence of dialogs to guide you through the process; at the conclusion of each dialog Orpheus moves automatically to the next.

In **Step 1**, Orpheus opens a dialog with the title "Enter a New Project (.PRJ) Name". If you enter a name without an extension, Orpheus appends ".PRJ" to form the name of the project file; but if you do include an extension, it must be ".PRJ". The root part of the name is then used for the project directory and the other files discussed above. To continue this tutorial, type in "test" and press <Enter>.

In **Step 2**, Orpheus opens the Project Options dialog. Notice that the dialog title includes the name of your project: settings you make in this dialog apply to the current project only. As you will see, there are lots of things you can do here, in particular customizations of how the Reader will look and respond when it opens your hypertext.

> Read about these options in online Help. With one exception, you can change any option at any time during the life of a project; just open the Project menu, select "Project Options", and do it. The exception is the "Card extension" item in the group box entitled "Your project". This refers to the file extension that will be used for all of the uncompiled cards and certain other files in your work in progress; you can ONLY change this now, during Step 2 of starting a new project.

For this tutorial we'll only make one setting, the "Title" for your project. Type in something appropriate like "Testing, Testing, Testing!" and press <Enter>. Then press <F10:Done> to conclude the dialog.

In **Step 3**, Orpheus opens the Define Card dialog. Whenever a new card is created, this dialog comes up so that you can immediately determine certain properties of that card. (You can

change these properties later by selecting "Define Card" from the Card menu.) Since the most common use for this dialog is to edit a card's Topic entry, you start right off at the Topic item. A Topic describes the subject or purpose of a card; if you make a habit of ALWAYS entering the Topic when making a new card, you will find that you have a rich resource with which to provide additional forms of access to your work, plus a superb outline for your own use. In this case, Orpheus is generating the Homecard for your project, so it would be appropriate to type in something like "Table of Contents".

With the last dialog concluded, you are in the Homecard of your new project. That is, you are in an apparently empty text file that happens to be the project's Homecard. The filename, "1-001.TES" if you are following the tutorial exactly, is shown on the statusbar at the far right. The line number at the cursor, 1, is left of the filename. The current window number, 1, is beside the "Esc:Menu" prompt. And here's what is on your hard disk (details depend on your installation):

```
c:\oh2\TEST.PRJ          ...project file for TEST project
c:\oh2\test\             ...its project directory
c:\oh2\test\dir1\        ...its first storage directory
c:\oh2\test\dir1\1-001.TES   ...and its default Homecard
```

This constitutes the bare essentials of a project in Orpheus.

Expanding a Project

In this section and the one following we'll add flesh to your empty Homecard, setting it up like a table of contents. On each of three titles we'll make a link to a new blank card, then will demonstrate three different methods of putting something in those cards. Begin by typing the four lines of text below into your Homecard:
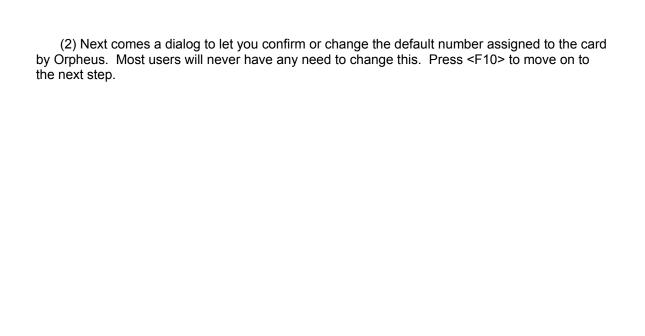
TABLE OF CONTENTS

How to Read This Book
What's New
About Orpheus

Next, move the cursor to any word in "How to Read This Book". (Since the number keypad is normally used for hypertext navigation in OH.EXE, you should use the middle keypad for cursor movement. Alternately, move the mouse pointer to the desired location and click the left button to set the cursor there.)

Open the Link menu by pressing <Esc> and then <L>, or simply <Alt-L>. We'll make the standard link type, which is called a "Door" in Orpheus. Since the selection bar is already on the Door command, press <Enter> to select it. This leads to the following set of dialogs:

(1) First comes the Resize Text Anchor dialog. Notice that the word where you left the cursor before opening the Link menu is now hilited. This represents the default link anchor. In this case it would make more sense to hilite all of "How to Read This Book" instead of just one word, so the purpose of this dialog is to help you do that right away. The box on the left lets you extend the hilite to either side; the box on the right lets you retract it from either side. To do this with the keyboard, press <Tab> to select the appropriate box, then press the <Left> or <Right> arrows to extend or retract the hilite. Using the mouse, just click on the box and then click on the appropriate arrow. When the link anchor includes the full section title, press or click on <F10:Done>.

(2) Next comes a dialog to let you confirm or change the default number assigned to the card by Orpheus. Most users will never have any need to change this. Press <F10> to move on to the next step.

(3) This brings you to the Define Card dialog, which you saw earlier when the New Project procedure brought you to the Homecard. The text of the link anchor is offered as the default topic of the new card. Press <Enter> when the topic is to your satisfaction. Normally you won't change any of the other options right away, except perhaps the one for "Scroll text" if you left scrolling optional in the Project Options dialog. Notice that in the "Built-in links" box, the "Origin" field contains "1-1"; this means that your new card will have an Origin link to card 1-001,   where you are right now. Press <F10> to conclude this dialog.

(4) The last dialog lets you select an Edit window in which to load the new blank card. (Up to this point, if you cancelled one of the three preceding dialogs by pressing <Esc> the entire process would be aborted.   However, once you've gotten this far the new card has been created and   the linking codes inserted. Cancelling this dialog merely leaves you in the current window without loading the new card.) In OH.EXE you can open up to 9 Edit windows. The dialog lists the first window, containing the Homecard, but hilites window #2 as the logical choice for loading a new card. Press <Enter> to accept this.

Finally you are in your new blank card. Notice that the filename at the upper right says "1-002.TES", and that the number at the left (beside the Esc:Menu prompt) indicates window #2. A little later we'll see that there is absolutely no connection between windows and cards, since any window can hold any card or other text file, and you can use any window to navigate throughout an entire project. For the moment, flesh out the new card by typing in the following two lines of text:

> If I've forgotten to provide special instructions for how to
> read this book, just press <F1> and the Reader will explain.

This illustrates the most obvious approach to developing a hypertext project: typing in new text as the project grows. To save this card to disk, open the File menu and select "Save", or click on the filename on the statusbar.

Importing Text

In this section we'll continue expanding the Test project, showing two methods of doing so that involve importing text from external files.   First we'll import an entire file in one step; then we'll load a long file in a separate window and import just a block of it.

To begin, return to the window containing the Homecard. (You can do this by clicking on the window number, by pressing <Ctl-V> or <Alt-V>, or by using the Window List dialog from the Window menu.) Move the cursor to the first letter of "What's New". We're going to make a link on this title, but this time we'll mark the link anchor before opening the Link menu. This will let us skip the Resize Text Anchor dialog.

With the cursor under the "W" of "What's New", press <F3> to begin marking. Using the middle keypad, move the cursor to the right until the entire title is hilited, or click on the space just to the right of "New".   (If you are using an older computer and the right arrow key just makes Orpheus say "bonk", press the NumLock key to switch out of navigation mode.) With the entire title hilited, press <F3> again to end marking.

Now open the Link menu again, press <Enter> to select the Door link type, and follow the dialogs as you did before. Notice that this time you don't go through the Resize Text Anchor dialog, since you've already marked your link anchor as desired. When you reach the Window dialog,   press <Enter> to load the new card into the window suggested by Orpheus, which should be #3.

We'll flesh out this card by importing an entire external text file, in this case the "UPDATE.TXT" file that came with your copy of Orpheus 2. Open the File menu and select "Import File". If your working directory is the one in which you installed Orpheus, just type in "UPDATE.TXT" and press <Enter>. If the Orpheus directory is somewhere else, type in the complete path to the file or else <Tab> into the file and directory and boxes until the desired filename is shown in the entry field.

With the procedure completed, the entire Update file has been copied into the card in your Orpheus project. Save this card by clicking on its filename on the statusbar.

Return once more to the window containing the Homecard, and make a third link on "About Orpheus". Follow the dialogs as before, and again load the new blank card into a new window, which should be #4. We'll flesh out this card by importing only a block of text copied from a much larger file, which will be loaded in another window. The file we'll use for this purpose is the manual you are now reading.

Open the File menu and select "Open". If your working directory is the one in which you installed Orpheus, just type in "MANUAL.TXT" and press <Enter>. If the Orpheus directory is somewhere else, type in the complete path to the file or use the file and directory boxes (pressing <Tab> to move into or out of a box) to select the desired file. When you conclude this dialog, the file is loaded into the next available Edit window, which in this case should be #5. (If all 9 windows were occupied Orpheus would show you the window list to let you choose which one to use.)

You should now be looking at the first page of MANUAL.TXT. Page down past the copyright line to the first paragraph describing Orpheus. We'll mark the three paragraphs so they can be copied in a block to your new card. To make this easy we'll mark the block using line mode. With the cursor anywhere on the first line of the first paragraph, press <F7> to begin marking in line mode. Use the <Down> arrow key to move the cursor down to the last line of the third paragraph. Then press <F7> again to finish marking the block. (You can also access block-operation commands through the Block menu.)

For the next step, return to the window containing your new blank card, using <Alt-V> or by clicking on the window number until you're there. To insert the marked block from the other window, press <Shift-F3> (or open the Block menu and select "Copy from window"). A dialog will open with a list of all windows containing marked blocks; since there is only one and it is already selected, press <Enter> to confirm it. The marked block is now copied into your new card. Press <F5> to unmark this copy, then save this card.

At this point your Test project contains three cards linked to the Homecard. All four cards are in separate full-screen windows, and MANUAL.TXT is in yet another window. If you were to exit the program now, the next time you loaded the TEST project you would find yourself in the current window, with all the other windows having exactly the same contents as when you left them. (You can reload a project either from the DOS prompt, giving the project name on the command line, or by selecting "Open Project" from the Project menu.) Now, what about those links you made? Do they really give you hypertext, or do you have to switch windows all the time to jump from card to card? In the next section we'll see that the links really do work -- and that the Edit windows are only there for your convenience while authoring.

Navigating Hypertext in OH.EXE

    To test hypertext navigation in OH.EXE, we'll start by closing all windows except the one containing the Homecard. Return to the window containing MANUAL.TXT and press <Alt-F2> to close that window. This should put you back into window #4, so press <Alt-F2> to close that one, and continue until you are back in window #1 with the Homecard. If you accidentally closed the window containing the Homecard, don't worry about it: open the Card menu and select "Home" to jump to the Homecard, and complete the jump in the current window (#1).

    You should now be looking at the Homecard with its three colored links to other cards. We're going to jump back and forth between each of those cards and the Homecard, first of all in window #1, then with the option of loading the target card in a new window. If you are using a mouse, it's easy: using the *left* mouse button, click on a link anchor to set the cursor on it, then click on the cursor to make the jump. Do this now on any of your three links, then retrace your steps (in this case to the Homecard) by clicking the *right* mouse button. Note that in Help and the Reader it's even simpler, since there's no cursor: you just left-click once on a link to jump through it, and right-click to return. For more about navigating with the mouse, read "Navigating Hypertext" in online Help.

    If you are using the keyboard, navigation is a little more complicated but more powerful in return. First, notice that all of the navigation commands are available on the Card menu, where the default accelerator keys are given as requiring the Shift key. What the menu does not show is that those accelerators are limited to the NUMBER KEYPAD, and if you are using a full-sized keyboard on a 286 or higher you do NOT have to use Shift. However, if you are using an XT (with or without a full-sized keyboard), OR if you are using certain laptop or notebook computers, you may have to turn on NumLock or add Shift as shown on the menu. In the rest of this section I'll assume that you are using a full-sized keyboard on a 286 or higher. Later on, you may wish to customize the keyboard for whatever combination of cursor control and navigational commands you find most convenient. This is done through the "Setup Orpheus" dialog on the   Other menu.

    Continuing now with the keyboard: using the middle keypad, OR the <Up> or <Down> arrows of the number keypad, move the cursor to the first link, "How to Read This Book". Now using the number keypad, press <Right>. This should jump you through the link and to the attached card,   without leaving window #1. To retrace your steps, press the <Left> arrow on the number keypad. (You could also press <Home>, but the point is that <Left> retraces your hypertext jumps no matter where you are, whereas <Home> goes straight to the Homecard.) Repeat this procedure with the second and third links, just to verify the hypertext connection.

    Now we'll do the same thing but with the option to load the target card in another window. The purpose of this ability is to make your work easier while authoring; there is no equivalent in the Reader. To use the window facility, hold down the Control key while giving your normal navigational command. This applies whether you are using the keyboard or the mouse. After jumping to each card, give the same retrace command as you did before (right-click or the <Left> numpad arrow) *without* using the Control key. Notice that Orpheus returns to window #1 because the Homecard (your destination) is already loaded there. Repeat the procedure until all three child cards are loaded in separate windows.

    To find out more about navigating hypertext in Orpheus, be sure to check online Help on the topic "Navigating Hypertext".

What's in a Card?

Now that you have made some links (and some cards), and used them, it's time you saw what is actually in those cards that enables Orpheus to give you hypertext linking. Return to the Homecard if you are not there already. Now press <Ctl-F1>. This switches Orpheus into "ascii mode", in which it displays the card's true contents. As you can see, in ascii mode your link anchors are shown with the embedded codes revealed. This is exactly what you would see if you were to load the Homecard file in any standard text editor or file viewer. If you press <Ctl-F1> again Orpheus will return to "hyper mode", in which it displays your links the way your end-users will see them.

Still in ascii mode, press the <Up> arrow to move the cursor to the top of the Homecard, then once more to scroll up to a line that was formerly invisible. This line is the card header, and will look something like this:

OH2(TEXT,"Table of Contents")

Every uncompiled card in an Orpheus project begins with a header line of a similar nature. The header tells Orpheus what type of card it is, what its topic is (if any), and so on. For further discussion of cards and the various contents of header lines, read "Hypertext Cards" in Help. (You don't need to know anything about headers to use Orpheus, as long as you are aware that they exist.)

If you press <Ctl-F1> a few times to toggle back and forth between ascii mode and hyper mode, you may notice that besides the changing display of your text, there is one small change on the statusbar. In hyper mode, the symbol to the left of the line number is a figure 8 on its side (the infinity symbol). In ascii mode the symbol is a stack of three horizontal lines. The current mode is also indicated when you open the Edit menu.

Let's take a closer look at your three link anchors in ascii mode. Unlike the card header, which you should never need to edit manually, link anchors and the other embedded codes are good things to know about. Depending on the work you do, you may never need to leave hyper mode, and if that's the case, you'll never edit a linking code manually. But it's comforting to know that should you ever need to create or edit the linking codes yourself, you CAN do so, and do it easily; and for a few advanced functions you do in fact have to.

As you can see by the patterns repeated in the three sets of linking codes on your Homecard, every link anchor begins with a "<" left angle bracket, which is followed by the visible text of the link anchor. Then comes a "|" pipe symbol, followed by more codes and a final ">" right angle bracket. The "<|>" characters are the skeleton of every embedded code in Orpheus, with visible text going on the left of the "|", and the actual codes on the right. In the case of your three links, the codes all begin with a "D" (if you have followed the tutorial exactly), which represents a Door link. This is followed by a two part number -- "1-2", "1-3", and "1-4" -- which you will recognize as referring to the linked cards 1-002.TES, 1-003.TES, and 1-004.TES respectively. (In filenames the second part of the number must be padded with 0's to make three digits, as in "1-002"; but in link references and dialogs you can omit the 0's.)

Although there's a lot more to linking than we can cover in a brief tutorial, there is certainly no mystery about it, nor is Orpheus particularly complicated. Read up on "Hypertext Links" in Help, check out the individual commands on the Link menu, experiment.

Finishing a Project

In this section we'll build a finished copy of the Test project, and see how it runs in the Orpheus Reader. Then we'll examine the virtues of incremental compilation, and try out the Simulator.

With the Test project loaded in OH.EXE, open the Project menu and select "Build". (For a full explanation of everything you see, press <F1> while in this dialog.) The most important part of the Build Project dialog is the "Action" box in the lower right corner. As you can see, it lets you choose among 7 combinations of three actions: compile, verify, and assemble. We need all three steps to build a finished product, so press <4> to select "Compile->Verify->Assemble". Then press <F10:Begin> to set the builder in motion.

If all goes well, Orpheus will swiftly compile your cards, verify all links accessible from the Homecard, and assemble the results into a finished product named after your project: in this case, TEST.HTX. When building is completed, press a key to clear the report window.

Now let's try out the Reader! Exit OH.EXE, either through the File menu or by pressing <Alt-F4>. Start up the Reader (OHREAD.EXE) with the following command at the DOS prompt (we assume that the project name is "test"):

       ohread test

The Reader should open your project to the Homecard. Unless you have already played around with customizing your project options, the Reader will display your work using the default Reader interface and the default screen layout, which includes a simple titlebar. All of this can be extensively customized.

Press <Esc> to open the Reader's menu system, and explore a little. You can customize, on a project-specific basis, end-user access to features like printing, or even to the menu system as a whole. Press <F1> while in any dialog or while looking a text card (but not while in a menu). Notice that the Reader provides a series of simple Help windows for all built-in features -- but this too can be customized.

Try out your links, observing how the Reader responds to keyboard and mouse input. Apart from the fact that everything is simpler (there's no distinction between the number keypad and the middle keypad, for example), you also have a choice between two sets of keyboard assignments. Actually you're seeing only a fraction of what you *could* see, because the fact   is that through OH.EXE's Project Options dialog you can customize your work so thoroughly that the Reader will look as if it was programmed just for you.

When you're ready, exit the Reader through the File menu or by pressing <Alt-X>. If you have a robust file-viewer like LIST.COM, you might enjoy sneaking a peak at the contents of your finished product, TEST.HTX. What you'll see is: completely indecipherable garbage. In other words, only the Reader can read it, and no one can hack a finished version of your work.

Now let's get back into OH.EXE to check out something else. Load up the Test project again, either by giving its name on the command line ("oh test"), or by using the "Open Project" command on the Project menu. We're going to see how you can test your finished work without leaving OH.EXE, and without actually building a fully finished copy. We'll do this using the Simulator (OHSIM.EXE), which you can run both as a program in its own right, or (more conveniently) from within OH.EXE.

Open the Project menu and select "Simulate Reader". This tells Orpheus to fire up the

Simulator; if all goes well, a message will appear reminding you of the Simulator's "emergency exit", which is <Alt-F4>. (The emergency exit is there in case you intentionally disable the standard exit hotkey -- as a project option -- and forget to replace it with some other way for users to exit your work.) Once past that message you should see exactly what you saw in the Reader. However, if you try to open the menus you'll discover that the Simulator doesn't have any, because it isn't a full copy of the Reader. The Simulator has to do certain things that the Reader does not, so in order to keep the program a reasonable size any unnecessary parts have been left out.

Unlike the Reader, which uses a *fully assembled* copy of your work, the Simulator only uses the compiled (but not assembled) portions of it, which reside in your project's "CMP" files. CMP files are generated during the compile stage of the build process; there is one in each storage directory, to hold compiled versions of all of the cards in that directory. The assembly stage puts together your finished work by copying these compiled parts found in the CMP files.

Let's demonstrate further the difference between Reader and Simulator. If you are still in the Simulator, press <Alt-F4> to exit it. With your project loaded in OH.EXE, type a line or so of text in whatever card you happen to be in, and save that card. Now open the Project menu and select "Compile, Simulate". This tells Orpheus to look for any cards that have changed or been added since the last compilation, compile them, and then open the Simulator. When the process is complete you will again be in the Simulator, looking at the compiled card with the new text that you just typed in. However, if you were to exit the Simulator and OH.EXE and then load up your project in the Reader as we did earlier, you would find that the Reader displays the old version of the card, because the HTX has not changed. To update the HTX, open the Build dialog and choose any action that includes the assembly stage.

Though the Edit windows of OH.EXE give you a pretty good view of what your work will look like in the Reader, the Simulator takes you one step closer to the goal without the extra processing required to build the final product. Besides this purpose, the Simulator is also used for testing scripts and adding hotspots to graphics.

Summing up: for a real project, you would devote a certain amount of time to tuning your project options and customizing the appearance and functionality of the Reader interface. This and many related topics are covered extensively in online Help: on the Help Contents page, select   "Finishing a Project" and explore. Over time you will dig into Help on various topics encountered in the Project Options, Build Project, and other dialogs, but some of the topics reached through "Finishing a Project" touch on wider considerations, such as distributing your work to end-users.