FHI O-PAK

FHL O-PAK

For the 64K TRS-80
Color Computer with OS-9

FHL FRANK HOGG LABORATORY

THE REGENCY TOWER • SUITE 215 • 770 JAMES ST. • SYRACUSE, NY 13203
PHONE (315)474-7856 • TELEX 646740

FHL O-Pak ©1983 Frank Hogg Laboratory, Inc
770 James Street
Syracuse, New York 13203
315-474-7856
All Rights Reserved

COPYRIGHT INFORMATION

DISCLAIMER

The supplied software is intended for use only as described in this manual. Use of undocumented features or parameters may cause unpredictable results for which Frank Hogg Laboratory, Inc. cannot assume responsibility. Although every effort has been made to make the supplied software and its documentation as accurate and functional as possible, Frank Hogg Laboratory, Inc. will not assume responsibility for any damages incurred or generated by such material. Frank Hogg Laboratory, Inc. reserves the right to make changes in such material at any time without notice.

TABLE OF CONTENTS

1.0.0 GETTING STARTED 1
1.1.0 What is O-Pak? 1
1.2.0 System Configuration 2
1.3.0 Before You Begin 2
2.0.0 HIRES 4
2.1.0 What is HiRes? 5
2.2.0 General Information 5
2.2.1 Using HiRes 5
2.2.2 How to Undo HiRes 7
2.2.3 Pass-through Mode 7
2.3.0 Advanced Use of HiRes 9
2.3.1 Changing the Cell Sizes 10
2.3.2 Using a Different Character Set 11
2.3.3 Terminal Display Features 12
2.4.0 Programming with HiRes 18
2.4.1 Philosophy of HiRes 18
2.4.2 How HiRes Works 19
2.4.3 HiRes Termination 19
2.4.4 HiRes Control Codes 20
2.4.5 Additional HiRes Get Status codes 22
3.0.0 CHARACTER SET EDITOR 23
3.1.0 What is CSEdit? 23
3.2.0 Using Csedit 23
3.2.1 The Main Menu 23
3.2.2 The Main Csedit Menu 26
3.2.3 Final Note about CSEdit 32
4.0.0 THE 'X' COMMANDS 33
4.1.0 What is an "X" Command 33
4.2.0 General Information 33
4.3.0 Command Syntax 35
4.3.1 Xcopy 35
4.3.2 Xdir 35
4.3.3 Xdump 36
4.3.4 Xlist 36
Appendix A - Control Codes 37
Appendix B - Glossary 39
Appendix C - Index 41

1/GETTING STARTED

1.1.0 What Is O-Pak?

O-PAK is a three-part utility package. It is designed to enhance the OS-9 Operating System for the TRS-80 Color Computer. The utilities in O-PAK are designed to work only with Color Computer OS-9 and are not guaranteed to work with any other computer or operating system.

HiRes

O-PAK consists of HiRes, CSEdit, and four "X" commands. HiRes is a program which provides a higher resolution screen display than the standard 32 by 16 characters supplied by Radio Shack. With HiRes you have the ability to change character sets (a variety of which have been supplied on the disk) and to mix graphics with text on the same screen. Do this in two-color or four-color mode.

CSEdit

You may wish to modify some of these character sets - or create some new ones of your own! You don't like the way we make our number seven? Change it! It's possible with CSEdit, the Character Set Editor. CSEdit is a screen-oriented, menu-driven program which is designed to make changing your character sets simple and fun.

xcopy, xdir, xdump, xlist

As you begin to use OS-9 you may wish to transfer some data files from a Disk BASIC or FLEX-formatted disk to an OS-9 disk. The four "X" commands; XCOPY, XDIR, XDUMP and XLIST will assist you in doing just that.

1.2.0 System Configuration

To run O-PAK you need a 64K Extended BASIC Color Computer (purchased as such or upgraded to an equivalent), at least one disk drive and controller card, and Color Computer OS-9 (available from Radio Shack).

1.3.0 Before You Begin

Please take the following steps before you begin using O-PAK. This will protect you, and FHL, from the expense of replacing your O-PAK disk if you should ruin it.

Understand OS-9

Users of O-PAK are assumed to have a good working knowledge of the OS-9 operating system. This documentation is not designed to teach you how to use OS-9. It is designed to teach you how to use an OS-9 utility: O-PAK.

For information on High Resolution and graphics, refer to Appendix B on page 125 of the Radio Shack OS-9 manual 'OS-9 COMMANDS' (The red book).

Write-Protect It

To the right of the disk label is a notch about one quarter inch wide. If this notch is not already covered with a write-protect sticker, please put one on it before placing the disk in a drive.

Make a Backup Copy

O-PAK is distributed on a standard 35-track, OS-9 formatted disk. The easiest way to make a backup copy is to use the OS-9 Backup command.

First you must format a blank disk to hold the backup copy. Use the OS-9 command:

```
format /d0 ENTER
```

to format a new disk in drive 0. Then simply reinsert the OS-9 master disk and:

```
backup /d0 ENTER
```

Complete instructions on how to do this are contained in the Radio Shack OS-9 manual, "Getting Started with OS-9."

Additional Information

List the file "info" on the O-PAK disk for additional information about O-PAK that is not included in the manual. Use one of the following procedures to do this:

Single-Disk Systems:

- OS9:load list ENTER
- Remove your OS-9 System disk and insert O-PAK disk.
- OS9:list /d0/info ENTER

Dual-Disk Systems:

- OS9:list /d1/info ENTER

2/HIRES

2.1.0 What Is HiRes?

HiRes is a program which provides a high-resolution screen display for Color Computer OS-9. With HiRes, the device TERM (which simply refers to your Color Computer keyboard and television screen or monitor) is placed in Graphics mode all of the time. Then HiRes draws the characters. This allows you to have true lower-case letters, more characters per line and the ability to combine graphics characters with text characters on the same screen!

2.2.0 General Information

2.2.1 Using HiRes

The first thing that we have to do is to copy some files from the supplied disk to your system disk. Make sure that your system disk is not write protected. The easiest way to start HiRes is to follow these steps:

Single-Disk Systems:

- Boot up OS-9
- `OS9:load copy` ENTER
- Remove your OS-9 System disk and insert O-PAK disk.
- `OS9:copy /d0/hires /d0/cmds/hires -s` ENTER
- Insert your OS-9 disk when prompted.
- After the above copy is complete, reinsert your O-PAK disk.
- `OS9:copy /d0/stdcs /d0/cmds/stdcs -s` ENTER
- Insert your OS-9 disk when prompted.
- `OS9:hires` ENTER

Dual-Disk Systems:

- Boot up OS-9
- Put your O-PAK disk in drive one.
- `OS9:copy /d1/hires /d0/cmds/hires` ENTER
- `OS9:copy /d1/stdcs /d0/cmds/stdcs` ENTER
- `OS9:hires` ENTER

You should now see the "OS9:" prompt in the upper-left corner of the screen, in smaller characters than you're used to. This is the default 51 column by 24 line character set supplied with HiRes. It is called "StdCS".

Lower Case Letters

Now that HiRes is running, type the command "tmode -upc" so you may see lower case letters. This command is from OS-9 (not

O-PAK) and is what allows OS-9 to display lower case letters. However, upper case letters will still be generated by your keyboard until you "take off the alpha-lock." This is done by holding down the CLEAR key and typing a zero.

We suggest you now try some of the commands you're already familiar with, to see how they look with HiRes. How about using the "dir" command to explore the O-PAK disk?

Installing O-Pak

At this point you may wish to install the rest of O-Pak's utility programs. They are on the disk in the /CMDS directory. Because not all of them may fit on you system disk you may not wish to install them now. This manual assumes that they are in your execution directory or in memory. The other two directories on the disk (CS2 and CS4) contain character set modules. List the file CharSets to find out more about them.

Potential Problems

If HiRes did not work as explained above, there are a couple of possible reasons why.

- Be sure that you have enough free memory. HiRes requires a little more than 9K. If you do not have enough, either re-boot OS-9 or unlink an unnecessary module.

- Make sure that you ran HiRes exactly as explained above. HiRes needs the file "StdCS", which it looks for in the default execution directory (the CMDS directory unless changed). If it still isn't working, you should try rebooting the system.

You can check to make sure your disk is OK using the OS-9 "ident" command. "ident hires" should indicate that the file HiRes contains the module HiRes, with a good CRC. "ident stdcs" similarly...

2.2.2 How to Undo HiRes

Unlike most commands, HiRes remains in effect after it exits and control is returned to OS-9. You will see that the module HiRes is still in memory after you run it. You will not be able to unlink or save it. The module "StdCS" is also still in memory. You will be able to unlink it, but if you do and you load something else into memory, you will no longer see recognizable characters on the screen. This is because HiRes gets information about how to draw the characters from StdCS.

HiRes may be terminated by sending control codes to it. One way to do this is using the "display" command. Type:

```
display 1f 1f 1f e 12 ENTER
```

to terminate HiRes and return you to the standard display mode. HiRes will automatically unlink itself and StdCS.

The sequence of three 1f codes is the actual command to terminate HiRes. It doesn't get rid of the graphics screen for you - in case you want to do something else with it. That's what the "e 12" does.

Hint: For purposes of experimenting and speed, load the display command into memory with:

```
load display ENTER
```

2.2.3 Pass-through Mode

HiRes has a mode in which it passes all commands directly to CCIO, the display device driver which comes with OS-9. The control code, Hex 1C, enters this mode. Example:

```
display 1c e 12 ENTER
```

to enter pass-through mode, deallocate the screen, and get back 6K of memory.

To end this pass-through mode, send three 1E codes.

```
display 1e 1e 1e ENTER
```

The triple-1e and triple-1f are the only codes recognized by HiRes in pass-through mode.

Please note: Both the 1e and 1f code are recognized by HiRes and CCIO in pass-through mode. For example, the CCIO command to draw a line to graphics location (30,30) is:

```
display 16 1e 1e ENTER
```

If you do that in pass-through mode, those two 1e's are also recognized by HiRes! Therefore, if you do

```
display 16 1e 1e 1e ENTER
```

then two of the 1e's will perform "double-duty" with the result being:

1. A line is drawn to (30,30) and
2. Pass-through mode is terminated.

Clearing the Screen

HiRes recognizes all of the text control codes recognized by CCIO. A very useful one is the clear screen command, Hex 0C.

If a graphics screen is already allocated when you run HiRes, then HiRes does not automatically clear the screen. To clear the screen (while in HiRes), simply type:

```
display c ENTER
```

Potential Problems

If you send 1C to HiRes without sending the E and 12 you may think that you are dead. You can type on the keyboard but nothing appears on the screen. This is because your output is going to the low res screen. The E and 12 are needed to get you back to the low res screen.

2.3.0 Advanced Use of HiRes

Characters and Cells

HiRes draws characters into pre-defined boxes on the screen, called cells. When HiRes is accessed it obtains the height and width, in pixels, of the cells from the character set module. For a visual picture of this, see CSEdit, 3.2.2 section 3.

Also contained in the character set module are the dimensions of the characters. All of the characters in a character set are the same size or smaller than the cell. The character is placed against the right-hand side of the cell. However, it may be placed at any vertical position in the cell, depending on further information contained in the character set module. This capability is used to provide descenders, which are the portions of certain lower-case letters such as "g" and "y" that extend below the baseline.

2.3.1 Changing the Cell Size

To change the width of the cell, use the following display command:

```
display 1b 54 <width>
```

where <width> is the desired width of the cell. The width of the cell must be at least the width of the character, and at most ten.

Note all numeric parameters, such as <width>, are "offset" by 32, or hex 20. By "offset" we mean to 'add' 32 or hex 20 to the <width>. Thus, to specify a cell width of 6 you type:

```
display 1b 54 26 ENTER
```

This convention follows the convention for addressing the text cursor in CCIO, and is used so that the parameters are usually printable characters and they won't get confused with control codes.

To change the cell height type:

```
display 1b 55 <height>
```

where <height> is the desired height of the cell. The height must be at least the height of the character and at most sixteen. It too has an offset of hex 20.

Note that if you make the height too small, the bottoms of some characters with descenders may be chopped off.

HiRes keeps track of the cursor position as an absolute pixel location, not a multiple of the cell dimensions. The cursor location actually points to the upper-right corner of the cell, but it is adjusted when you change the cell width to keep the upper-left corner in the same place. However, if you change the cell height, you may end up with the vertical cursor position in a place that is not an even multiple of the cell height. This causes the characters to appear chopped-in-half at the top of the screen. The easiest way to fix this is to clear the screen (refer to Appendix A in the back of this manual).

2.3.2 Using a Different Character Set

When you first call in HiRes the character set defaults to the Standard Character Set, STDCS. However, you may select a different character set when initializing HiRes. To do this, simply type HiRes followed by the name of the character. As an example:

```
HiRes Roman ENTER
```

where Roman is the name of the character set.

When you are running HiRes you may also switch to a different character set at any time. To accomplish this you must send to HiRes the hex codes 1B 4A, followed by the name of the character set (as a string of ASCII codes) and a carriage return. The easiest way to do this is to use the display and echo commands:

```
display 1b 4a ; echo <character set name>
```

The two commands must appear on the same line, separated by a semicolon as shown, to work properly.

The rules that apply to the OS-9 commands also apply to the character sets. That is, the character set must either be loaded in memory, or in the current execution directory, which upon boot up is the CMDS directory. Therefore, if your character set file is on another disk you must either load the file in memory (load <file name>), or change your execution directory to the directory the character set file is in (chx <directory>). Alternatively, you may give a complete pathname in the echo command. For example:

```
display 1b 4a ; echo /d1/CS2/Roman ENTER
```

indicating the character set Roman in a directory called CS2 located on drive one.

When you load in a new character set, HiRes reads in the new cell size from the character set module. Any previous cell size commands will be ignored. However, foreground and background colors and other parameters will remain in effect.

Character Sets Provided with O-Pak

There are several pre-defined character sets that are provided with O-PAK. These can be found in the directories CS2 and CS4 on the O-PAK disk. The character sets in CS2 are compatible only with two-color mode graphics, the sets in CS4 are compatible with both two-color and four-color mode graphics. The file "CharSets" on the O-PAK disk contains a list of the character sets that are provided, and gives a brief description of each. You may list this file (list charsets) if you want to review this information. It is a good idea to edit CharSets as you create new character sets, so you have an up-to-date record of what the file contains.

Potential Problems

If you attempt to use a non-character-set module for a character set, HiRes will in general refuse to load it. HiRes checks the module type to determine whether or not to link to the module. If you try to use a module which is a data module but is not a character set, unpredictable results will occur. There are no data modules supplied with OS-9.

If you attempt to load in a character set that has the same module name as a module already in memory, one of two things could occur. If the new module has the same or a lower revision number than the existing one, it will not be loaded in, and HiRes will attempt to use the one in memory. If it has a higher revision number, it will be loaded and the existing one will be unloaded.

2.3.3 Terminal Display Features

Foreground and Background

The Foreground color is the color of the characters themselves. The background color is the color behind the characters, or the color of a blank cell.

When HiRes initializes, it obtains the foreground and background colors from the character set module. If graphics memory is already allocated, it uses the graphics mode and colorset already selected, otherwise it obtains the graphics mode and colorset from the character set module.

Two Color Mode

In two-color mode (the default mode) the only colors that are allowed are 0 and 3. 0 is black, the default foreground color, and 3 is white (or buff or green), the default background color. To change the foreground color and the background colors use the following display commands respectively.

```
display 1b 52 <foreground>
          and
display 1b 53 <<background>
```

Both the foreground and background values have an offset of hex 20. Therefore, the codes you can use are (hex) 20, 21, 22 and 23, and only 20 and 23 in two-color mode. You may also set your foreground and background color using one display command. For example, to display white characters on a black background you could use the following command:

display 1b 52 23 ~~53~~ 1b 53 20 ENTER

Note: do not set the foreground and background to the same color, this would make the characters the same color as the background and therefore invisible. You will not be able to read your text.

You may change the graphics mode/colorset using the existing CCIO commands. For example, to switch from buff/black (the default) to green/black use the following command:

```
display f 0 1 ENTER
```

Four Color Mode

HiRes supports the four-color 128 by 192 graphics mode supported by OS-9. In this mode, each horizontal pair of pixels is combined to form one double-wide pixel, which has one of four colors. HiRes copies the character shape directly from the character set module to the graphics memory area, which means that the character set must be specifically designed to work in four-color mode.

Some of character sets provided with O-PAK are compatible with four-color graphics. They are located in the CS4 directory. You already know all of the commands necessary to experiment with them, so we suggest you go ahead. But first, heed the following:

At some time in your experimentation with Hires, you will inevitably find yourself in a situation where unreadable characters are appearing on the screen. The characters will appear to be poorly-formed or fuzzy. This occurrence is due to one of two possible problems, depending on which graphics mode HiRes is in. HiRes may be in two-color mode, but either the foreground or the background color is set to 1 or 2 (instead of 0 or 3); or HiRes may be in four-color mode, but the character set is not a four-color compatible character set. In either case, the following command (which sets the graphics mode and the foreground and background colors) should clear the problem:

```
display f 0 5 1b 52 20 1b 53 23 (ENTER)
```

If it doesn't, you should terminate HiRes and start over.

Scroll Rate

HiRes is endowed with a feature known as smooth scrolling. When HiRes is initialized is has a fast scrolling rate. However, you can change the rate with the display command. This is accomplished as follows:

```
display 1b 58 <rate>
```

where rate is the amount that HiRes moves the screen, in pixels, at a time. For a complete scroll, HiRes always moves the screen a total number of pixels exactly equal to the cell height. To accomplish this, it moves it by <rate> pixels as many times as possible, and then moves it by the remaining amount. The rate has an offset of hex 20, and must be at least 1 (21 with offset), and at most sixteen (30 in hex, with offset). For smooth scrolling type:

```
display 1b 58 21 (ENTER)
```

Block or Underline Cursor

When HiRes is initialized with StdCS, it uses a steady underline cursor. This cursor is defined by the character set module, so it can be any shape as long as it is within the limits of what a character can be. However, to keep things simple we will refer to it as an underline cursor.

You may switch from the underline cursor to a block cursor (which is defined by filling the entire cell) by:

```
display 1b 56 ENTER
```

To return to the underline cursor type:

```
display 1b 76 ENTER
```

Blinking Cursor

You may also select a blinking cursor, and the rate of blink is adjustable. The parameter which determines this actually specifies the time interval, in increments of 1/60 second, during which the cursor stays on or off. The command is:

```
display 1b 57 <interval>
```

The interval has an offset of hex 20 and can have a maximum value of 255, which translates to 1F with the hex offset. Some examples are:

```
display 1b 57 2a ENTER
```

causes the cursor to blink three times per second,

```
display 1b 57 20 ENTER
```

turns off the blink. Note the maximum value for the interval (255) is very long and is not a practical interval.

Graphics

HiRes is completely compatible with the graphics commands supplied with OS-9. This means that you can combine text and graphics on the same screen. It also makes it much easier to experiment with the graphics commands using "display", because you can see what you're typing. Please note that graphics lines and circles are scrolled off the screen just like text is, and text printing interferes with the graphics if they occur in the same place.

Also note that HiRes has a foreground color and background color, and OS-9 (CCIO to be exact) has a foreground color and background color. They are separate entities, and neither one knows or cares what the other one does. The HiRes ones are changed by the HiRes commands and affect the HiRes characters; the CCIO ones are changed by the CCIO commands and affect CCIO lines and circles.

We need to make you aware of a problem with CCIO in this early version of OS-9 - because you are sure to tell us about it if we don't. CCIO should allow you to select two-color graphics, preset the screen to buff or green, and select the foreground color of black (this is the default mode selected by StdCS). However, it doesn't. If you try it, the foreground color (for lines and circles) will stay white. This problem does not occur in HiRes, and is not attributable to HiRes. CCIO will give you the same result even if you're not using HiRes. If you would like to play with graphics in two-color mode, we suggest you go to "inverse video," that is, foreground=3 and background=0.

Windows

HiRes provides the ability to divide your screen up into different windows. A "window" is a section of your display screen that acts as a separate screen. An example of where you might want to use this feature is if you're running a graphics program in Basic09 that draws a picture on your screen. Now you would like to edit the program, but you don't want your editing to scroll the picture off the screen. By putting the picture in a window, it will be protected from any scrolling that may occur.

Windows recognize the following operations: text wrap-around, scrolling, cursor up, right, left (backspace) and down (linefeed). Other commands ignore the window boundaries. For example, the Cursor XY command may be used to move the cursor from one window to another. This command is defined in "OS-9 Commands."

The screen may be divided up into as many as eight windows. The divisions may be in the horizontal direction only.

Windows are created and deleted by creating and deleting boundaries. The screen starts out as one big window, and you split it into two smaller windows by creating a new boundary somewhere in the middle. Two windows are combined into one by deleting the boundary in between them.

The command "display lb 43" creates a window boundary immediately above the line where the cursor is. Note that if you change the cell size, you can end up with window boundaries that do not lie on character cell boundaries; this can lead to some interesting effects. Only eight windows may exist at one time. If you create a new boundary and this results in nine windows, the two windows at the bottom of the screen are combined into one.

The command "display lb 44" causes the window containing the cursor to be combined with the window above it, if any. This command deletes the window boundary above the cursor.

The command "display lb 4e" will remove ALL of the windows.

2.4.0 Programming With HiRes

This chapter is intended for those who wish to write programs that work with HiRes. It contains information on how HiRes works in the OS-9 environment, and on the control codes provided by HiRes for applications programs.

We have included some examples on how to manipulate HiRes from an application program in the 'EXAMPLES' directory on the O-Pak disk.

2.4.1 Philosophy of HiRes

HiRes is not a standard OS-9 program. It is loaded into memory like a standard OS-9 user program, but then it functions like a device driver.

To accomplish this, HiRes breaks a few of the rules of OS-9 programming. It was after careful consideration of many different alternatives for providing the high resolution screen that we decided to write HiRes the way we did.

There is a "correct" way to implement a high-resolution text screen, and that is to write a complete device driver to replace CCIO. The fact is however, that there are not many advantages to doing this other than that it is the "correct" way to do it - and there are several disadvantages. For one, it would not be possible to use the high resolution screen by simply typing "HiRes". Instead, you would have to create a new OS-9 system disk which booted up with the new driver, instead of CCIO. For another, we would need to copy the graphics routines from CCIO to the new driver to maintain compatibility with other programs, such as Basic09. To do this we would need to license them, which might not be possible. Most likely this would force the price of O-PAK to be higher.

Indeed, there are many advantages to the way in which HiRes works. You can run it like any other command, so there is very little "installation" to fuss over; it uses all of the graphics routines you bought with OS-9, so you don't have to give them up and you don't have to buy them twice; and it is reversible - that is, you can remove HiRes from the system easily, returning system memory and restoring the system to exactly its original state.

2.4.2 How HiRes Works

When HiRes is run, the very first step it takes is to modify itself. This is ordinarily prohibited in OS-9 programs, because two users or processes could be using the same program at the same time. However, the program that is run when you type "HiRes" is a built-in installation program, and there is no reason for that program to be run more than once.

HiRes has a built-in safeguard to prevent it from being "double-installed." This built-in "semaphore" is simultaneously set and tested by the installation program. If the test finds that it has already been set, the program exits without proceeding any further. Thus it is impossible for more than one process to proceed to the rest of the installation.

Next HiRes links itself, so it stays in memory after it exits. This is because some of the internal routines will be patched into CCIO, providing the high-res screen. Next it modifies its own module header, making itself non-sharable so you can't unlink it; this would cause the system to crash.

Finally, HiRes modifies the CCIO module to substitute the HiRes routines for the CCIO device output routines. The result is a very smooth and fool-proof transition. Even though HiRes breaks the rules, it does it in a way that prevents any problems from occurring.

2.4.3 HiRes Termination

When HiRes terminates, similar precautions are taken. For example, HiRes unlinks itself when it exits, but the actual code to do the unlink is not in the HiRes module when it is executed. If it were, it might be overwritten, causing a system crash. Instead this code is placed on the system stack before it is executed.

2.4.4 HiRes Control Codes

The following HiRes control codes are useful in applications programming, in addition to those given in 2.2 and 2.3. To avoid confusion, these codes are given in the same format as the others, but without the word "display." Thus the notation

"1b 58 <parameter>"

means an ASCII ESCAPE character, followed by the letter "X" (hex 58), followed by a parameter encoded as a single character.

Erase to End of Line

The sequence "lb 41" causes all cells, beginning with the one containing the cursor and ending at the edge of the screen, to be cleared to background color. Any remaining pixels to the right of the last cell on the line are also cleared.

Erase to End of Window

The sequence "lb 42" has the same effect as "lb 41" but in addition, all pixels below the cursor (but within the same window) are cleared.

Insert Line

The sequence "lb 45" causes a line to be inserted at the cursor position. The line at the bottom of the window is deleted, and the line at the cursor position and those below it, if any, scroll down. The variable scroll rate is effective for downward scrolling.

Delete Line

The sequence "lb 46" causes the line containing the cursor to be deleted. The lines below it and within the window scroll upward, and a blank line is inserted at the bottom of the window.

Shift Right

The sequence "lb 47" causes the line containing the cursor to shift right by eight cells. The eight cells on the left are cleared to background.

Shift Left

The sequence "1b 48" causes the line containing the cursor to shift left by eight cells. The eight cells on the right are cleared to background.

Fine Cursor Right

The sequence "1b 49" causes the cursor to shift right by one pixel, provided the full character cell still fits on the screen.

Fine Cursor Up

The sequence "1b 4b" causes the cursor to shift up by one pixel, provided the full character cell still fits in the window.

Fine Cursor Down

The sequence "1b 4c" causes the cursor to shift down by one pixel, provided the resulting character cell still fits in the window.

Fine Cursor Left

The sequence "1b 4d" causes the cursor to shift left by one pixel, providing the resulting character cell still fits on the screen.

Remove All Windows

The sequence "1b 4e" causes all previously defined windows to be removed.

Overstrike Mode

In overstrike mode, new characters print on top of existing characters or graphics. Pixels that are already foreground color are not cleared to background by character drawing. Overstrike mode is enabled by "1b 51" and disabled by "1b 71".

2.4.5 Additional HiRes Get Status Codes

HiRes adds several additional "Get Status" codes. These are useful for determining the current setting of many HiRes parameters. The assembly language syntax for obtaining these values is:

```
lda #1          (path number)
ldb #code       (as listed below)
os9 I$GetStt
```

Passed: nothing

Returns: parameters in registers, as explained below.

The codes are given in decimal. The parameters returned do NOT have an offset of 32.

Code 128 returns A=foreground color, B=background color.

Code 129 returns A=character width, B=character height.

Code 130 returns A=cell width, B=cell height.

Code 131 returns A=maximum displayable ASCII code.

Code 132 returns A=number of text columns on screen, B=number of rows.

Code 133 returns A=cursor Y position, B=cursor X position. These values are the number of pixels the upper right corner of the text cell is offset from the upper left corner of the screen. In other words, if A=0 and B=the cell width minus one, the cursor is at the home position.

Code 134 returns A=blink period, B=1 if block cursor, B=0 if "underline".

Code 135 returns A=1 if in overstrike mode, 0 if not.

Code 136 returns A=scroll rate (1-16).

Code 137 returns the Y coordinates of the tops of all windows, in A and B and in the MSB and LSB of X, Y, and U. The value 192 (hex C0) corresponds to the line below the bottom of the screen, and indicates a non-existent window. A is always zero, signifying the top of the screen.

3/CHARACTER SET EDITOR

3.1.0 What Is CSEdit?

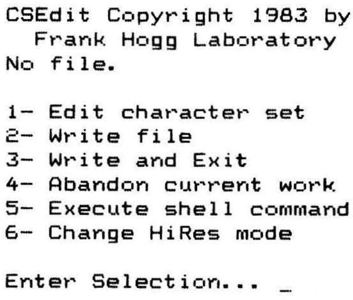
You may use CSEdit to create new character sets from scratch, or to modify existing character sets such as the ones provided with O-PAK. NOTE: CSEdit makes extensive use of the capabilities of HiRes to combine text and graphics on the same screen, therefore, it is necessary to be running HiRes when using CSEdit.

3.2.0 Using CSEdit

CSEdit may be run just like any other OS-9 command. However due to a bug in the early version of OS-9, you should load it first, then run it. If you don't load it first and you get an ERROR #207, it is due to a problem in OS-9 and not in CSEdit.

3.2.1 The Main Menu

When you run CSEdit you are presented with a menu. A menu is a list of possible actions, together with a clue as to how to cause that action to occur. The CSEdit main menu looks like this:



```
CSEdit Copyright 1983 by
  Frank Hogg Laboratory
No file.

1- Edit character set
2- Write file
3- Write and Exit
4- Abandon current work
5- Execute shell command
6- Change HiRes mode

Enter Selection... _
```

The first two lines, which combine the functions of title and copyright notice, form the menu header. Each of the menus in CSEdit has a distinctive-appearing header to help you remember where you are.

The third line, which says "No file", is a "status line." Later on, when you start to edit a character set in a file, this message will change to indicate the file name.

The next six lines are choices of what action to take next. At the bottom of the menu, CSEdit is telling you that it is waiting for you to enter your selection.

In the sections below, we will discuss what each of these choices means. First, let's address one obvious question: Why does this menu only occupy a small portion of the screen? (I'm assuming you are still using StdCS.) That assumption is really the answer - the menu is designed to fit on the screen no matter what character set and cell size you're using. For this reason, it is restricted to using 25 characters per line, and 12 lines.

CSEdit Main Menu Choices

1—Edit character set

"Edit character set" is the obvious choice for editing an old character set or creating a new one. CSEdit will ask you for a file name, which it will read, looking for a character set module. It is very tolerant in reading and will accept bad CRC bytes and numerous other problems. However, if it can't make out a sensible character set or if the file doesn't exist at all, CSEdit will assume that you want to start a new character set and will initialize all of the parameters and proceed. If this happens, you will start off with a module name of "CS," a cell size and character size of 1 pixel by 1 pixel, and one character defined - the cursor.

After reading the file, CSEdit will present you with another menu, the Main Edit Menu. This menu, and its choices, are discussed after we're done with the CSEdit Main Menu.

2—Write File

After you're done Editing the file, you will return to the Main menu. At this point you will probably wish to exercise the "Write File" option to write out the file. CSEdit will reprint the file name for you, and ask you for permission to overwrite this file. Be sure that you want to change this file before you say yes!

After a brief pause, you will be asked to enter two values - the revision number and the edition number. They are contained in the header information of the character set module and they are printed out by the ident command. If, after creating your new file, you wish to print this information use the following syntax:

```
ident filename
```

Most important, the revision number determines which copy of a module is kept if two modules of the same name are loaded in.

After you enter these values, CSEdit will write the file.

Suppose you have done some editing and you wish to save your results, but you decide you don't want to overwrite the file you started with? There is no way to have CSEdit write directly to a different file so we recommend the following procedure: Use the "Shell command" option (see below) to copy the file to a different file, then proceed with writing the file.

3—Write and Exit

This option does the same thing as "Write", and then returns you back to OS-9. Note that if the Status line says "No file", then this option just returns to OS-9, without writing anything.

4—Abandon current work

If you wish to discard the changes you have made (assuming it was just some "playing around") so that you can exit from CSEdit without writing, use this option. You will be asked to verify your intention, after which the status line will return to "No file." Be careful - everything you have in CSEdit will be erased by this action!

5—Execute Shell command

The "Execute Shell Command" option will ask you to enter an OS-9 command, which will be passed to the Shell for processing. You should be familiar with the Shell and the available commands through your OS-9 documentation. With this option you can suspend what you are currently doing, execute your OS-9 command and then return to what you were originally working with. If you enter a blank line, you will be given a shell for processing a series of commands until you enter an EOF (clear-break).

You may use this option to save the contents of a character set file before you write to it. Simply use the "copy" command to copy it to another file.

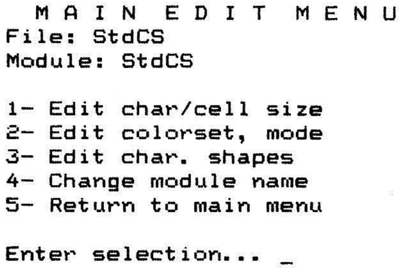
6—Change HiRes mode

This option allows you to change some of the parameters under which HiRes is currently operating. Basically, you can change your character set, your cell dimensions, your colorset and graphics mode. These changes have nothing to do with the character set you are editing (if any). They simply cause CSEdit to send out the codes to HiRes, much as you would do with the "display" command.

You are presented with a sub-menu by this option. The options and choices are self-explanatory, so we won't waste effort detailing them here. You may use this, for example, to select a character set compatible with four-color graphics before you edit a four-color character set.

3.2.2 The Main Edit Menu

The Main Edit Menu looks like this:



```
   M A I N   E D I T   M E N U
File: StdCS
Module: StdCS

1- Edit char/cell size
2- Edit colorset, mode
3- Edit char. shapes
4- Change module name
5- Return to main menu

Enter selection... _
```

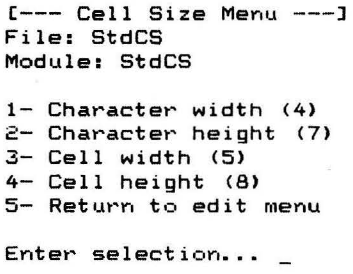
The main edit menu, like some of its sub-menus, has two status lines: one of these gives the filename, the other gives the module name. In this example, both of these names are "StdCS".

The first four choices allow you to change various aspects of the character set module you are editing. We will discuss these choices below. Choice 5 returns you to the "CSEdit Main Menu", where the copyright notice appears, which we discussed above.

1—Edit char/cel size

This choice would be your first action if you were creating a new character set. It is also used to change the size of the characters or the cell in an existing character set.

This menu looks like this:



```
[--- Cell Size Menu ---]
File: StdCS
Module: StdCS

1- Character width (4)
2- Character height (7)
3- Cell width (5)
4- Cell height (8)
5- Return to edit menu

Enter selection... _
```

The numbers in parenthesis to the right of choices 1 through 4 tell you the current value of that parameter. For a new character set, all of these values start off at 1. As you choose one of these, you will be asked to enter a new value. You may not make the cell width greater than 10, or the cell height greater than 16. You are allowed a character width as great as 8 and a character height of 16. Any other values greater than those listed are considered illegal by CSEdit and will not be accepted.

Notice that to make everything larger, you must change the cell size first, then the character size. To make everything smaller, you must shrink the character first. This is because the character must always fit in the cell.

2—Edit colorset, modes

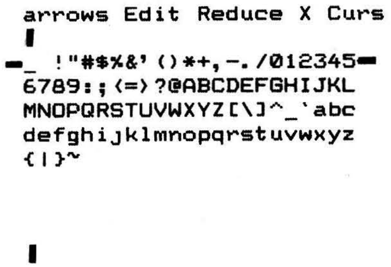
The appearance of the "Edit colorset, modes" menu, and the rules that govern its operation, are similar to that of the "Edit char/cell size" menu. This menu allows you to change the foreground color, the background color, the graphics mode, and the colorset.

One note is in order here: If you change the graphics mode to 1 (for four-color graphics), CSEdit will immediately make some changes to the character shapes. This is to coerce the shape table into compatibility with four-color graphics. Remember that for four-color graphics, the pixels come in pairs of two, both of which must be the same.

3—Edit char shapes

Finally, we get to the core of CSEdit, where all of the fun is to be had. This choice allows you to modify the shapes of the characters in the character set module.

Here you are presented with a menu of a different form. An example would be:



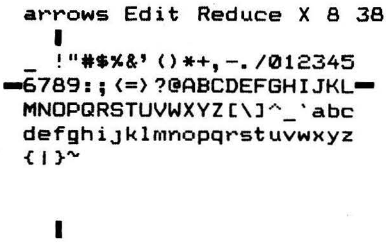
```
arrows Edit Reduce X Curs

_ !"#$%&'()*+,-./012345
6789:;<=>?@ABCDEFGHIJKL
MNOPQRSTUVWXYZ[\]^_`abc
defghijklmnopqrstuvwxyz
{|}~
```

In this display, the top line is not exactly a menu, but rather a reminder of what actions you can take. These, which will all be explained below, are to use the arrow keys, the E key (Edit), the R key (Reduce), and the X key (eXit).

The final item on the top line is a status display. In the example above it says "Curs". It is related to the rest of the display, which consist of an array of characters and a set of crosshair pointers. In the example, the pointers are pointing at the upper-left character, which is the underline cursor, and the status item on the top line indicates the cursor is currently selected.

Arrows

If you were to press the right-arrow twice and the down-arrow once, the display would then look like this:



```
arrows Edit Reduce X 8 38
   ▌
_ !"#$%&'()*+,-./012345
▬6789:;<=>?@ABCDEFGHIJKL▬
MNOPQRSTUVWXYZ[\]^_`abc
defghijklmnopqrstuvwxyz
{|}~


   ▌
```

As you can see, the pointers have moved to point to the character "8", and the status item indicates "8 38", which is the character selected, 8, and its ASCII code in hexadecimal, 38.

As you may have guessed, this is how you select which character you want to work on. For this reason we call this the "select menu." The entire character set being edited is displayed here. The status item shows how the character you're selecting appears in the character set that HiRes is currently running.

Edit

To edit the character you have selected, press the E key. This will lead to another menu, discussed below.

You may also select a character that does not yet exist. Simply move the pointers to a blank section of the array. You can tell what you're selecting by looking at the status item. When you press "E", the character set will be extended to the new character. (All newly created characters are initialized to a single pixel in the upper left corner, so they are visible in the select menu.)

Reduce

The character set may by extended up to ASCII code 255, or hex FF. How do you Reduce the size of it? One character at a time, by pressing the R key. Each time you press the R key, the last character in the character set is deleted. This can be a slow process, so don't extend it more than you really want to.

Exit

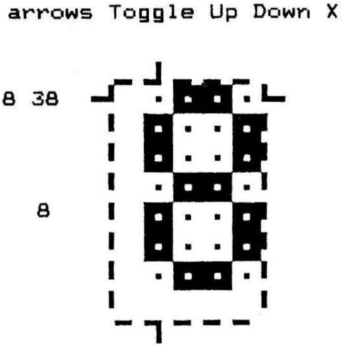
As hinted at above, the X key will return you to the Main Edit Menu.

I J K M or (Arrows)

Finally, for convenience in hopping around the character set, some alternate keys are defined. The I,J,K, and M keys, which form a diamond shape on the keyboard, can be used instead of the arrow keys. The direction each one corresponds to is the direction it is offset from the center of the diamond. Also, the space key may be used instead of the "E" key.

Editing a character

The menu for editing a character is a cousin of the select menu. It looks like this:



Once again, we have a set of crosshairs - now they point at a pixel in the blow-up representation of the character. To the upper left we have the character and its hex code, as it appears in the character set HiRes is running. Also, below that, the character as it appears actual size, as edited. As you make changes to this character, it will change, allowing you to view your work in its proper perspective. And once again, the top line reminds you of your choices.

Arrows, Toggle, Up and Down

The choices are to : move the pointer, using the arrows or IJKM; to Toggle the state of the pixel you are pointing at, using "T" or the space bar; to move the character Up or Down in the cell, using "U" and "D"; and to eXit to the select menu.

If you are editing in four-color mode, the blow-up will show the pixels in double format, and Toggle will affect both of them. This is obvious when you look at it on the screen. Also, note that you can only point at, and toggle, pixels that have a dot in the center. The rest of them are the boundary area, inside the cell but not in the character.

Character Set Module Specifications

A character set module is a standard OS-9 memory module, as described in "OS-9 Technical Information". The module type/language byte is $40, "data module."

At location $0009 in the module (right after the module header) is the offset, or relative address of the actual character set table. Following this (starting at offset $000B) are the following items:

Offset	Contents
$000B	Width of character, in pixels
$000C	Height of character, in pixels
$000D	Size of one row of character set table, in bytes
$000E	ASCII code of maximum character in table
$000F	Unused byte
$0010	Width of cell, in pixels
$0011	Height of cell, in pixels
$0012	Default graphics mode (0/1)
$0013	Default color set ($00-$0F)
$0014	Default foreground color (0-3)
$0015	Default background color (0-3)

The character set table is divided into rows, there being as many rows as the cell height plus one more. Each row has the number of bytes specified by the value at offset $000D.

The information in each row is packed without regard to byte boundaries; it is best to think of a row as being a sequence of bits, starting with the most significant bit of the first byte and ending with the last significant bit of the last byte. This sequence of bits is divided into segments of length equal to the width of the character (offset $000B). One segment corresponds to each character. The first segment corresponds to the "underline" cursor, the second segment corresponds to the space character (ASCII code $20), and so on to the last segment, which corresponds to the character having the ASCII code equal to the value at offset $000E. Note that the last few bits of the row may be unused.

The first row contains what we refer to as the descender counts. That is, each segment of the first row contains a binary number which tells how far from the top of the cell the character should be placed. The second row contains the bit pattern giving the shape of the first row of the actual character, and so on.

3.2.3 Final Note about CSEdit

If you have read through this documentation without running CSEdit, you may feel that you don't understand everything. CSEdit is meant to be experimented with, and will become much clearer as you run it. We suggest you don't spend a lot of time doing important work on a character set until you have had a chance to experiment with CSEdit to become a bit more familiar with it.

4/THE 'X' COMMANDS

4.1.0 What is an "X" Command?

The "X" commands XCOPY, XDIR, XDUMP, and XLIST are utility commands that allow you to manipulate Disk Extended Color BASIC (DECB or RS) and FHL FLEX files from Color Computer OS-9. These commands are similar to the corresponding commands provided with OS-9, except that the pathname has been extended to designate an RS or FLEX file.

4.2.0 General Information

File System (DOS) specifier

The disk operating system (DOS) to be accessed is specified by a file system specifier <dos>. The following are valid specifiers:

rs% or decb% - a Disk Extended Color BASIC file.

flex% - an FHL Flex text file, with space compression. *

flexb% - an FHL Flex binary file, without space compression. *

If no file system specifier is indicated, the specifier defaults to the Color Computer OS-9 operating system.

Specify Your Device

The device specifier <dev> is used to designate the drive to be accessed. The specifier is defined the same way a drive is specified by its corresponding DOS. For example:

flex% or flexb%	- <drive>.
rs or decb%	- <drive>:
coco OS-9	- /<device>/

(*) For information on FLEX space compression, refer to the FLEX Advanced Programmer's Guide page 25 (Byte 59) and page 46 ($09 ASCII HT or HORIZONTAL TAB).

File Name

The file name <file> indicates the file to be accessed. The syntax of the file name must be the same as the syntax used by its corresponding DOS. Therefore, the syntax for each DOS is:

```
flex% or flexb% - <name>.<ext> Ex: NAME.TXT
rs% or decb% - <name>/<ext>    Ex: NAME/TXT
coco OS-9 - <name>             Ex: name
```

Complete File Specification

The complete file specifier <filespec> consists of the file system specifier, device specifier and the file name that are to be accessed. The specifier has the following syntax:

```
<dos><dev><file>
```

Please note that, for the purpose of finding a file which is to be read from a disk, upper and lower case letters are considered to be equivalent.

Examples

Examples of valid file specifiers are:

```
/d1/DEFS/os9defs (standard CoCo OS-9 file)
flex%0.DATA.TXT  (FHL Flex text file)
flexb%1.PROG.BIN (FHL Flex binary file)
rs%0:PROG/BAS    (Disk BASIC program)
rs%1:PROG/BAS    (Disk BASIC program)
```

4.3.0 Command Syntax

4.3.1 Xcopy

```
XCOPY <filespec> <filespec>
```

Copies the data of the first file to the second. Xcopy will copy in any direction between Disk Extended Color BASIC format disks, FHL FLEX format disks and CoCo OS-9 format disks. The first file must already exist. An error will be reported if the second file already exists.

Note that copying a file does not translate the file in any way, it just means that byte for byte information in the file is copied from one disk format to another. Therefore, XCOPY is only useful for copying ASCII (text) files, such as source files, BASIC ASCII files etc. For example, you may copy an OS-9 text file to a FLEX file so that you may edit the file with an editor that runs under FLEX. Or you may want to copy a RS BASIC ASCII file to OS-9 and complete the necessary modifications so it will run as a BASIC09 program. Although XCOPY will copy binary files, the results will be useless unless you have a particular reason for doing so. It is beyond the scope of XCOPY to convert a binary file. The task is very complex and we do not have any plans of ever doing such a program.

Note: Unpredictable results may occur if you attempt to run xcopy concurrently with itself.

4.3.2 Xdir

```
XDIR <dos><dev>
```

Displays a directory of the files on the disk indicated. Xdir can only be used with FLEX or RS files; i.e. for OS-9 files the OS-9 DIR command must be used.

4.3.3 Xdump

XDUMP <filespec>

List out a hex/ASCII display of the contents in the file.

FLEX% files are considered to be FLEXB files, in other words space compression is not expanded.

4.3.4 Xlist

XLIST <filespec>

Xlist lists the file to standard output (typically the screen).

Examples

```
xdir flex%1 ENTER
xdump rs%1:ARMENU/BAS ENTER
xlist flex%1.STARTUP.TXT ENTER
xcopy rs%1:ARMENU/BAS flex%2.ARMENU.BAS ENTER
xcopy rs%1:ARMENU/BAS /d0/source/armenu ENTER
```

The first example will do a directory of a FLEX disk in drive 1. The second example will dump the file "ARMENU/BAS" on a Radio Shack DOS disk in drive 1. (Useful to determine if it is an ASCII file) The third will list the FLEX file in drive 1 by the name of "STARTUP.TXT". The fourth example will copy the file in the Radio Shack DOS disk in drive 1 called "ARMENU/BAS" to the FLEX disk in drive 2 with the name of "ARMENU.BAS". The last example will copy the file in the Radio Shack disk in drive 1 by the name of "ARMENU/BAS" to the OS-9 disk in drive 0 into the directory 'source' under the file name of "armenu".

Appendix A—Control Codes

Condensed Summary of OS-9 Display Control Codes:

Hex	Dec			
$01	01		Home alpha cursor	
$02	02	x,y	Position alpha cursor	x=Col+$20 y=Row+$20
$03	03		Erase line	
$06	06		Cursor right	
$08	08		Cursor left	
$09	09		Cursor up	
$0a	10		Cursor down	
$0c	12		Clear screen	
$0d	13		Carriage return	
$0e	14		Select alpha mode	
$0f	15	a,b	Select graphics mode	a=Mode, b=Color Code
$10	16	a	Preset screen	a=Color Code
$11	17	a	Select Color	a=Color Code
$12	18		Deallocate graphics screen	
$13	19		Erase screen	
$14	20		Home graphics cursor	
$15	21	x,y	Move graphics cursor	X Coord Y Coord
$16	22	x,y	Draw line to X,Y	X Coord Y Coord
$17	23	x,y	Erase line to X,Y	X Coord Y Coord
$18	24	x,y	Set point at X,Y	X Coord Y Coord
$19	25	x,y	Clear point at X,Y	X Coord Y Coord
$1a	26	r	Draw circle	r=Radius

HiRes Display Control Codes

Hex	Decimal	Function
$1b $41	27 65	Erase to end of line
$1b $42	27 66	Erase to end of window
$1b $43	27 67	Create window boundary
$1b $44	27 68	Combine window w/1 above
$1b $45	27 69	Insert line
$1b $46	27 70	Delete line
$1b $47	27 71	Shift right
$1b $48	27 72	Shift left
$1b $49	27 73	Fine cursor right
$1b $4a n $0d	27 74 n 13	Change char set <n>ame
$1b $4b	27 75	Fine cursor up
$1b $4c	27 76	Fine cursor down
$1b $4d	27 77	Fine cursor left
$1b $4e	27 78	Remove all windows
$1b $51	27 81	Enable overstrike mode
$1b $52 f	27 82 f	Change <f>oreground color
$1b $53 b	27 83 b	Change <b>ackground color
$1b $54 w	27 84 w	Change <w>idth of cell
$1b $55 h	27 85 h	Change <h>eight of cell
$1b $56	27 86	Block cursor
$1b $57 r	27 87 r	Set blinking cursor <r>ate
$1b $58 r	27 88 r	Change scroll <r>ate
$1b $71	27 113	Disable overstrike mode
$1b $76	27 118	Underline cursor
$1c	28	Enter pass-through
$1e $1e $1e	30 30 30	Terminate pass-through
$1f $1f $1f	31 31 31	Terminate HiRes

Appendix B—Glossary

Background Color	Color behind the characters or the color of a blank cell.
CCIO	Color Computer OS-9's input/output module.
Character Set Editor	The O-PAK editor that allows a user to create and edit character sets.
CharSets	An O-PAK file which contains a listing of available character set modules.
CRC	Cyclic Redundancy Count (see page 19 of OS-9 Technical Information Manual).
CSEdit	See Character Set Editor.
Descenders	Portions of certain lower-case letters that extend below the base line.
Fine Cursor Down	Movement of the cursor down by one pixel.
Fine Cursor Left	Movement of the cursor left by one pixel.
Fine Cursor Right	Movement of the cursor right by one pixel.
Fine Cursor Up	Movement of the cursor up by one pixel.
Foreground Color	Color of the characters themselves in High Resolution Graphics Mode.
Four-Color Mode	Refer to page 127 of the OS-9 Commands Manual.
Graphics Mode	The mode that allows a user to display, control and generate graphics characters (see page 127 of the OS-9 Commands Manual).
High Resolution	The Color Computer graphics mode which allows the display of two color (256 by 192) or four color (128 by 192) graphics.
HiRes	An O-PAK utility which provides a high-resolution screen display for Color Computer OS-9.
Low Resolution	The standard Color Computer screen display of 32x16 characters.
Offset	The difference of Hex 20 between the desired value and the actual value, for control code parameters; and the difference between the actual address and the address of the beginning of the module, for bytes within a memory module.

Pixel	A single graphics point on a high-resolution graphics screen.
Smooth Scrolling	The act of displayed text being moved up on the screen by a consistent and adjustable number of raster lines.
Status Line	A protected line of text which contains information.
StdCS	An O-PAK file which contains HiRes' default standard character set.
tmode -upc	An OS-9 command that allows a OS-9 to display lower-case letters.
Two-Color Mode	The default high-resolution screen color mode in which black and white (or buff or green) characters are generated.
Utility	A program, (usually machine language) that performs a specific function.
Window	A protected section on the display screen which acts, for certain functions, as a separate display screen in and of itself.
"X" Commands	The four utility commands that allow a user to manipulate RS and FHL FLEX files from Color Computer OS-9.
xcopy	O-PAK utility which copies data from one file to another.
xdir	O-PAK utility which displays the directory of a specified disk.
xdump	O-PAK utility which lists out a Hex/ASCII display of the contents in the file indicated.
xlist	O-PAK utility used to list a file to the designate output device.

Appendix C—Index

Arrow keys 30
Background 12
Background Color 27
Backup 3
CCIO 7
Cell Height 27
Cell Size 10
Cell Width 27
Cells 9
Character Set Specs 31
Character set table 31
Character Sets 11
Characters 9
Clear Screen 8
Complete File Specifier 34
CS2 Directory 13
CS4 Directory 13
CSEdit 23
CSEdit Abandon work 25
CSEdit Main Edit Menu 26
CSEdit Write and Exit 25
CSEdit Write File 24
Cursor, Blinking 15
Cursor, Block 15
Cursor, Underline 15
Delete Line 20
descender counts 32
Device Specifier 33
Dual-Drive Systems 5
Edit char/cell size 27
Edit character set 24
Edit character shapes 28
Edit Colorset, modes 27
Editing a character 30
Edition number 24
Erase to End of Line 20
Erase to End of Window 20
Exit Main Edit Menu 30
File Name 34
File Specifier 33
Fine Cursor Down 21
Fine Cursor Left 21
Fine Cursor Right 21
Fine Cursor Up 21
Foreground 12
Foreground Color 27
Format 3
Four Color Mode 13
Four-Color Graphics 27
General Information 5
Get Status Codes 22
Graphics 15
Graphics mode 27
Graphics, Two Color 16
HiRes - Problems 6
HiRes Control Codes 20
HiRes Philosophy 18

HiRes Programming 18
HiRes Terminate 7
HiRes Termination 19
HiRes, How it Works 19
HiRes, install 5
I J K M 30
Ident 6
info 3
Insert Line 20
Letters, lower-case 6
Letters, upper-case 6
Overstrike Mode 21
Pass-through Mode 7
Pass-through problem 8
Pass-through terminate 7
Pre-defined Sets 12
Reduce character set 29
Remove All Windows 21
Revision number 24
Scroll Rate 14
Select Menu 29
Shift Left 21
Shift Right 20
Single-Drive Startup 5
Stdcs, install 5
Tmode -upc 6
Two Color Mode 13
Two-Color Graphics 27
Window Commands 16
Windows 16
Windows, Creating 17
Windows, Deleting 17
Windows, Number of 16
Write-protect 2
X Commands 33
xcopy 35
xdir 35
xdump 36
xlist 36

Notes

FHL FRANK HOGG LABORATORY

THE REGENCY TOWER • SUITE 215
770 JAMES ST. • SYRACUSE, NY 13203
PHONE (315) 474-7856 • TELEX 646740

FHL O-PAK

770 JAMES ST. SYRACUSE NY 13203 315-474-7856