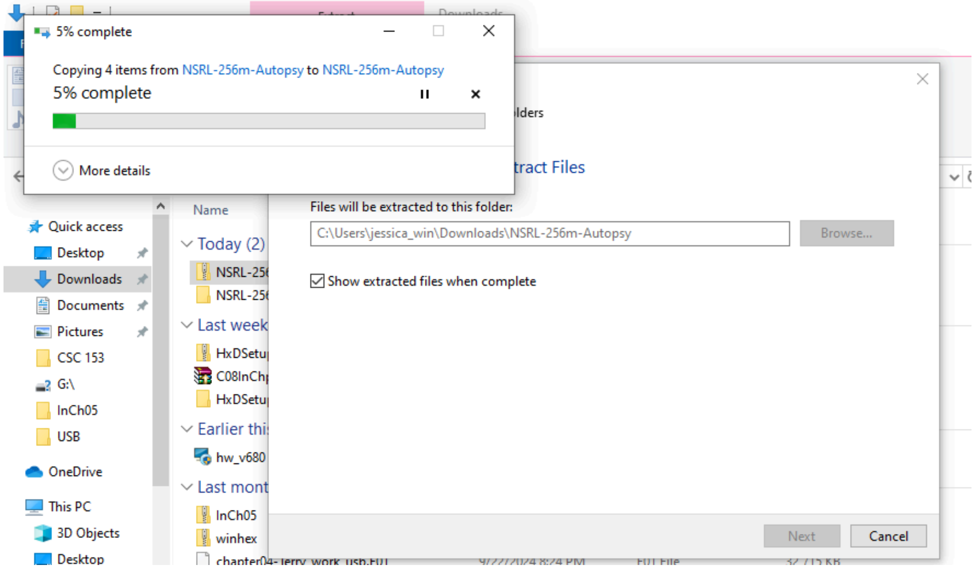
Task 1: Autopsy Hashsets

Step 1: Setup NSRL in Autopsy



5% complete

Copying 4 items from NSRL-256m-Autopsy to NSRL-256m-Autopsy

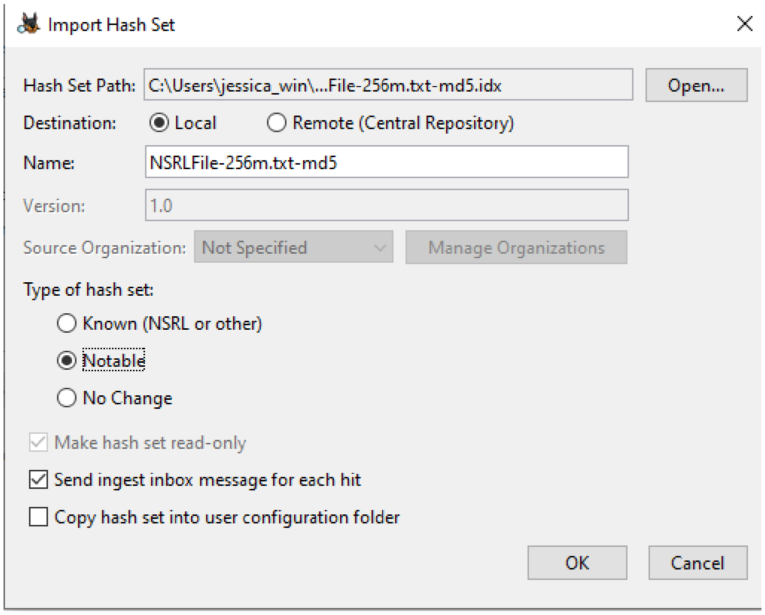
5% complete

More details

Files will be extracted to this folder:

C:\Users\jessica_win\Downloads\NSRL-256m-Autopsy

☑ Show extracted files when complete



Import Hash Set

Hash Set Path: C:\Users\jessica_win\...File-256m.txt-md5.idx

Destination: ● Local ○ Remote (Central Repository)

Name: NSRLFile-256m.txt-md5

Version: 1.0

Source Organization: Not Specified

Type of hash set:

○ Known (NSRL or other)

● Notable

○ No Change

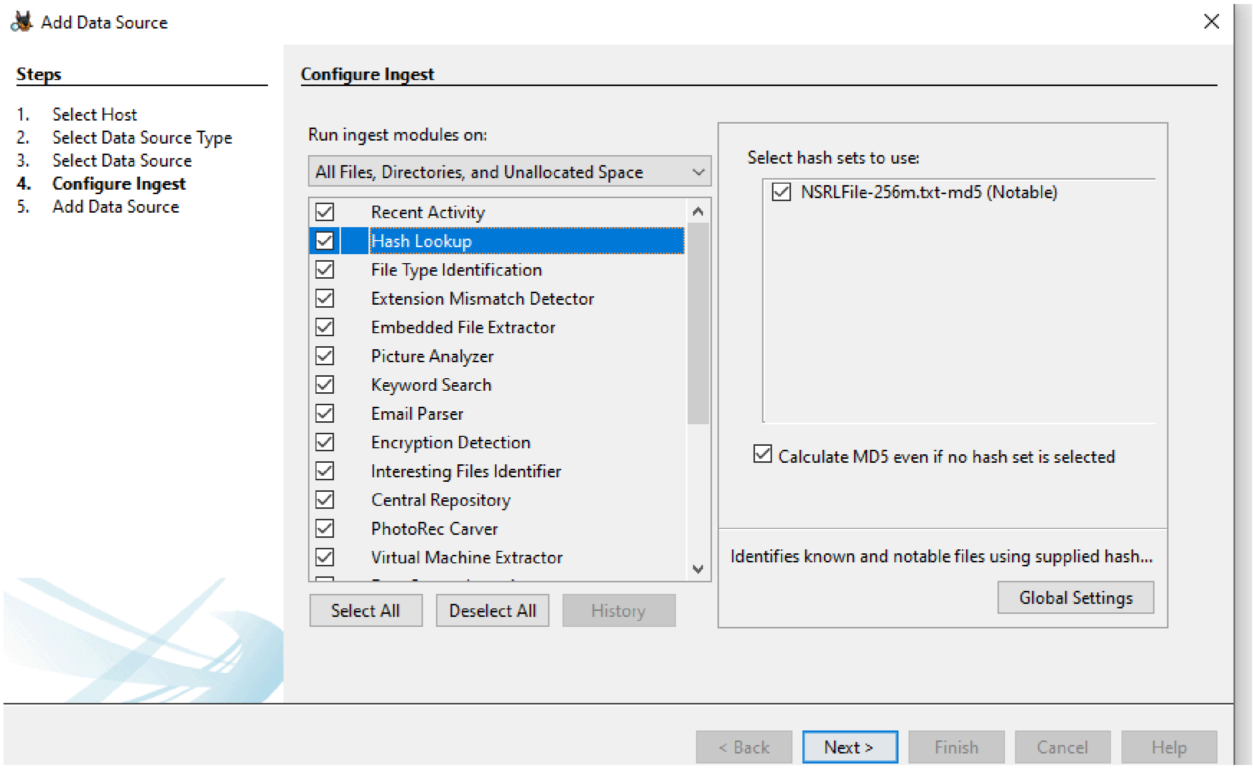
☑ Make hash set read-only

☑ Send ingest inbox message for each hit

☐ Copy hash set into user configuration folder

Successfully imported hash set into Autopsy.

Step 2: Create a Case



Run ingest modules on: All Files, Directories, and Unallocated Space

☑ Recent Activity
☑ Hash Lookup
☑ File Type Identification
☑ Extension Mismatch Detector
☑ Embedded File Extractor
☑ Picture Analyzer
☑ Keyword Search
☑ Email Parser
☑ Encryption Detection
☑ Interesting Files Identifier
☑ Central Repository
☑ PhotoRec Carver
☑ Virtual Machine Extractor

Select hash sets to use:

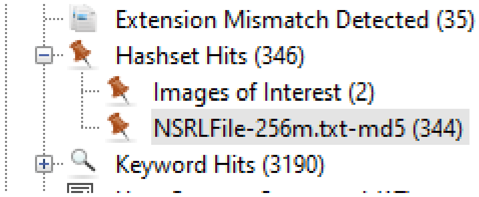
☑ NSRLFile-256m.txt-md5 (Notable)

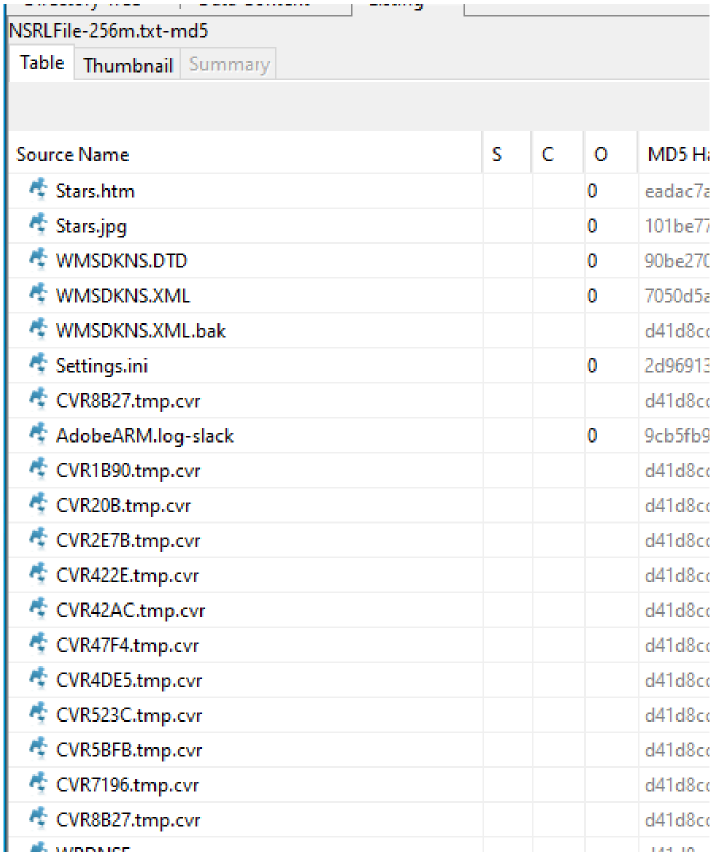
☑ Calculate MD5 even if no hash set is selected

Identifies known and notable files using supplied hash...

Created a new case with the files provided and the observed the hash lookup ingest module.

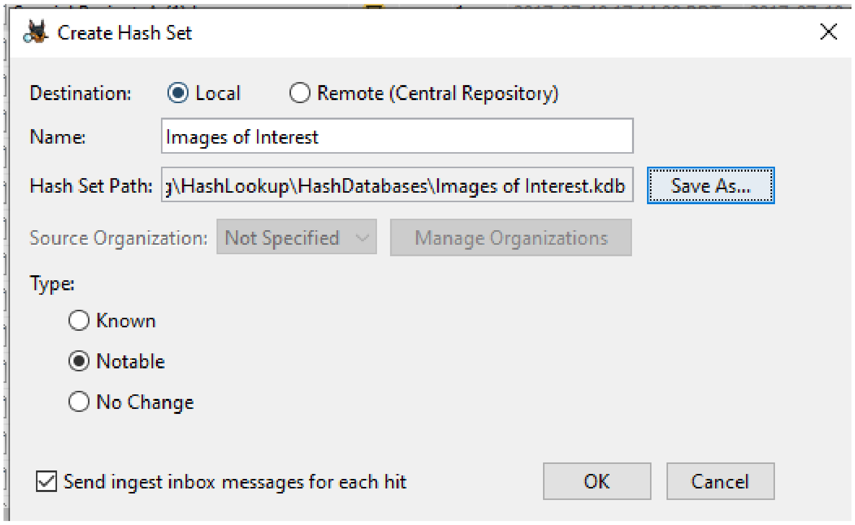
Step 3: Hashset Tracking


Extension Mismatch Detected (35)
Hashset Hits (346)
Images of Interest (2)
NSRLFile-256m.txt-md5 (344)
Keyword Hits (3190)



NSRLFile-256m.txt-md5

Source Name	S	C	O	MD5 H
Stars.htm			0	eadac7a
Stars.jpg			0	101be77
WMSDKNS.DTD			0	90be270
WMSDKNS.XML			0	7050d5a
WMSDKNS.XML.bak				d41d8cc
Settings.ini			0	2d96913
CVR8B27.tmp.cvr				d41d8cc
AdobeARM.log-slack			0	9cb5fb9
CVR1B90.tmp.cvr				d41d8cc
CVR20B.tmp.cvr				d41d8cc
CVR2E7B.tmp.cvr				d41d8cc
CVR422E.tmp.cvr				d41d8cc
CVR42AC.tmp.cvr				d41d8cc
CVR47F4.tmp.cvr				d41d8cc
CVR4DE5.tmp.cvr				d41d8cc
CVR523C.tmp.cvr				d41d8cc
CVR5BFB.tmp.cvr				d41d8cc
CVR7196.tmp.cvr				d41d8cc
CVR8B27.tmp.cvr				d41d8cc



Create Hash Set

Destination: ● Local ○ Remote (Central Repository)

Name: Images of Interest

Hash Set Path: g\HashLookup\HashDatabases\Images of Interest.kdb

Source Organization: Not Specified

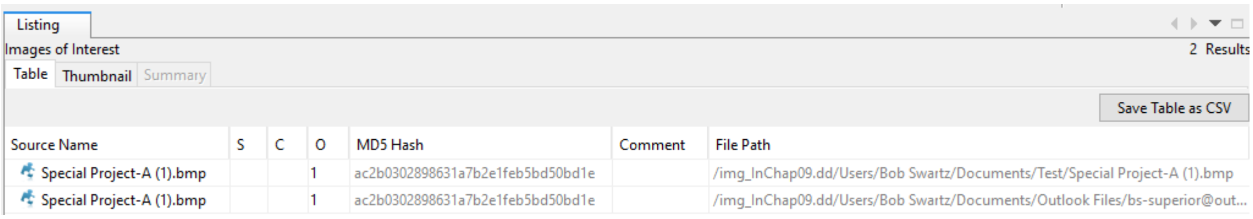
Type:

○ Known

● Notable

○ No Change

☑ Send ingest inbox messages for each hit



Images of Interest

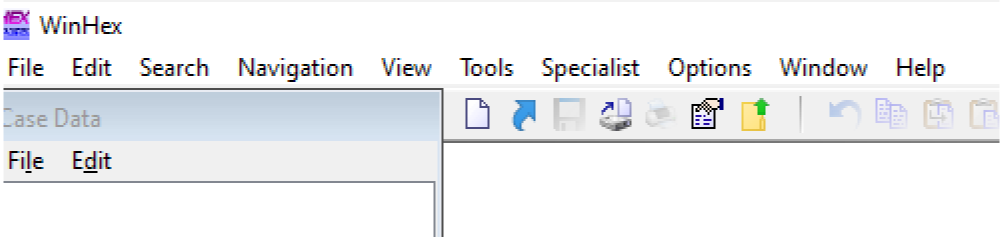
2 Results

Source Name	S	C	O	MD5 Hash	Comment	File Path
Special Project-A (1).bmp			1	ac2b0302898631a7b2e1feb5bd50bd1e		/img_InChap09.dd/Users/Bob Swartz/Documents/Test/Special Project-A (1).bmp
Special Project-A (1).bmp			1	ac2b0302898631a7b2e1feb5bd50bd1e		/img_InChap09.dd/Users/Bob Swartz/Documents/Outlook Files/bs-superior@out...

Successfully identified the images of interest using the hashset.

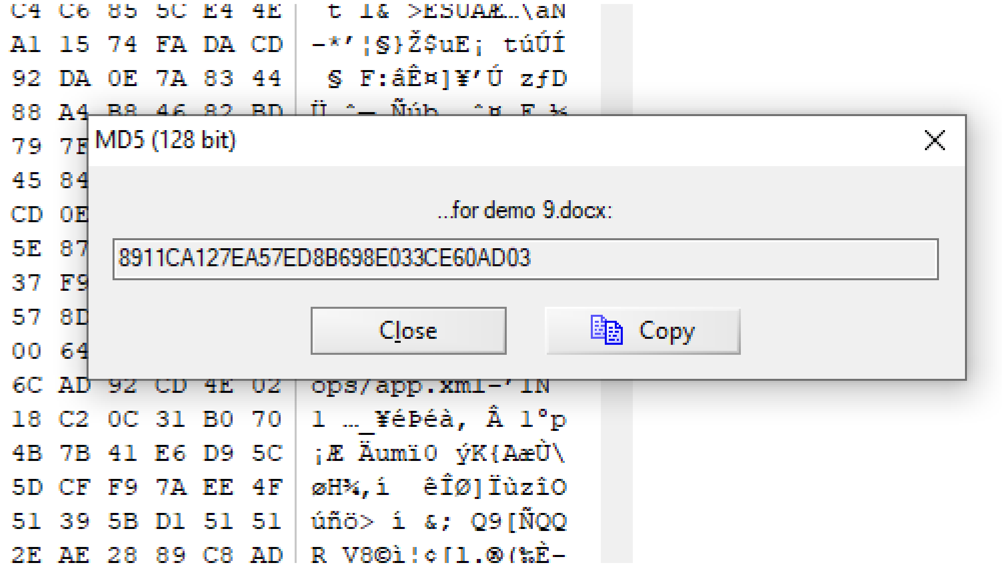
Task 2: Hashing with WinHex

Step 1: Install WinHex



Successful installation of WinHex application.

Step 2: Hash a Word File



```
C4 C6 85 5C E4 4E    t 1& >ESUAÆ…\aN
A1 15 74 FA DA CD   -*'¦§}Ž$uE; túÚÍ
92 DA 0E 7A 83 44    § F:âÊ¤]¥'Ú zfD
88 A4 B8 46 82 BD   Ü ^— Ñúb  ^¤ F ¾
79 7E
45 84
CD 0E
5E 87
37 F9
57 8D
00 64
6C AD 92 CD 4E 02   ops/app.xml-'IN
18 C2 0C 31 B0 70   1 …_¥éÞéà, Â 1°p
4B 7B 41 E6 D9 5C   ¡Æ ÄumiO ýK{AæÙ\
5D CF F9 7A EE 4F   øH¾,í  êÎØ]ÏùzîO
51 39 5B D1 51 51   úñö> í &; Q9[ÑQQ
2E AE 28 89 C8 AD   R V8©ì!¢[1.®(‰È-
```

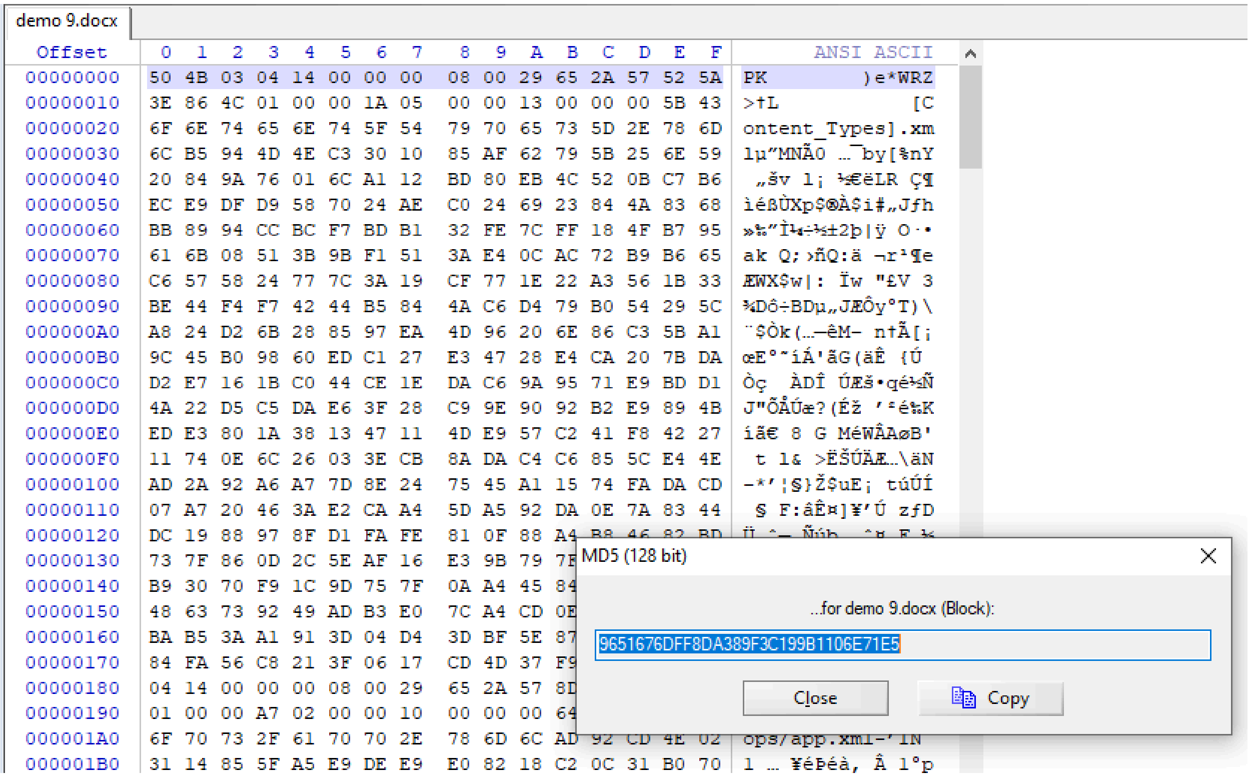
MD5 (128 bit)

...for demo 9.docx:

8911CA127EA57ED8B698E033CE60AD03

Successfully computed the hash of the Microsoft Word document.

Step 2: Hash a File Section



demo 9.docx

```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F     ANSI ASCII
00000000  50 4B 03 04 14 00 00 00  08 00 29 65 2A 57 52 5A  PK        )e*WRZ
00000010  3E 86 4C 01 00 00 1A 05  00 00 13 00 00 00 5B 43  >†L           [C
00000020  6F 6E 74 65 6E 74 5F 54  79 70 65 73 5D 2E 78 6D  ontent_Types].xm
00000030  6C B5 94 4D 4E C3 30 10  85 AF 62 79 5B 25 6E 59  lµ”MNÃ0 …¯by[%nY
00000040  20 84 9A 76 01 6C A1 12  BD 80 EB 4C 52 0B C7 B6   „šv l¡ ½€ëLR Ç¶
00000050  EC E9 DF D9 58 70 24 AE  C0 24 69 23 84 4A 83 68  ìéßÙXp$®À$i#„Jƒh
00000060  BB 89 94 CC BC F7 BD B1  32 FE 7C FF 18 4F B7 95  »‰”Ì¼÷½±2þ|ÿ O·•
00000070  61 6B 08 51 3B 9B F1 51  3A E4 0C AC 72 B9 B6 65  ak Q;›ñQ:ä ¬r¹¶e
00000080  C6 57 58 24 77 7C 3A 19  CF 77 1E 22 A3 56 1B 33  ÆWX$w|: Ïw "£V 3
00000090  BE 44 F4 F7 42 44 B5 84  4A C6 D4 79 B0 54 29 5C  ¾Dô÷BDµ„JÆÔy°T)\
000000A0  A8 24 D2 6B 28 85 97 EA  4D 96 20 6E 86 C3 5B A1  ¨$Òk(…—êM– n†Ã[¡
000000B0  9C 45 B0 98 60 ED C1 27  E3 47 28 E4 CA 20 7B DA  œE°˜`íÁ'ãG(äÊ {Ú
000000C0  D2 E7 16 1B C0 44 CE 1E  DA C6 9A 95 71 E9 BD D1  Òç  ÀDÎ ÚÆš•qé½Ñ
000000D0  4A 22 D5 C5 DA E6 3F 28  C9 9E 90 92 B2 E9 89 4B  J"ÕÅÚæ?(Éž ’²é‰K
000000E0  ED E3 80 1A 38 13 47 11  4D E9 57 C2 41 F8 42 27  íã€ 8 G MéWÂAøB'
000000F0  11 74 0E 6C 26 03 3E CB  8A DA C4 C6 85 5C E4 4E   t l& >ËŠÚÄÆ…\äN
00000100  AD 2A 92 A6 A7 7D 8E 24  75 45 A1 15 74 FA DA CD  -*’¦§}Ž$uE¡ túÚÍ
00000110  07 A7 20 46 3A E2 CA A4  5D A5 92 DA 0E 7A 83 44   § F:âÊ¤]¥’Ú zƒD
00000120  DC 19 88 97 8F D1 FA FE  81 0F 88 A4
00000130  73 7F 86 0D 2C 5E AF 16  E3 9B 79 7F
00000140  B9 30 70 F9 1C 9D 75 7F  0A A4 45 84
00000150  48 63 73 92 49 AD B3 E0  7C A4 CD 0E
00000160  BA B5 3A A1 91 3D 04 D4  3D BF 5E 87
00000170  84 FA 56 C8 21 3F 06 17  CD 4D 37 F9
00000180  04 14 00 00 00 08 00 29  65 2A 57 8D
00000190  01 00 00 A7 02 00 00 10  00 00 00 64
000001A0  6F 70 73 2F 61 70 70 2E  78 6D 6C AD
000001B0  31 14 85 5F A5 E9 DE E9  E0 82 18 C2 0C 31 B0 70  1 … ¥éÞéà‚ Â 1°p
```

MD5 (128 bit)

...for demo 9.docx (Block):

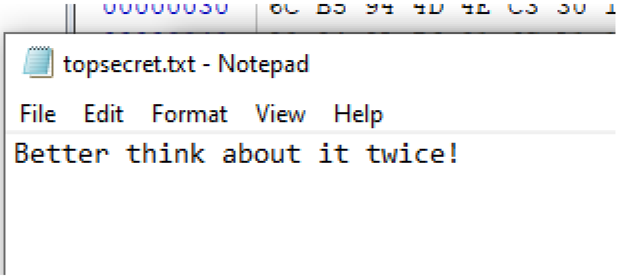
9651676DFF8DA389F3C199B1106E71E5

Created a hash for a segment of the file that is unique to the initial has of the whole file.

Task 3: Bit Shifting with WinHex

Step 1: Install Winhex (Already Installed)

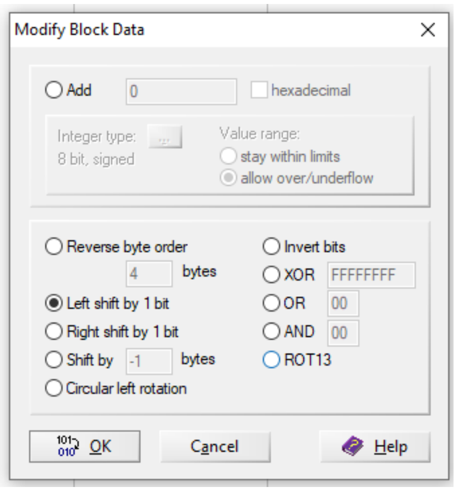
Step 2: Create a Secret File



Better think about it twice!

Created text document.

Step 3: Bit Shift Secret File


Modify Block Data

○ Add 0 □ hexadecimal

Integer type: 8 bit, signed

Value range: ○ stay within limits ● allow over/underflow

○ Reverse byte order 4 bytes

● Left shift by 1 bit

○ Right shift by 1 bit

○ Shift by -1 bytes

○ Circular left rotation

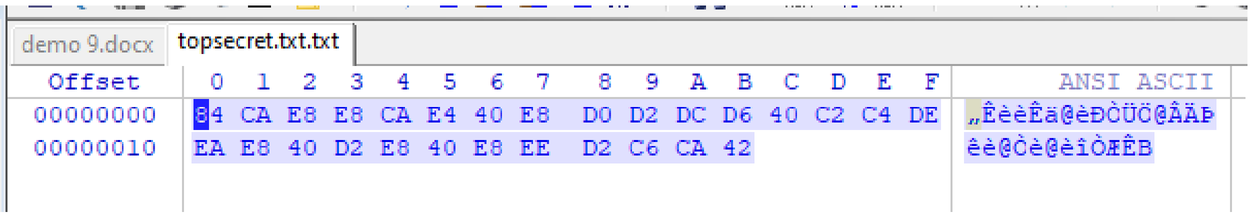
○ Invert bits

○ XOR FFFFFFFF

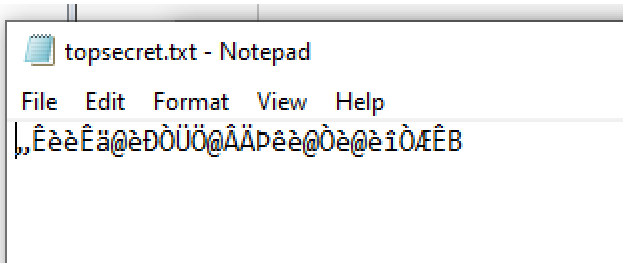
○ OR 00

○ AND 00

○ ROT13



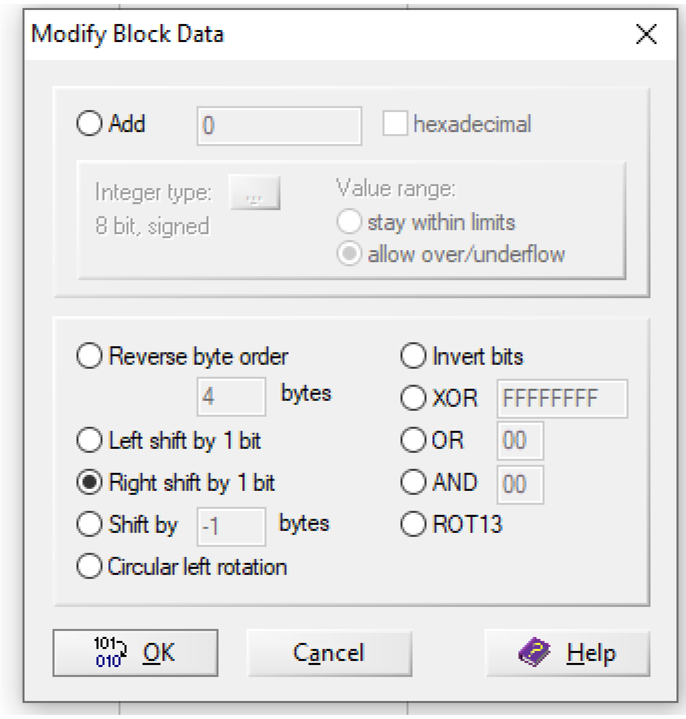
```
  Offset   0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F     ANSI ASCII
00000000  84 CA E8 E8 CA E4 40 E8  D0 D2 DC D6 40 C2 C4 DE  „ÊèèÊä@èÐÒÜÖ@ÂÄÞ
00000010  EA E8 40 D2 E8 40 E8 EE  D2 C6 CA 42              êè@Òè@èîÒÆÊB
```



```
„ÊèèÊä@èÐÒÜÖ@ÂÄÞêè@Òè@èîÒÆÊB
```

Successfully bit shifted the text file and made the ASCII phrase unreadable.

Step 4: Recover Bit Shifted File



Modify Block Data

○ Add 0 □ hexadecimal

Integer type: 8 bit, signed

Value range: ○ stay within limits ● allow over/underflow

○ Reverse byte order 4 bytes

○ Invert bits

○ XOR FFFFFFFF

○ Left shift by 1 bit

○ OR 00

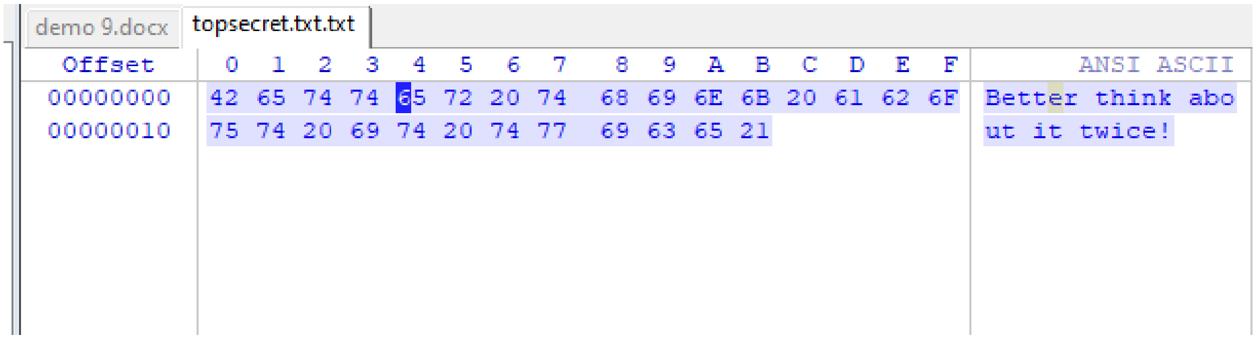
● Right shift by 1 bit

○ AND 00

○ Shift by -1 bytes

○ ROT13

○ Circular left rotation



```
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ANSI ASCII
00000000   42 65 74 74 65 72 20 74  68 69 6E 6B 20 61 62 6F   Better think abo
00000010   75 74 20 69 74 20 74 77  69 63 65 21               ut it twice!
```

Successfully recovered bit shifted file.