BRISTOL COMMUNITY COLLEGE
CIT-155 Intro to Computer Forensics

WinHex Tutorial

The Specialist version of WinHex is the baby brother of the X-Ways Forensics software package. WinHex is not only an extraordinarily powerful hex editor, but Specialist has several powerful tools to help in the low-level analysis of media and file systems. This low-level analysis is important for the understanding of both the logical structures of a file system and the limitation of manual methods for computer forensics examinations -- particularly as disk drives and other storage media grow larger and larger.

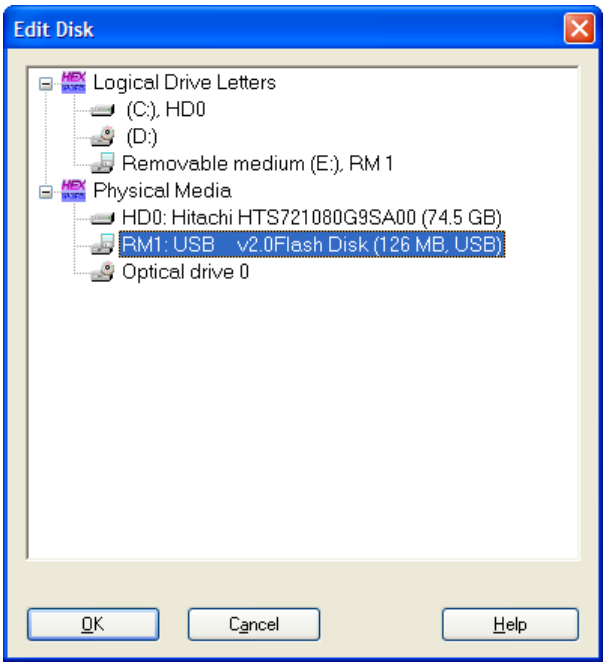
This tutorial will cover some of the essential functions of WinHex Specialist that will be useful in our course. These functions include:

- Sanitizing media and verifying the initialization
- Restoring a disk image
- Opening an image file as a disk
- Logical structure templates
- The data interpreter
- Gather free space
- Copying blocks and creating files
- Exporting a file list

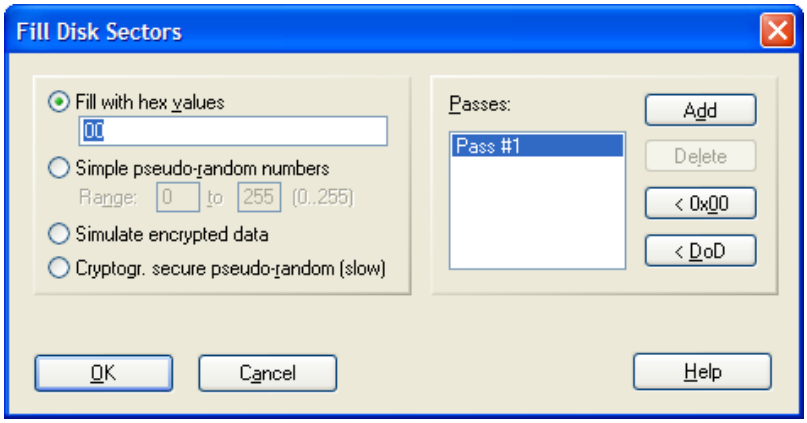
Sanitizing media and verifying the initialization

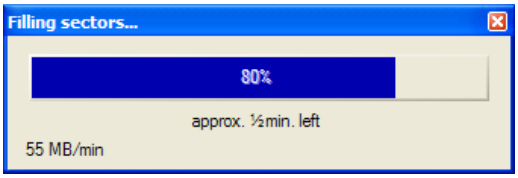
WinHex can be used to wipe target media to prepare it prior to copying an image. The software can also be used to verify that the wiping has been successful. This example describes how to overwrite the contents of a USB thumb drive with all zeroes:

1. Insert the target USB flash media into a USB port.
2. Start WinHex.
3. Open the USB drive using the Tools, Open Disk... command (or press the F9 key). Be sure to select the USB drive as a *physical* device.



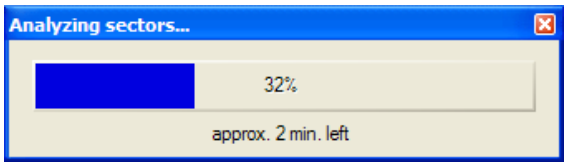
4. Use the Edit, Fill Disk Sectors... command (or press control-L[1]) to zero-fill all sectors on the USB drive.



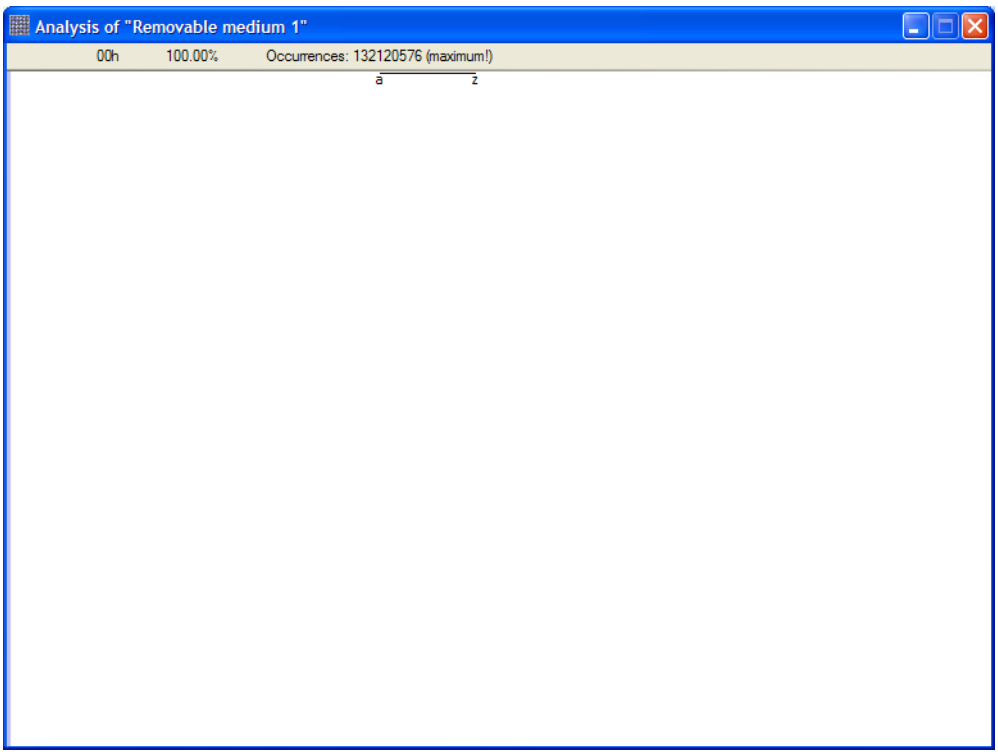


[1] The "control" character may also be denoted by the "^" character, so that "^L" should be interpreted at control-L.

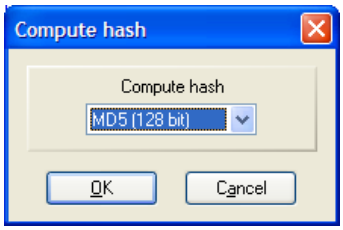
5. Visually verify that the drive has been zero-filled. The most effective way to do this is with the Tools, Analyze Disk command (F2).

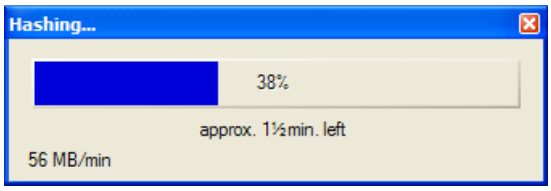


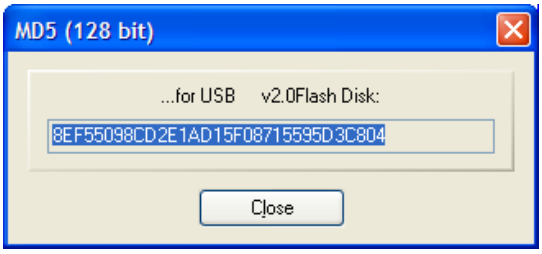
A window will open that appears to be blank; what it shows is the relative occurrence of the 256 bit patterns from 0x00-0xFF. If you move your mouse all the way to the left of the screen until you see the display indicate *00h*, you should see the middle figure indicate *100%* and the "Occurrences" value indicates *(maximum!)*.



6. You can calculate the hash value of a medium using the Tools, Compute Hash... command (^F2). You can select from a number of hash algorithms, but MD5 and SHA-1 are the most common for digital forensics applications.





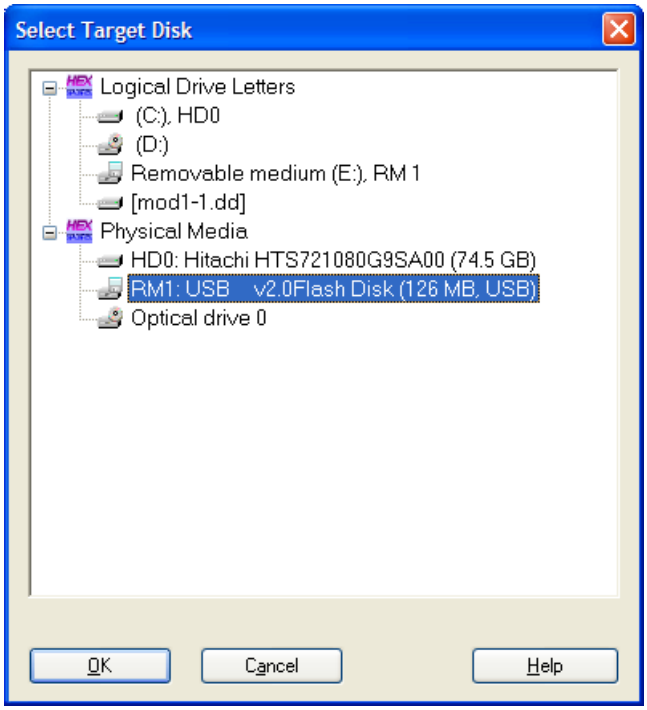


Note that WinHex can also be used to securely wipe individual files. For this function, see Tools, File Tools, Wipe Securely...

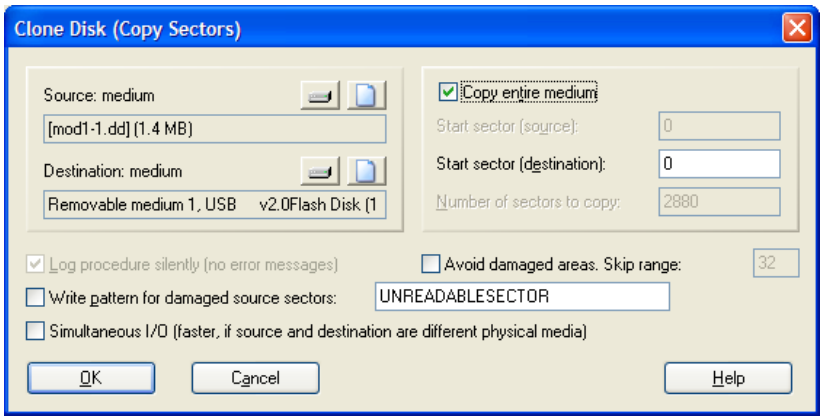
Restoring a disk image

A disk image can be restored using the File, Restore Image command.

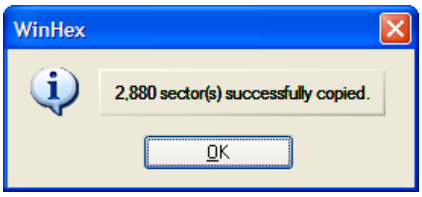
1. Select the image file, which can be in .001, .ctr, .dd, .img, or .whx (WinHex Backup) format.



2. Select the appropriate output medium.



3. Confirm that the source is the image file and the destination medium is correct, and click OK.

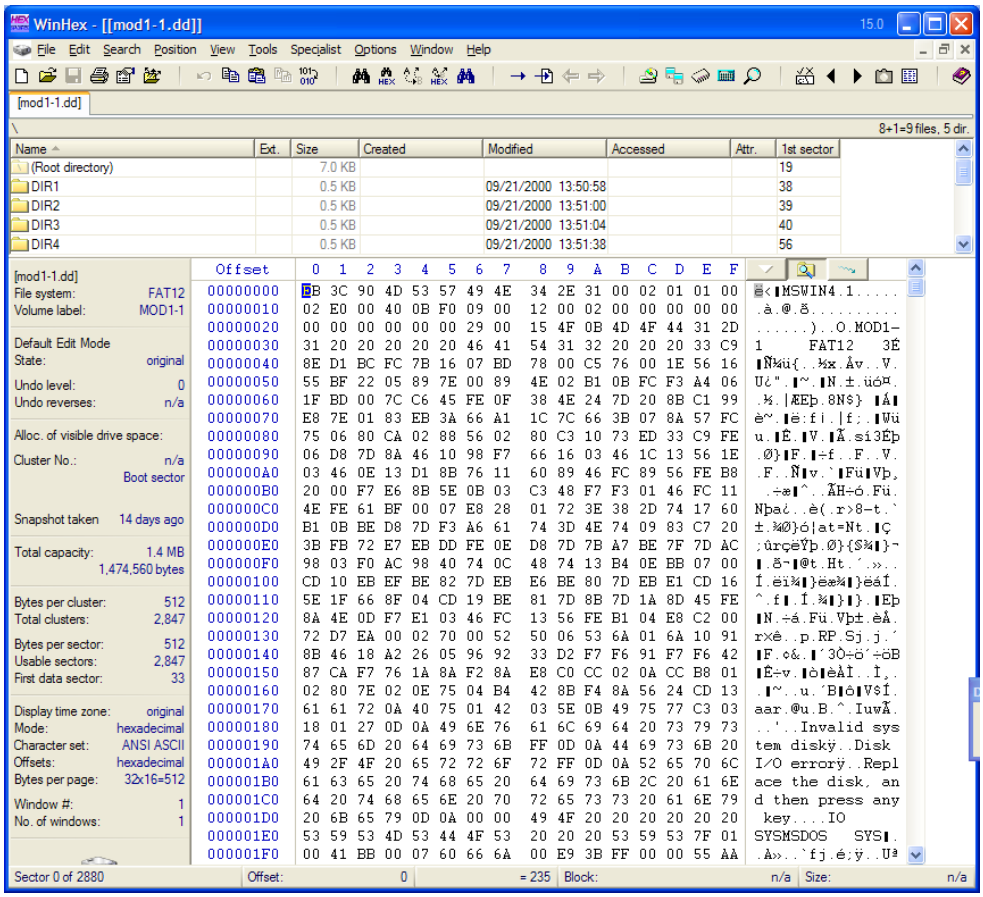


You can open a disk using the Tools, Open Disk command (F9).

Opening an image file as a disk

An image file can be opened directly and used as if it were from a disk.

1. Open the image file using the File, Open command (^O).
2. Use the Specialist, Interpret Image File As Disk command to have the file open as if it were the original disk



3. Note in the display above the presence of the directory browser, which are the lines that show the directories and files. Clicking on a directory or file will move you to that location. If the directory browser is not displayed, use the View, Show, Directory Browser command (^F7) or click on the  icon.

Usage hints:

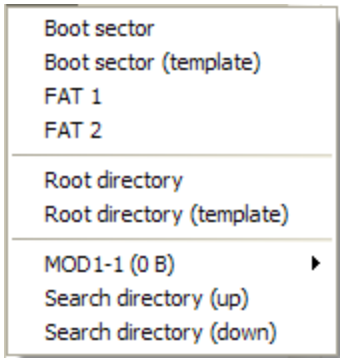
a. *Note in the display above that the width of the screen is set to show 16 columns, numbers 0-F. This is the best setting to ensure a view of 16 bytes per line.*
b. *Note also that 32 lines are shown here (offset 0x0000-0x01F0). You can display any number of lines that you would like but 32 lines is 512 bytes, or exactly 1 sector (see the sector counter at the lower left of the screen). With this number of lines, the display goes to the top of a new sector every time you hit Page Down.*
c. *Note that the values listed in the offset column are in hexadecimal. If you click anywhere in the column, the numbers to change to decimal. Keeping the values in hex is probably simpler. As an aside, 0x200 = 512, which is the size of a sector.*

Logical structure templates

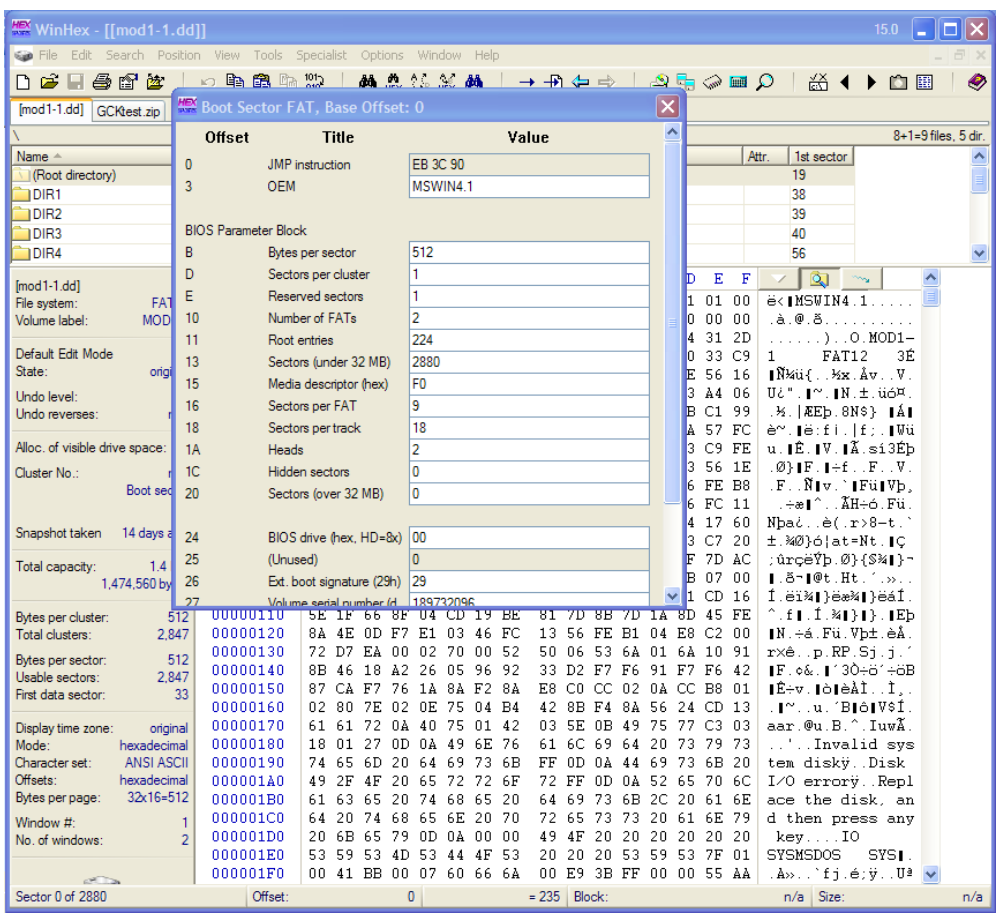
WinHex provides a number of templates that make the interpretation of logical structures much simpler for the user. Indeed, while the user can always see -- and change -- the raw hex values,

the templates make interpretation that much simpler. For our course purposes, the most useful templates are those for the master boot record (MBR) and directories.

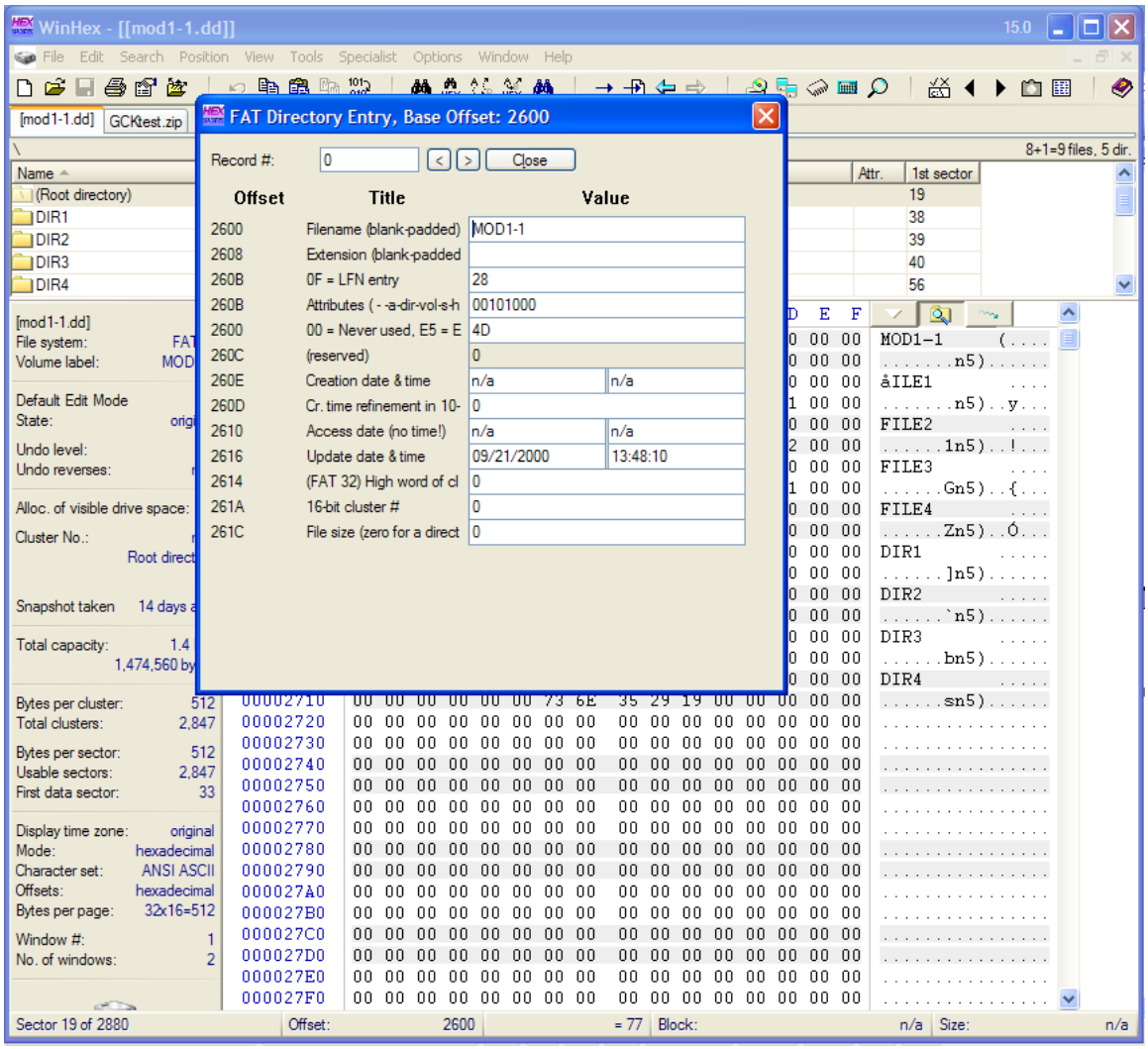
The icon can be used to expose a pulldown menu such as:



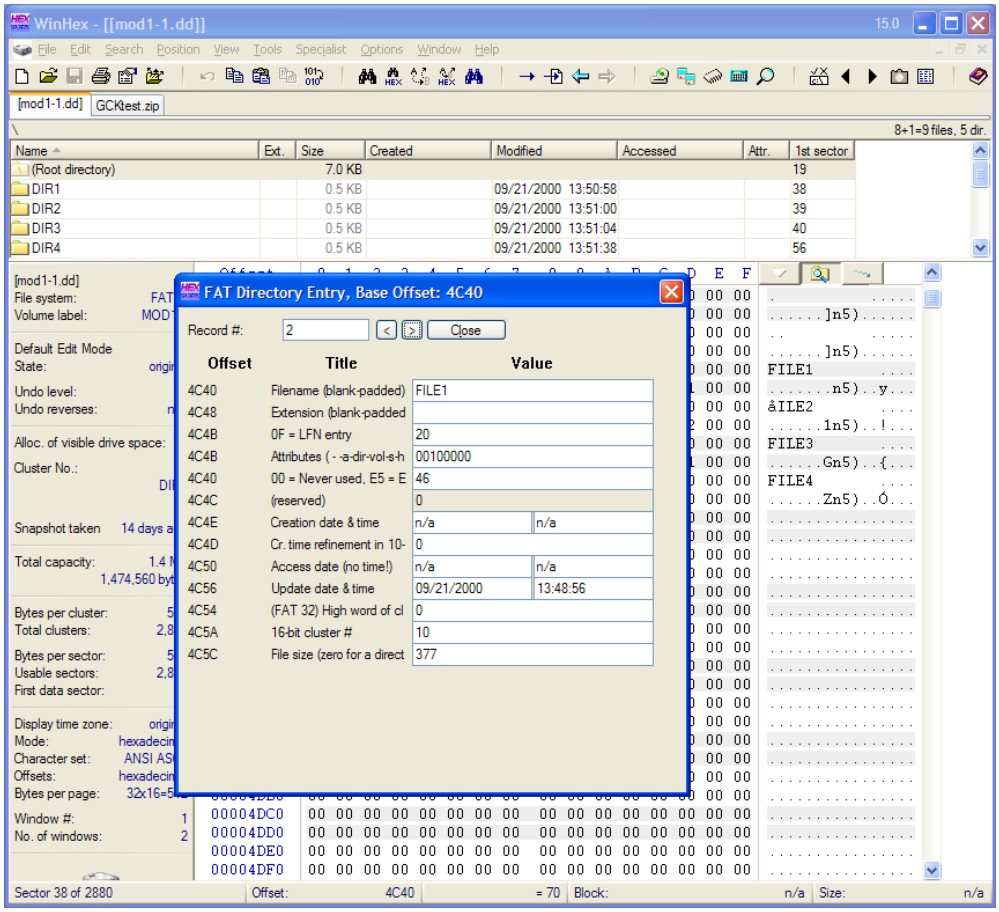
Clicking on *Boot sector* will move you to the boot sector portion of the image, whereas clicking on *Boot sector (template)* will show you the boot sector and provide an interpretation of all of the fields.



Similarly, clicking on *FAT1*, *FAT2*, and *Root directory* will move the display to those portions of the image. Clicking on *Root directory (template)* moves to the root directory and provides an interpretation of the field of each directory entry; click on the left and right arrows to go back or forward within the directory.

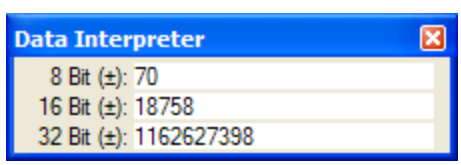


The  pulldown menu also allows you to search for subdirectories, although there is no obvious way to see the directory template. If you want the directory template when perusing a subdirectory, use the View, Template Manager... command (alt-F12), select FAT Directory Entry, and click Apply!

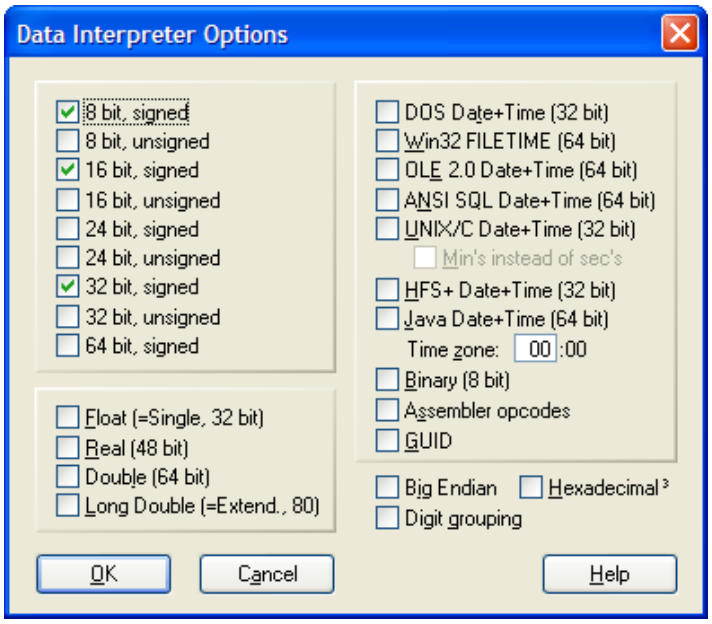


The data interpreter

The data interpreter can be an invaluable aid when trying to determine the value of a field, particularly when the field spans more than one byte.



The Data Interpreter window, as shown above, indicates the 8-, 16-, and 32-bit value starting from the byte location where the cursor is currently highlighted.

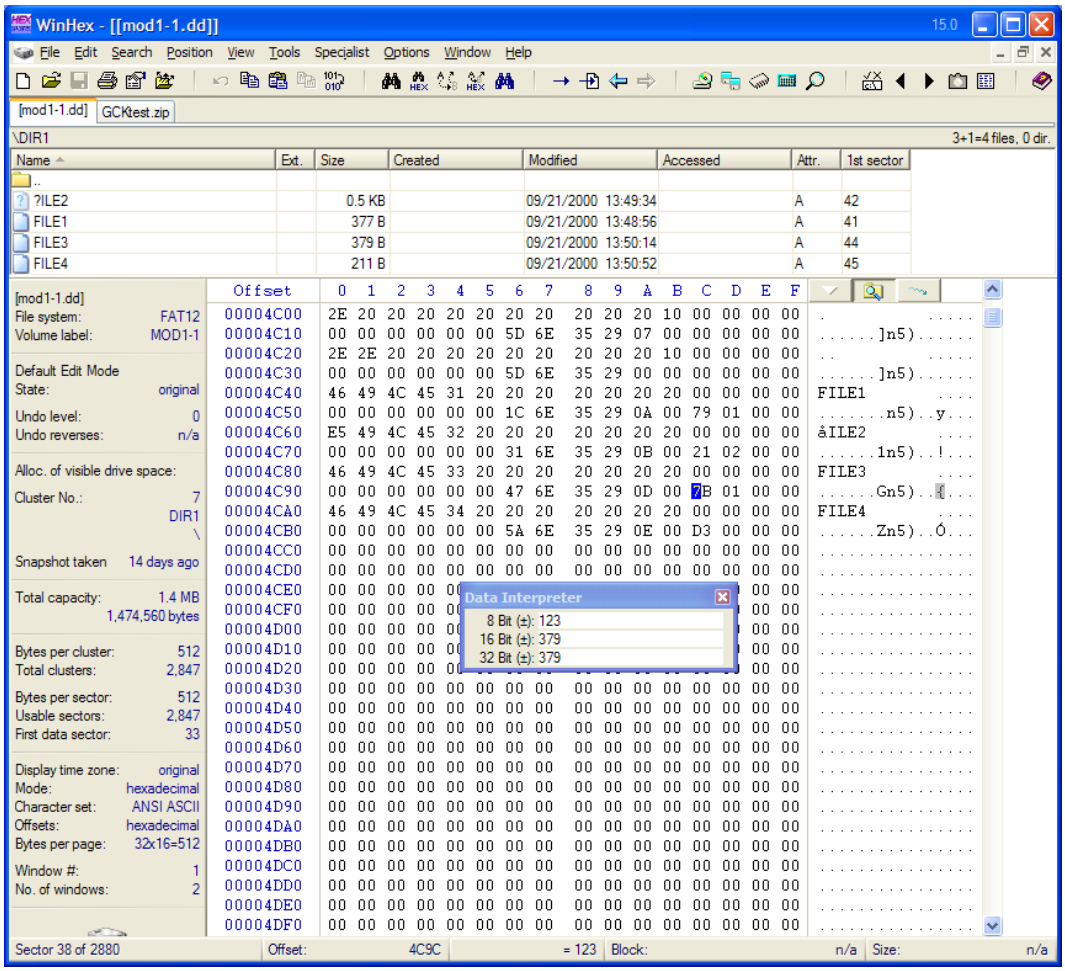


The values shown in the Data Interpreter can be configured using the Options, Data Interpreter... command (alt-F5). As shown above, the Data Interpreter is currently displaying 8-, 16-. and 32-bit unsigned values starting at the current cursor position; this is the WinHex default although many other interpretations can also be shown.

The default byte ordering is Little Endian, meaning the first byte in the field is the arithmetically least significant byte (LSB) and the last byte in the field is the arithmetically most significant byte (MSB). By way of example, suppose a three-byte had the following hex values:

```
00 1C 22
```

With a Little Endian interpretation, most common on Intel processors, these three bytes represent the number 0x221C00 (decimal 2,235,392). With a Big Endian interpretation, these bytes represent the number 0x001C22 (decimal 7,202).



An example in the use of the data interpreter is shown in the screen shot above. In this example, the cursor is highlighting byte offset 0x4C9C. This screen shows entries in a subdirectory listing; the specific byte that is highlighted is byte 28in the directory entry for a file named FILE3; byte offsets 28-31 in the directory entry represent the file length.

The Data Interpreter shows the following:

- The 8-bit value (i.e., byte offset 28 alone or 0x7B) is 123
- The 16-bit value (i.e., byte offsets 28-29 or 0x017B) is 379
- The 32-bit value (i.e., byte offsets 28-31 or 0x0000017B) is 379

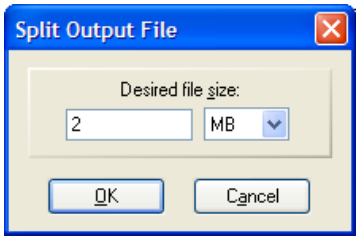
Since the file length field is a 32-bit field starting at byte offset 28, we want the 32-bit interpretation. By the way, this can be verified by looking at the directory listing above that clearly shows FILE3's length to be 379.

Finally, if you close the Data Interpreter mini-window and want to reopen it later, use the View, Show command and check the Data Interpreter box.

Gather free space

One essential function when doing the analysis of a disk drive is to examine the information in unallocated space to find remnants of deleted files. In the WinHex vernacular, *unallocated space* is called *free space*; for our purposes, the two terms are synonymous.

Gathering free space in WinHex is very easy. Once an image or disk is opened, merely use the Specialist, Gather Free Space... command. When the Split Output Window dialogue box appears, choose some value that makes sense for your storage capabilities; any value between 100 KB and 4 MB is probably ok.



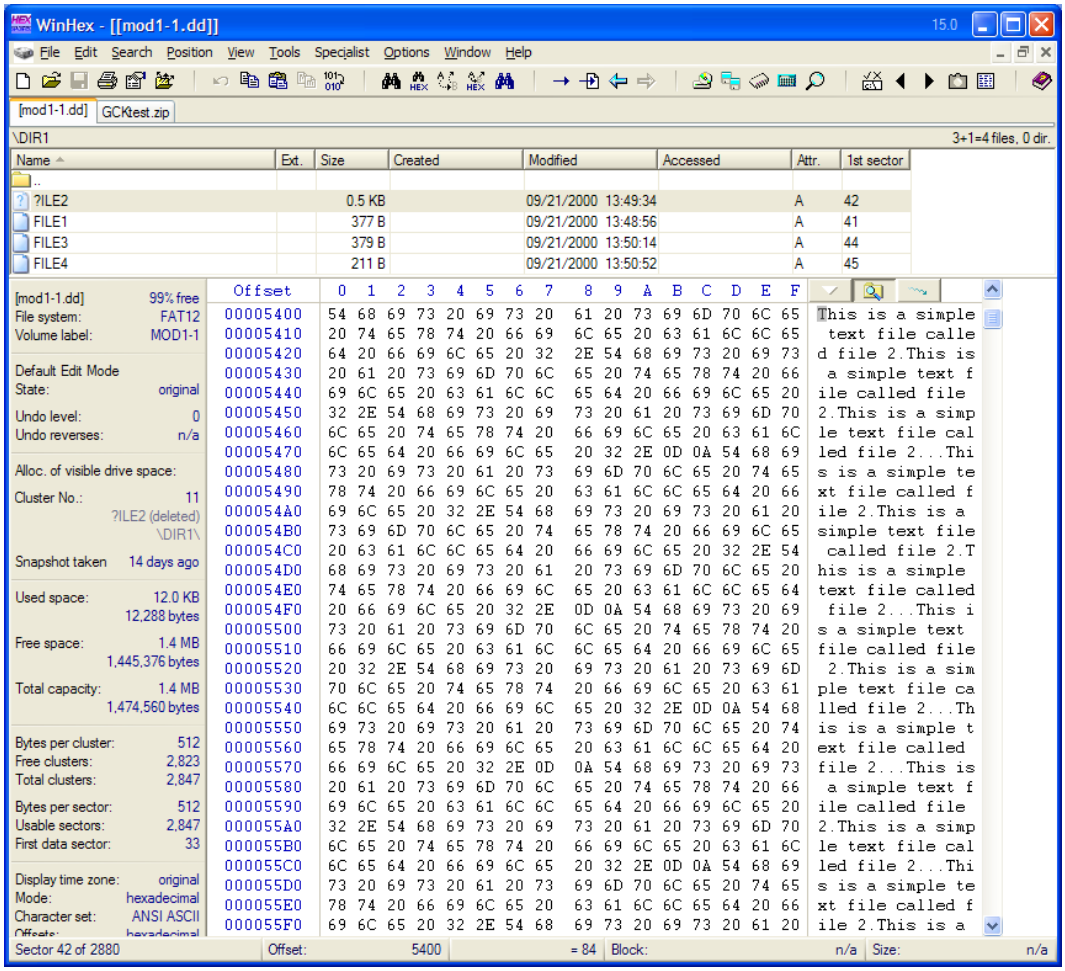
Gathering file slack space is equally easy; use the Specialist, Gather Slack Space... command.

Note: When specifying the output file name, it is best to place the file in a directory of its own. The reason is that many disk carving utilities, such as Simple Carver Lite, use the contents of an entire directory as the input rather than individual files. Placing the Free Space (and Slack Space) files in their own directory provides some protection against accidentally including files from some other source other than the image.

Copying blocks and creating files

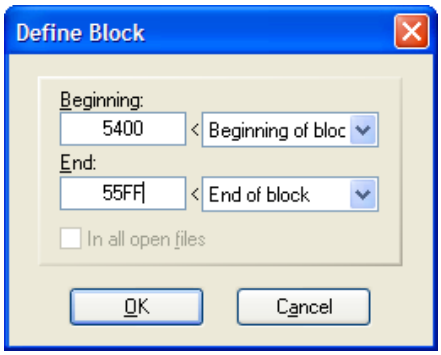
The ability to export user-defined blocks of data as files is an important one in WinHex. This section will describe how to define a block of data, export a single block of data as a file, and export multiple blocks of data into a single file.

Define block

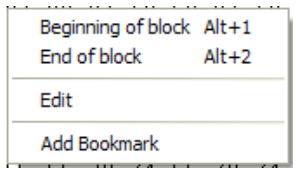


Defining a block of data is relatively straight-forward. The screen shot above shows sector 42, which is byte offsets 0x5400-55FF. Note that the cursor is highlighting the first byte in the sector. There are at least three ways to define this block:

a. Note the starting and ending byte offsets. Use the Edit, Define Block... command, enter the beginning and end offset numbers, and click OK.

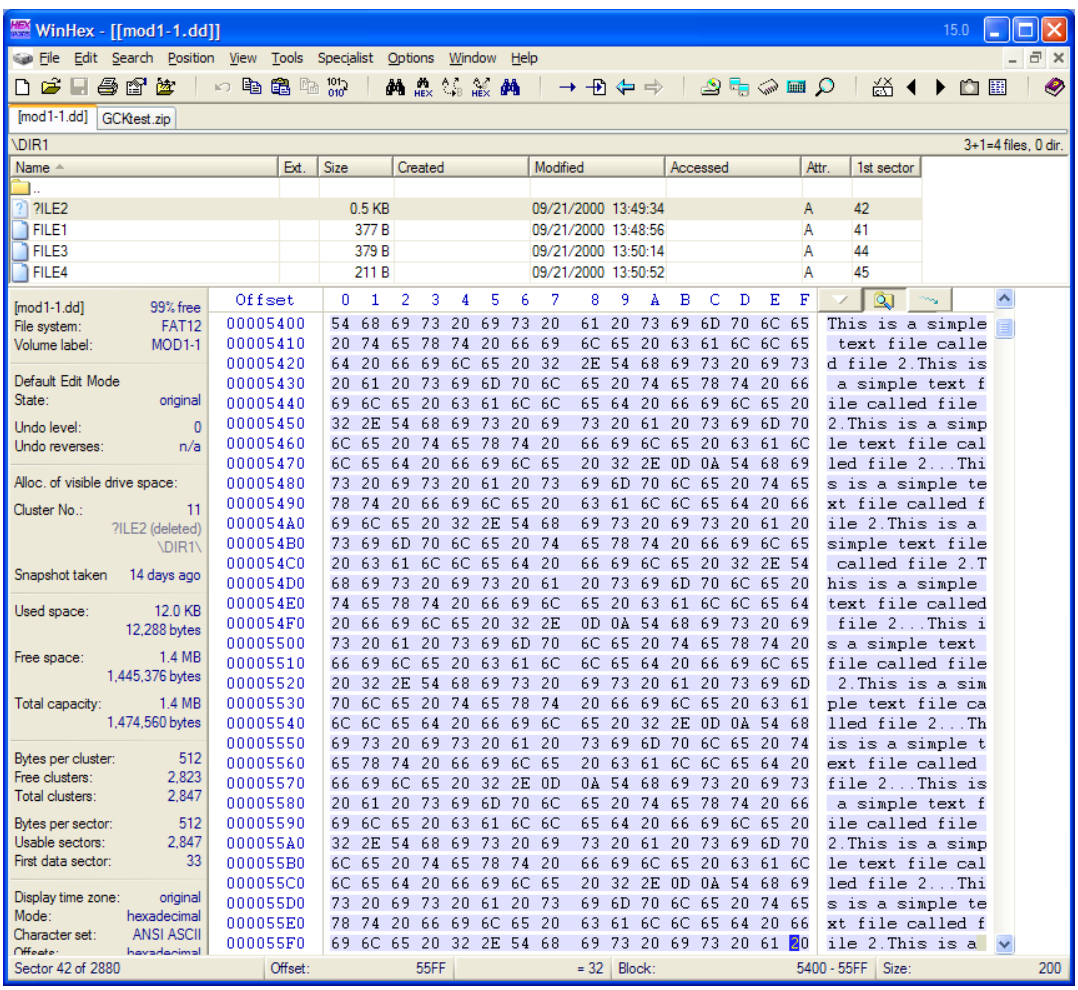


b. Position the cursor over the first byte in the block; then, either press alt-1 *or* right-click and select Beginning of block. Now, position the cursor over the last byte in the block, then either press alt-2 *or* right-click and select End of block.



c. Select the block as you would text in a document; place the cursor over the first byte of the block, hold down the SHIFT key, drag the cursor to the last byte in the block, and then release the SHIFT key.

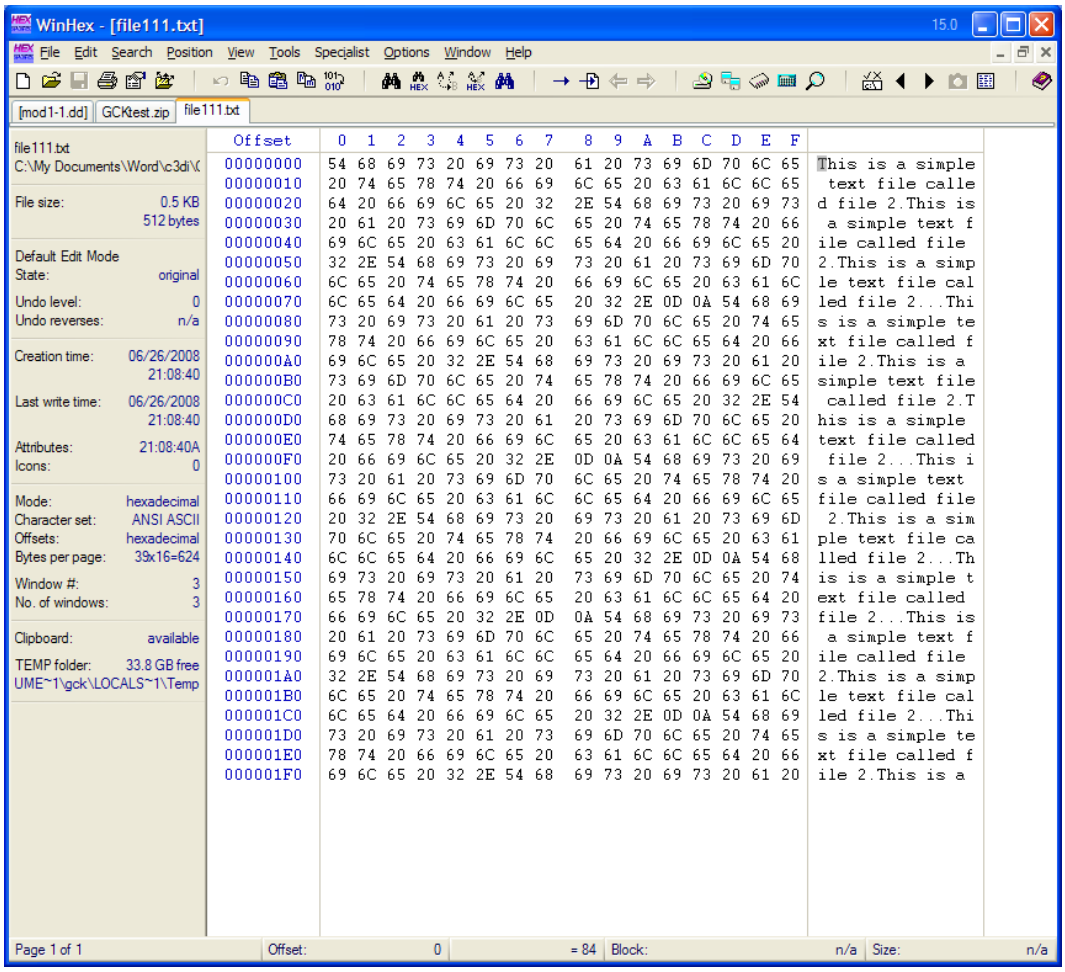
Regardless of the method, the block will display as highlighted, per the screen shot below.



Note: Press <ESC> to clear the block highlighting.

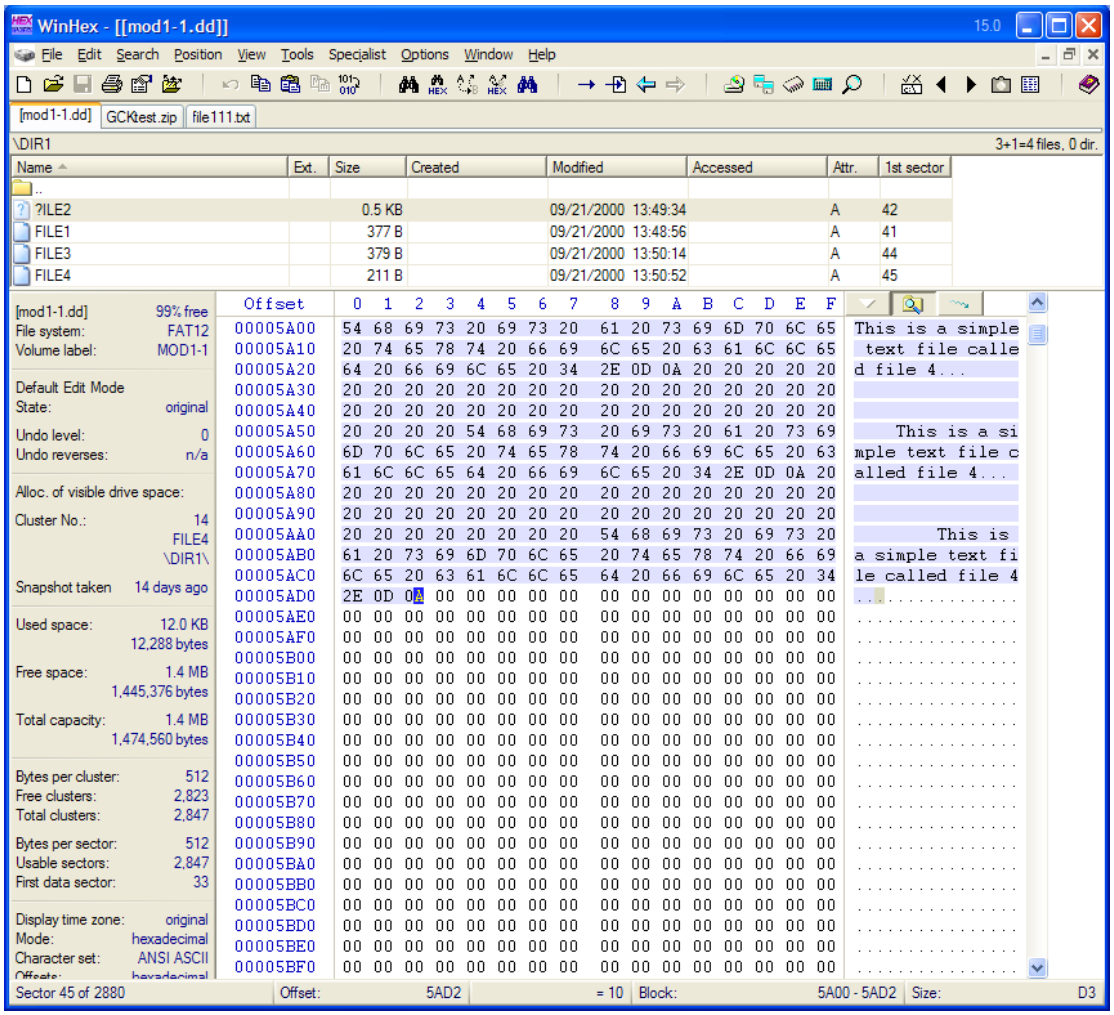
Copy block to a file

Copying the block to a file is also relatively straight-forward. Use the Edit, Copy Block, Into New File command (control-shift-N) and provide a file name. The new file will open in WinHex.



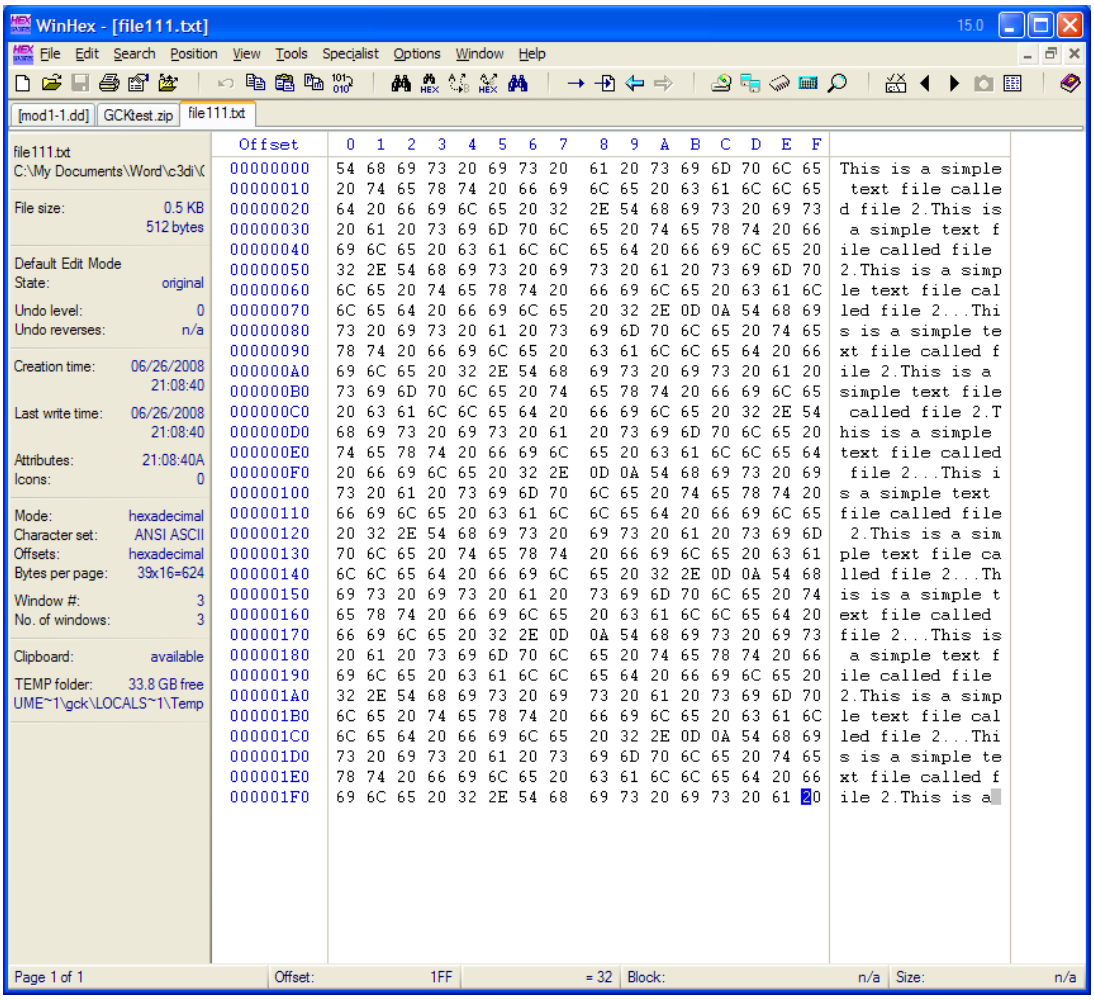
Export multiple blocks of data into a single file

Exporting multiple blocks of data into a single file is relatively simple but is a multi-step process.

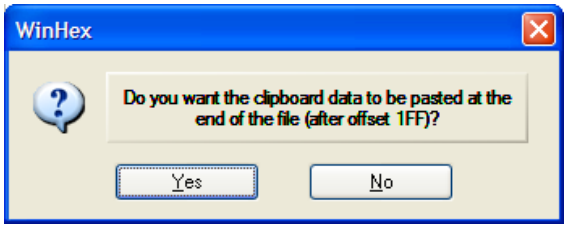


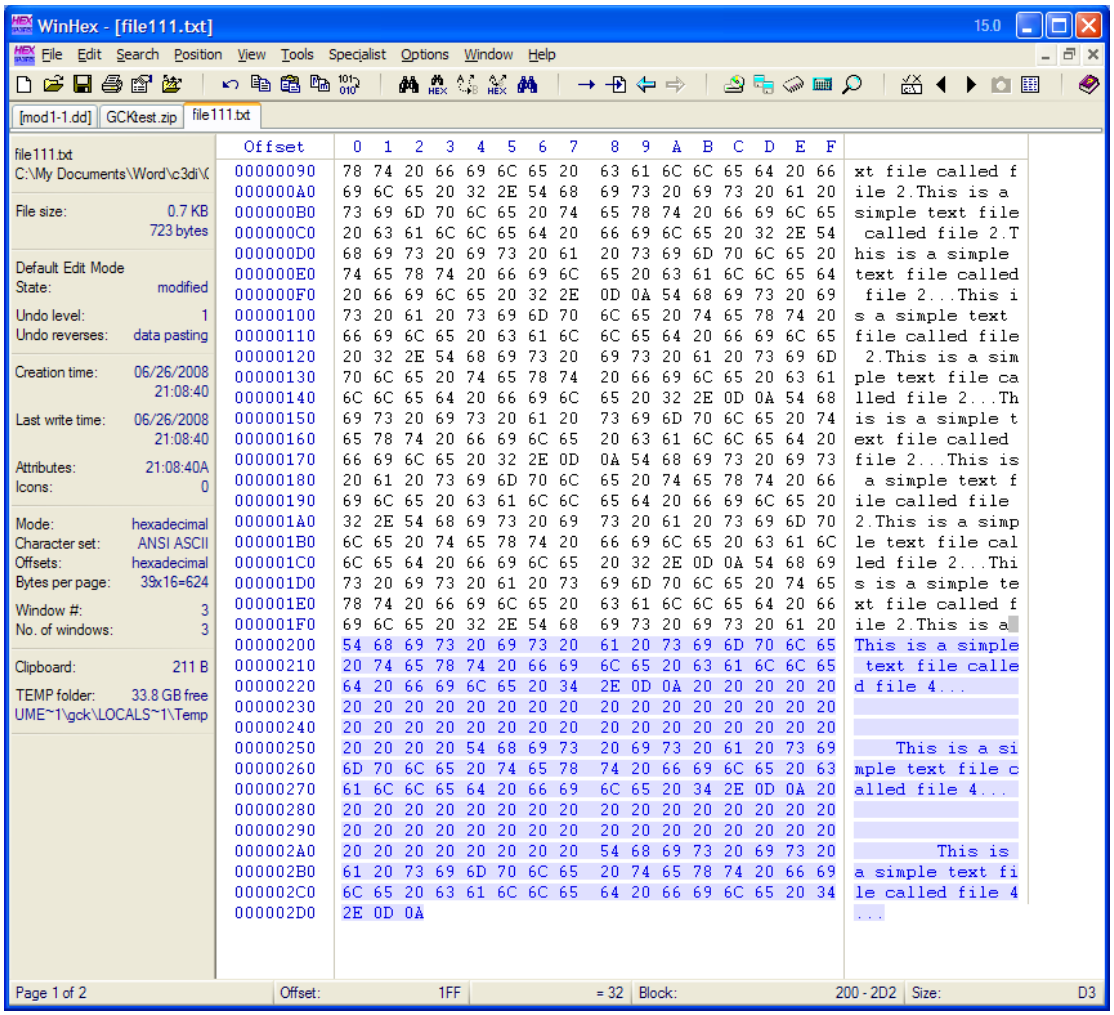
The first step is to define and export the first block into a file as described above. Suppose we now wanted to append bytes 0x5A00-5AD2 to the file created in the last step. Start by defining the new block using one of the three methods described above.

Next, copy the block into the clipboard using the Edit, Copy Block, Normally command (^C).



Finally, click on the tab corresponding to the file where the data is to be added, click on the last byte of the file, and then use the Edit, Clipboard Data, Paste command (^V). Since we want the data at the end of the block, click YES when the dialogue box above appears.





Additional blocks can be added by following these same steps. Be sure to save the updated file; data added to the file that has *not* yet been saved will be displayed in **blue**.

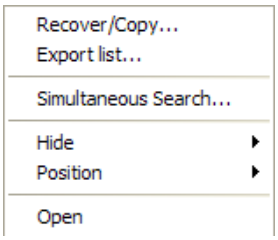
Note: This same method will work to paste the second block anywhere in the file.

Exporting a file list

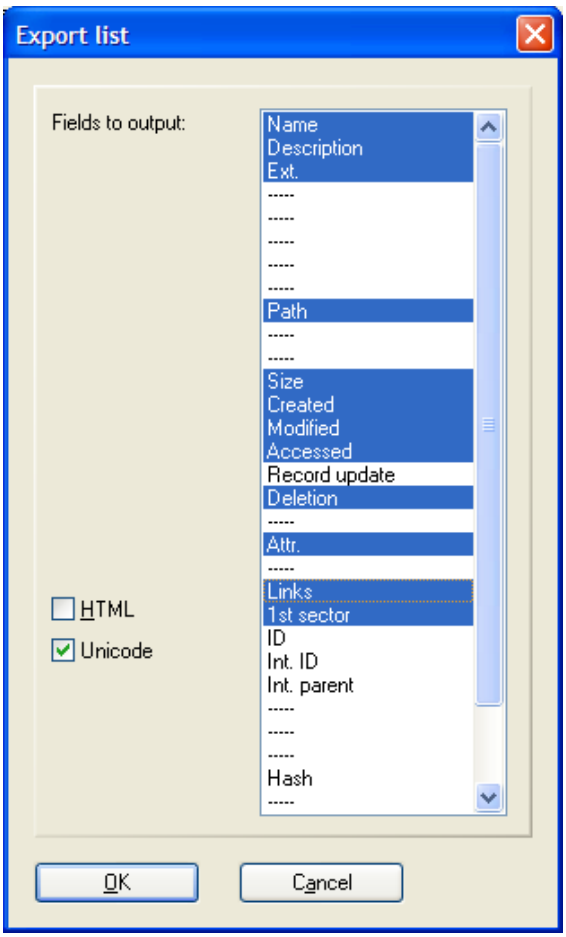
The directory browser feature of WinHex is a great way to navigate around the medium or image file. Copying the directory browser information provides a nice way to build a list of all of the files found in the image. The only bad part is that WinHex only allows you to grab one directory at a time, namely the one that is displayed; thus, if you want to get multiple directories, it is a multi-step process.

Suppose you want to get a complete directory listing of an image with to create an Excel spreadsheet listing all files and their attributes. The following steps will accomplish this:

Navigate to the root directory and highlight the entire directory browser section. Right-click and select Export list...



When the dialogue box opens, be sure to select Unicode (you may have to first unselect HTML). Use this opportunity to highlight the fields that you want to save.



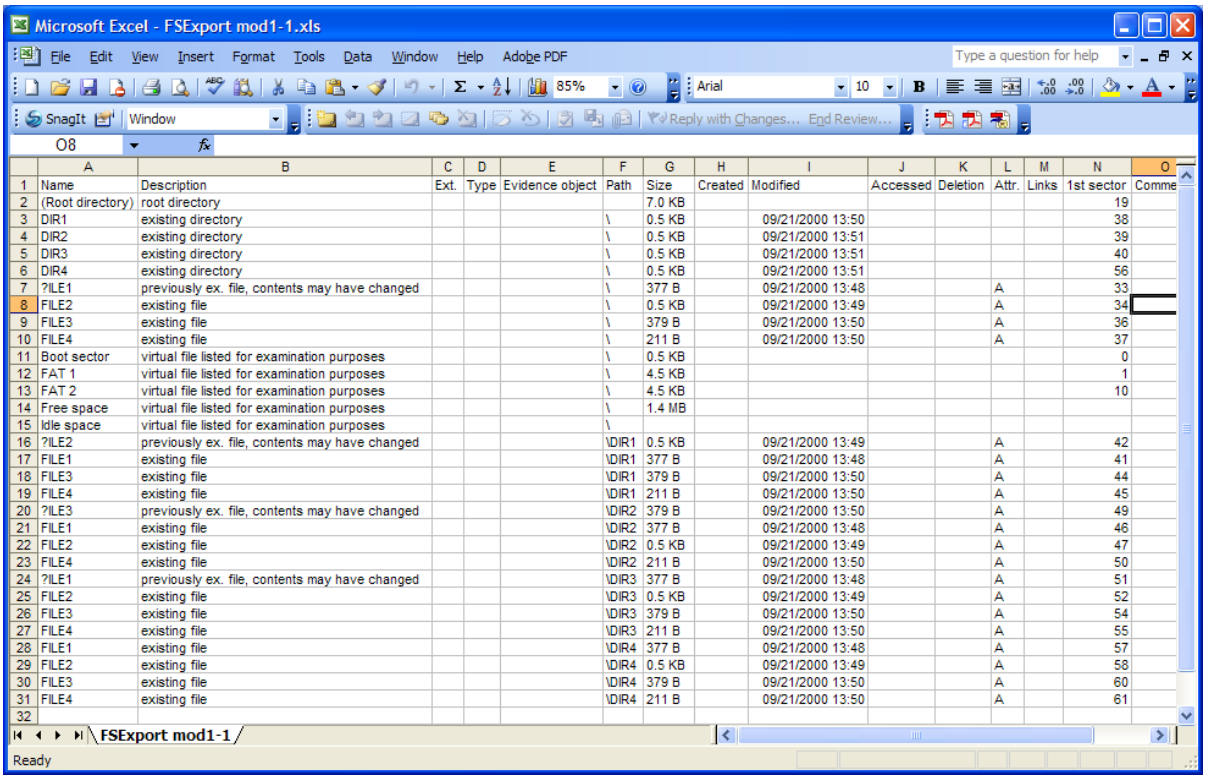
After clicking OK, you need to provide a file name.

Continue navigating to the subdirectories on the medium and repeat these steps. **Be sure that you select unique file names when you save subsequent files.**

To create the file list in an Excel spreadsheet, do the following:

Open the first export file from Excel. Since it is a text file, you will see the Text Import Wizard dialog box; click on FINISH to open the file.

Open the second file just as you did the first file. Copy the contents of the second file and paste into the first. Repeat for the remaining files; open, copy the contents, and paste into the first file. After this somewhat tedious process, you will have a spreadsheet with the complete file list. Add additional comments, as desired.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Name	Description	Ext.	Type	Evidence object	Path	Size	Created	Modified	Accessed	Deletion	Attr.	Links	1st sector	Comme
2	(Root directory)	root directory					7.0 KB							19	
3	DIR1	existing directory				\	0.5 KB		09/21/2000 13:50					38	
4	DIR2	existing directory				\	0.5 KB		09/21/2000 13:51					39	
5	DIR3	existing directory				\	0.5 KB		09/21/2000 13:51					40	
6	DIR4	existing directory				\	0.5 KB		09/21/2000 13:51					56	
7	?ILE1	previously ex. file, contents may have changed				\	377 B		09/21/2000 13:48			A		33	
8	FILE2	existing file				\	0.5 KB		09/21/2000 13:49			A		34	
9	FILE3	existing file				\	379 B		09/21/2000 13:50			A		36	
10	FILE4	existing file				\	211 B		09/21/2000 13:50			A		37	
11	Boot sector	virtual file listed for examination purposes				\	0.5 KB							0	
12	FAT 1	virtual file listed for examination purposes				\	4.5 KB							1	
13	FAT 2	virtual file listed for examination purposes				\	4.5 KB							10	
14	Free space	virtual file listed for examination purposes				\	1.4 MB								
15	Idle space	virtual file listed for examination purposes				\									
16	?ILE2	previously ex. file, contents may have changed				\DIR1	0.5 KB		09/21/2000 13:49			A		42	
17	FILE1	existing file				\DIR1	377 B		09/21/2000 13:48			A		41	
18	FILE3	existing file				\DIR1	379 B		09/21/2000 13:50			A		44	
19	FILE4	existing file				\DIR1	211 B		09/21/2000 13:50			A		45	
20	?ILE3	previously ex. file, contents may have changed				\DIR2	379 B		09/21/2000 13:50			A		49	
21	FILE1	existing file				\DIR2	377 B		09/21/2000 13:48			A		46	
22	FILE2	existing file				\DIR2	0.5 KB		09/21/2000 13:49			A		47	
23	FILE4	existing file				\DIR2	211 B		09/21/2000 13:50			A		50	
24	?ILE1	previously ex. file, contents may have changed				\DIR3	377 B		09/21/2000 13:48			A		51	
25	FILE2	existing file				\DIR3	0.5 KB		09/21/2000 13:49			A		52	
26	FILE3	existing file				\DIR3	379 B		09/21/2000 13:50			A		54	
27	FILE4	existing file				\DIR3	211 B		09/21/2000 13:50			A		55	
28	FILE1	existing file				\DIR4	377 B		09/21/2000 13:48			A		57	
29	FILE2	existing file				\DIR4	0.5 KB		09/21/2000 13:49			A		58	
30	FILE3	existing file				\DIR4	379 B		09/21/2000 13:50			A		60	
31	FILE4	existing file				\DIR4	211 B		09/21/2000 13:50			A		61	
32															