

1. What is an DD Image file and why are there different forensic image files and what are the considerations you should take when selecting an particular type of image file (.E01, .AD1, .001, .AFF, etc) you want?

Answer: **DD Image File:** A DD image file, often simply referred to as a "DD", is a bit-for-bit copy of a storage device (like a hard drive or a phone's internal memory). It's created using the dd command in Unix/Linux systems, hence the name. The 'dd' command essentially reads raw data from the source device and writes it to the destination file without any interpretation or modification. This creates an exact replica of the original drive, including all its data, file system structures, and even unallocated space.

➤ **Why different forensic image files exist:** Different forensic image file formats have evolved to cater to various needs and optimize the forensic investigation process. Here's a brief overview of some common formats:

- **.E01 (EnCase Image File):** Proprietary format developed by Guidance Software for use with their EnCase forensic suite. It offers features like compression, metadata storage, and built-in error detection.
- **.AD1 (AccessData Image File):** Proprietary format used by AccessData's FTK (Forensic Toolkit). Similar to E01, it provides compression and metadata support.
- **.001 (Raw Image with Metadata):** A raw (uncompressed) image file often accompanied by a separate metadata file (.info or .txt) containing information about the acquisition process and hash values.
- **.AFF (Advanced Forensic Format):** An open-source format designed to be flexible and extensible, supporting various compression and encryption options.

➤ **Considerations when selecting an image file format:** When choosing a forensic image file format, several factors should be considered:

- **Tool Compatibility:** Ensure that the chosen format is compatible with the forensic tools you intend to use for analysis. Some tools might have limitations or preferences for specific formats.
- **Compression:** If storage space is a concern, consider formats that offer compression, like E01 or AD1. However, keep in mind that compression can impact processing time during analysis.

- **Metadata Storage:** Some formats, like E01 and AD1, allow for embedding metadata about the acquisition process within the image file itself. This can be helpful for documentation and chain of custody purposes.
- **Open vs. Proprietary:** Open formats like AFF offer flexibility and interoperability between different tools, while proprietary formats might have specific features or optimizations for the corresponding forensic suite.
- **Legal Requirements:** Certain legal jurisdictions or investigative bodies might have specific requirements or preferences for image file formats.

A DD image file is a raw, bit-by-bit copy of a storage device. Different forensic image formats exist to address various needs and optimize investigations. The choice of format depends on tool compatibility, compression needs, metadata requirements, and potential legal considerations.

2. What is the file system of this image e.g. FAT 12, FAT 32, NTFS?

Answer: The File System of the image is **FAT**.

- **Evidence:** Based on the file architecture and layout, there are several reasons that this is a FAT file architecture:
- **"FAT12 3" in \$MBR file:** This is the most direct evidence. It explicitly mentions "FAT12", which is a specific type of FAT file system.
 - **"IO.SYS" and "MSDOS.SYS" in \$MBR:** These are core system files typically found in the root directory of FAT-formatted volumes, further supporting the FAT file system hypothesis.
 - **Presence of \$FAT1 and \$FAT2:** These are typical metadata structures found in FAT file systems.
 - **\$FAT1 and \$FAT2** are the two copies of the File Allocation Table, a core component of FAT that tracks which clusters on the disk are allocated to files and which are free.
 - **Absence of NTFS-specific Metadata:** The current file system on the device also doesn't show any structures commonly associated with NTFS, such as \$MFT (Master File Table) or \$LogFile.

➤ **Evidence location:** /img_CCE_Sample_PE.001/\$MBR

Table Thumbnail Summary

Page: 1 of 1 Pages: Go to Page:

| Name | S | C | O | Modified Time | Change Time | Access Time | Created Time | Size |
|------------------------------|---|---|---|-------------------------|---------------------|---------------------|---------------------|------|
| \$OrphanFiles | | | | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0 |
| \$FAT1 | | | 3 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 4608 |
| \$FAT2 | | | 3 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 4608 |
| \$MBR | | | 2 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 512 |
| \$CarvedFiles | | | | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0 |
| \$Unalloc | | | | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0 |
| BYE-BYE (Volume Label Entry) | | | | 2004-09-15 14:33:58 EDT | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0 |

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Strings Extracted Text Translation

Page: 1 of 1 Page Matches on page: - of - Match 100% Reset

25BYE-BYE
8NS)
r>8-t
at=Nt
Invalid system disk
Disk I/O error
Replace the disk and then press any key
IO
SYSMSDOS SYS

3. What is the volume label of this image?

Answer: The volume label of this image is "BYE-BYE". This information can be found in the file metadata of the image file. Also, the information under the "Name" column in the table, where it is labeled as "(Volume Label Entry)" specifies the name.

➤ **Evidence Location:** /img_CCE_Sample_PE.001/BYE-BYE (Volume Label Entry)

Table Thumbnail Summary

Page: 1 of 1 Pages: Go to Page:

| Name | S | C | O | Modified Time | Change Time | Access Time | Created Time |
|------------------------------|---|---|---|-------------------------|---------------------|---------------------|---------------------|
| \$OrphanFiles | | | | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| \$FAT1 | | | 3 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| \$FAT2 | | | 3 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| \$MBR | | | 2 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| \$CarvedFiles | | | | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| \$Unalloc | | | | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| BYE-BYE (Volume Label Entry) | | | | 2004-09-15 14:33:58 EDT | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

From The Sleuth Kit istat Tool:

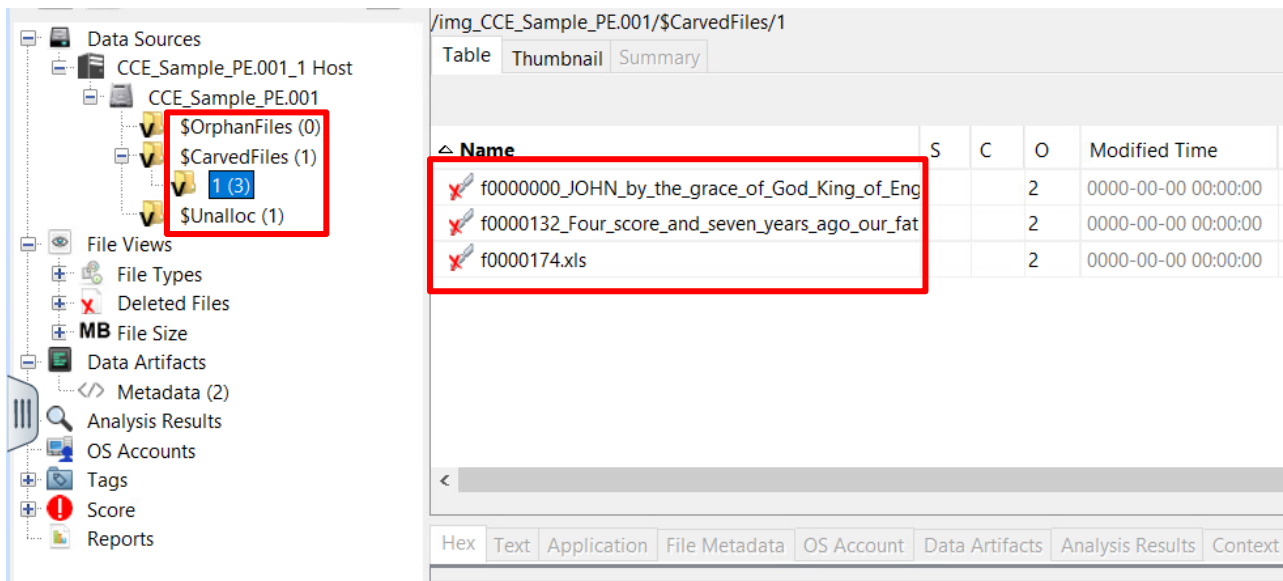
Directory Entry: 3
Allocated
File Attributes: Volume Label, Archive
Size: 0
Name: BYE-BYE
Directory Entry Times:
Written: 2004-09-15 14:33:58 (EDT)

4. This image looks to be empty – where do you think data could be recovered e.g. FAT 1, FAT 2, VBR, unallocated space and why?

Answer: Yes, despite the image appearing empty, there's still various location within the image from where data can be recovered.

➤ **Potential Locations:**

- I. **Unallocated Space (\$Unalloc):** This area holds data that has been deleted or marked as free by the file system but hasn't been overwritten yet.
 - **Potential Data:** Fragments or even complete files that were once present on the drive but were later deleted.
 - **\$Unalloc (1)** indicates that there's currently one item or data fragment detected within the unallocated space of the analyzed disk image.
- II. **File Carving (\$CarvedFiles):** File carving techniques can scan the entire image (including unallocated space) and try to identify files based on their headers and footers (unique patterns at the beginning and end of files).
 - **Potential Data:** Files that were deleted or whose file system structures are damaged, making them inaccessible through traditional means.
 - **(3) under \$CarvedFiles** shows that three files have been successfully carved or recovered from the image and placed within the \$CarvedFiles folder.
- III. **\$OrphanFiles:** These are file fragments that Autopsy couldn't associate with any file system structure.
 - **Potential Data:** Parts of deleted files or data from corrupted file systems.
 - **(0) under \$OrphanFiles** suggest that there are no files located in this portion of the image.



5. You decide that file carving is necessary. What is file carving and how might that be helpful to you?

Answer: File carving is a forensic technique used to recover files from a storage device even when the file system's metadata (like file allocation tables or directory structures) is damaged, missing, or deliberately erased. It operates by scanning the raw data on the storage media and identifying files based on their characteristic headers and footers (unique patterns of bytes at the beginning and end of files).

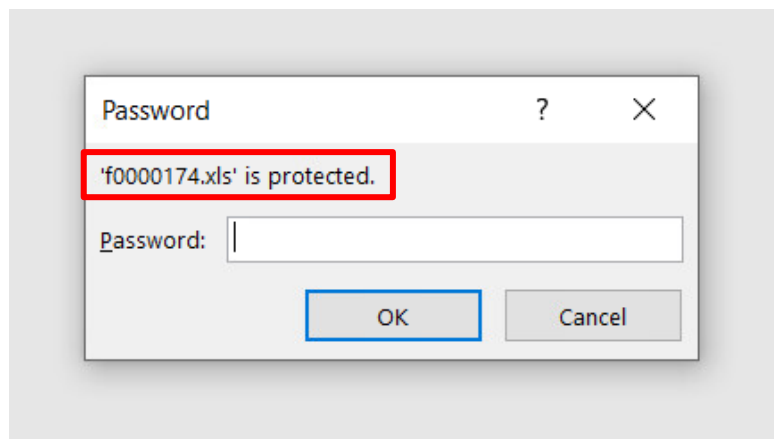
➤ How File Carving Might Be Helpful in this Case:

- I. **Overcoming File System Issues:** When the forensic image seems to have some issues with the file system like "Invalid system disk" and "Disk I/O error" messages, file carving can bypass these problems by looking directly for file content rather than relying on the file system's organization.
- II. **Recovering Deleted Files:** Even if files were deleted from the FAT file system, their data might still exist in the unallocated space. File carving can identify and extract these files based on their headers and footers.
- III. **Finding Hidden Data:** If someone tried to hide data by removing it from the file system's view, file carving could potentially uncover it.

In the context of this image, file carving can help recover files that were previously on the disk but are not visible in the file system's current state. This could include user documents, images, emails, or any other type of file that has a recognizable header and/or footer.

6. You see some Word and Excel documents. One of those Excel files is password-protected. What is the MD5 Hash value of that password protected file?

Answer: The file “**f0000174.xls**” is the password protected file. To verify it is password protected, it needs to be extracted first and when tried to access, it asks for password. Thus, it is a passwords protected file.



- **Evidence Location:** /img_CCE_Sample_PE.001/\$CarvedFiles/1/f0000174.xls
- **MD5 Hash:** 20d237dcc709a1ad25796cf0869d5a6c. This MD5 hash can be found in the file metadata section of the excel file.

/img_CCE_Sample_PE.001/\$CarvedFiles/1

Table Thumbnail Summary

| Name | S | C | O | Modified Time | Change Time | Access Time |
|---|---|---|---|---------------------|---------------------|---------------------|
| f0000000_JOHN_by_the_grace_of_God_King_of_Eng | | | 2 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| f0000132_Four_score_and_seven_years_ago_our_fat | | | 2 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| f0000174.xls | | | 2 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Metadata Allocation: Unallocated

Modified: 0000-00-00 00:00:00

Accessed: 0000-00-00 00:00:00

Created: 0000-00-00 00:00:00

Changed: 0000-00-00 00:00:00

MD5: 20d237dcc709a1ad25796cf0869d5a6c

SHA-256: 56e1b34930c4742636fd0c327ac0f9926c7acf00ab5fb6723a7a7227ff74c8da

Hash Lookup Results: UNKNOWN

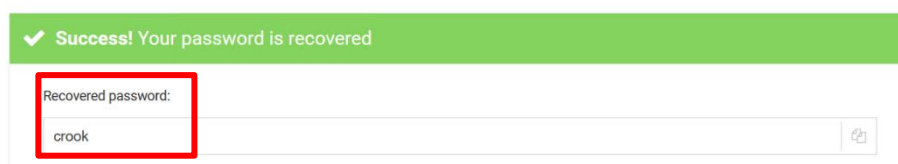
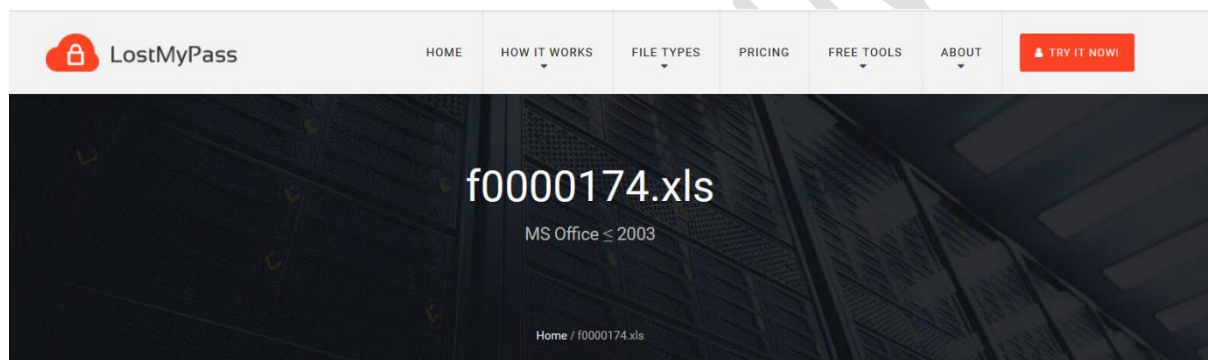
Internal ID: 16

```
C:\Users\student\Desktop>certutil -hashfile f0000174.xls MD5
MD5 hash of f0000174.xls:
20d237dcc709a1ad25796cf0869d5a6c
CertUtil: -hashfile command completed successfully.
```

Verified the hash and found to be matching with the one in Autopsy.

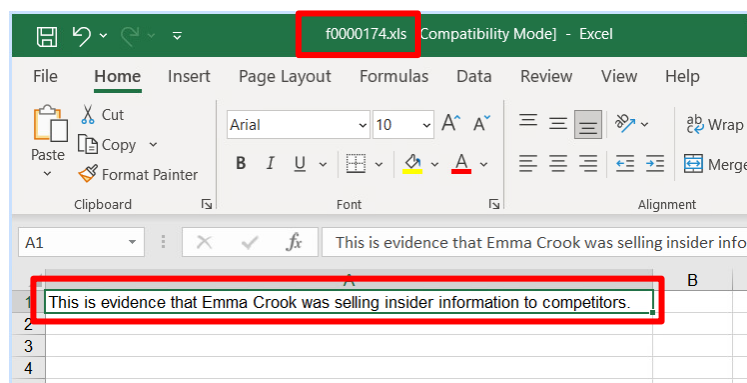
7. You find the password protected file. It is asking you for a password.
What is the password?

Answer: The password of this file is the Last name of Emma i.e. “**crook**”. This file is not only password protected but also write protected which doesnot allow anyone to edit or modify its content until correct password is entered. The password for the write protected encryption is also “**crook**”.



8. You figure out the password – what is the information in the file?

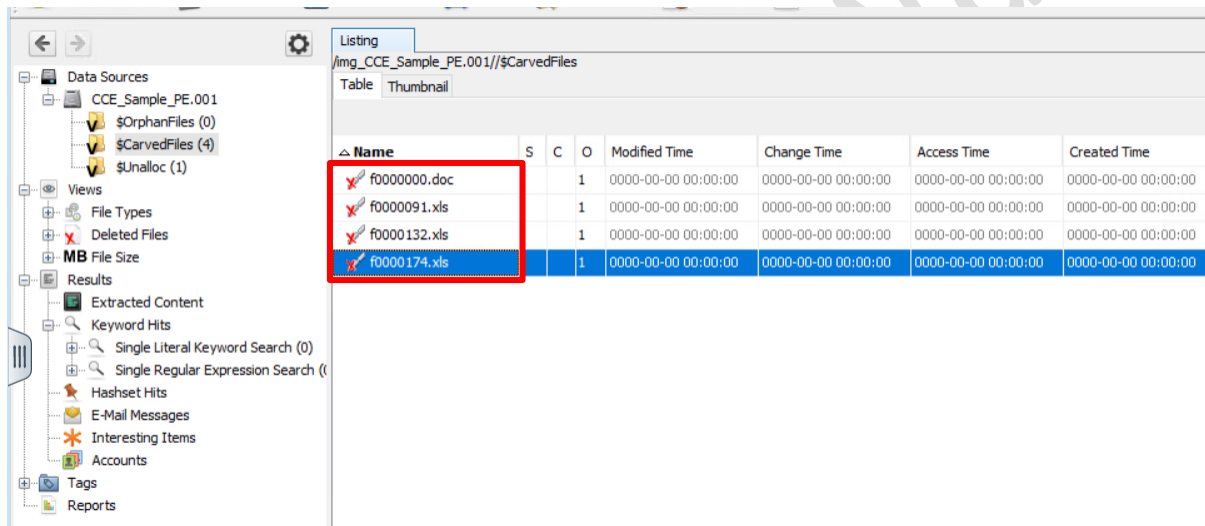
Answer: The information in the file is “**This is evidence that Emma Crook was selling insider information to competitors.**”



9. You look at the other Excel spreadsheet file(s). You can view the metadata and its content in Autopsy but when you extract them and try to open them in Excel you get an “invalid file format”. You puzzle over it then you figure it out – it’s because this is not an Excel file. It is something else. What kind of file (document type) is this? and how can you fix this?

Answer: There are a total of 4 files available in the system. They are “f0000000.doc, f0000091.xls, f0000132.xls and f0000174.xls”. But only 2 files namely, **f0000000.doc** and **f0000174.xls** have the correct file extension and is in valid format. Rest 2, **f0000091.xls** and **f0000132.xls** have incorrect file format.

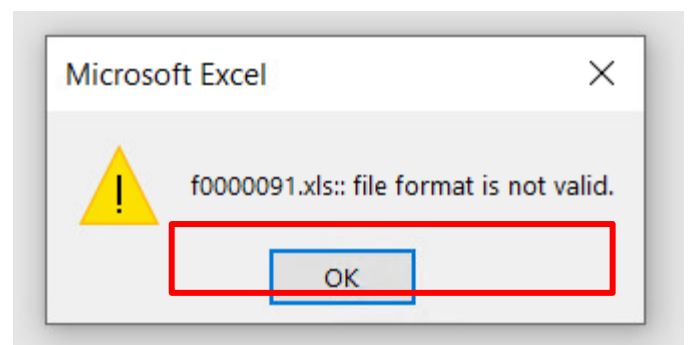
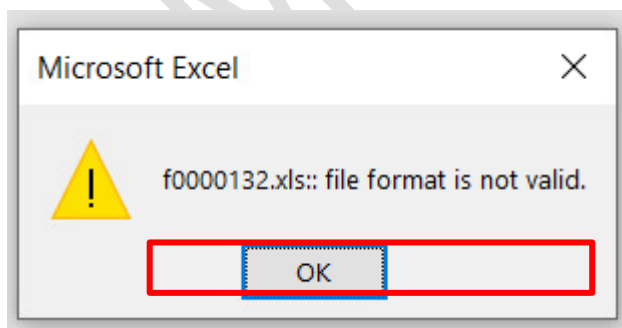
➤ **Evidence:**



| Name | S | C | O | Modified Time | Change Time | Access Time | Created Time |
|----------------|---|---|---|---------------------|---------------------|---------------------|---------------------|
| ✗ f0000000.doc | | | 1 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| ✗ f0000091.xls | | | 1 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| ✗ f0000132.xls | | | 1 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |
| ✓ f0000174.xls | | | 1 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |

➤ **Evidence Location:**

/img_CCE_Sample_PE.001//CarvedFiles/f0000000.doc

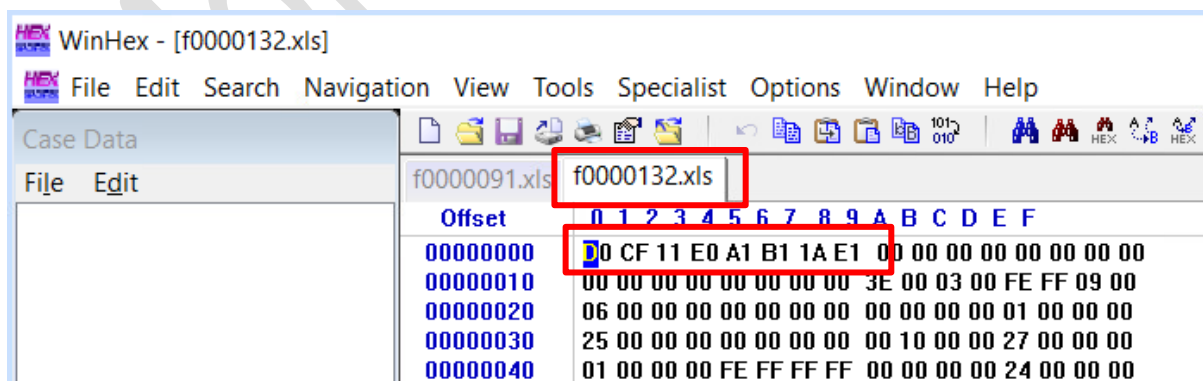
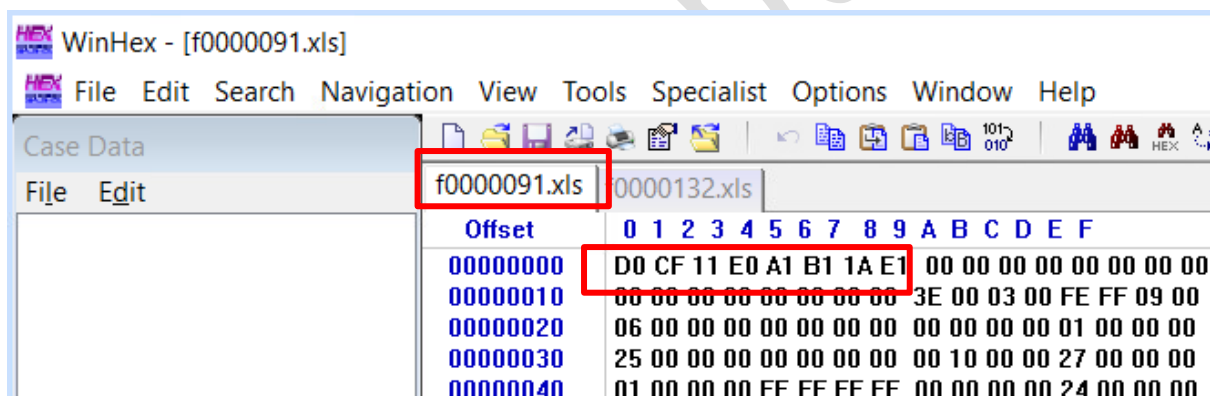


➤ **Verification of incorrect file format:**

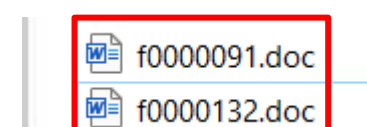
The rest of the 2 files namely **f0000091.xls** and **f0000132.xls** are not .xls files but they are .doc (Word) file. It can be obtained from the **header of the file signature** (HEX value) of both the file. Because the file signature of the .doc file is **“D0 CF 11 E0”**

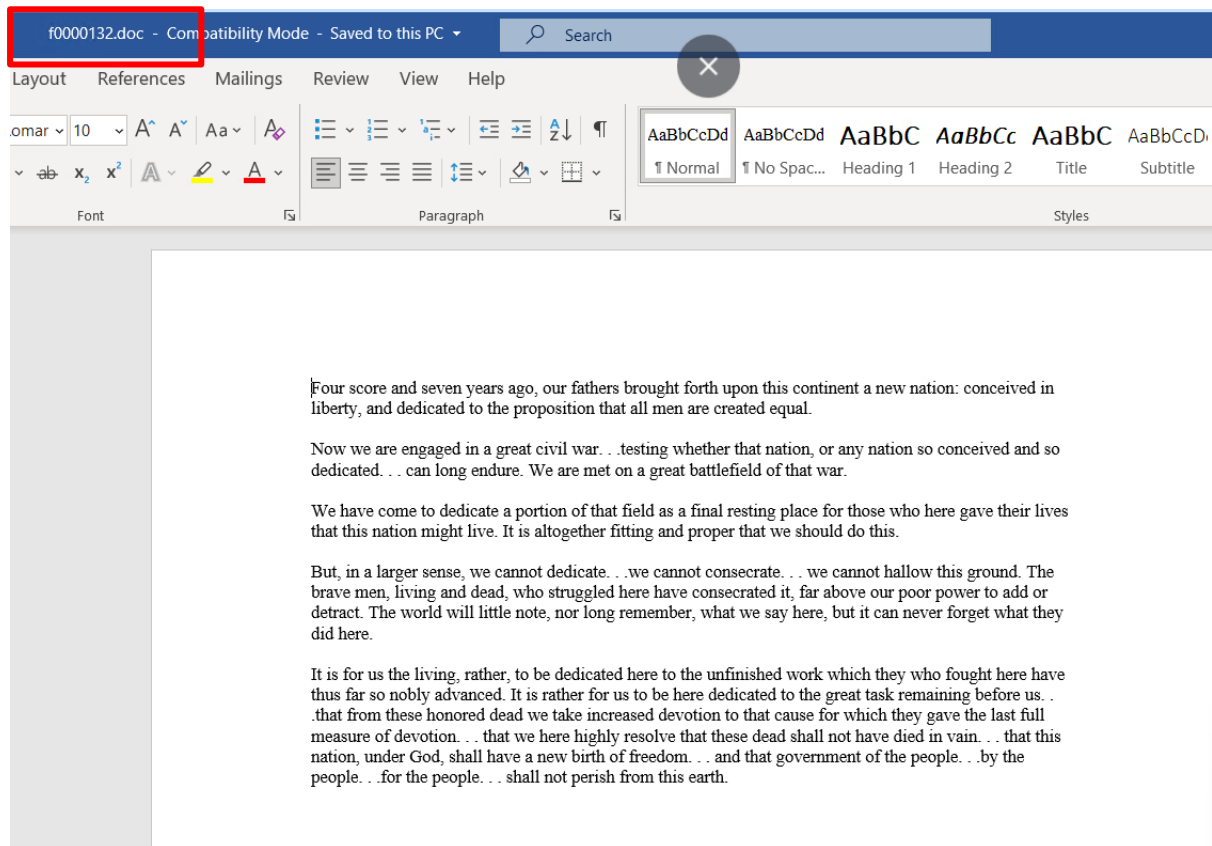
| | | | | |
|-------------------------|-----------|---|---------------------------------|---|
| D0 CF 11 E0 A1 B1 1A E1 | Đİ°\à ±°á | 0 | doc xls ppt msi msg | Compound File Binary Format, a container format defined by Microsoft COM. It can contain the equivalent of files and directories. It is used by Windows Installer and for documents in older versions of Microsoft Office. ^[43] It can be used by other programs as well that rely on the COM and OLE API's. |
|-------------------------|-----------|---|---------------------------------|---|

- Here, in this case, the file signature of both the file is:

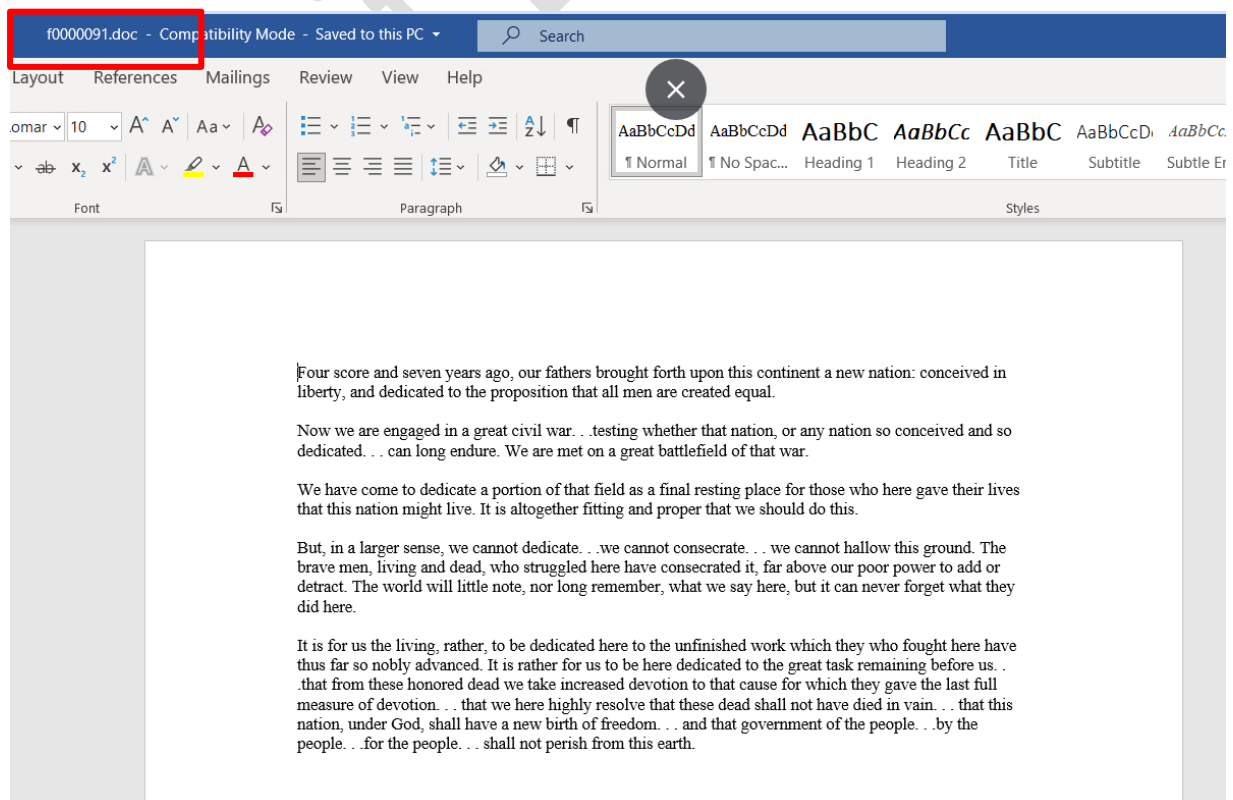


To get the correct file format, change the file extension from .xls to .doc. Doing this will give the file with correct file format.





- As soon as the file extension is changed to **.doc**, the correct file will be obtained.



Extra Credit Question

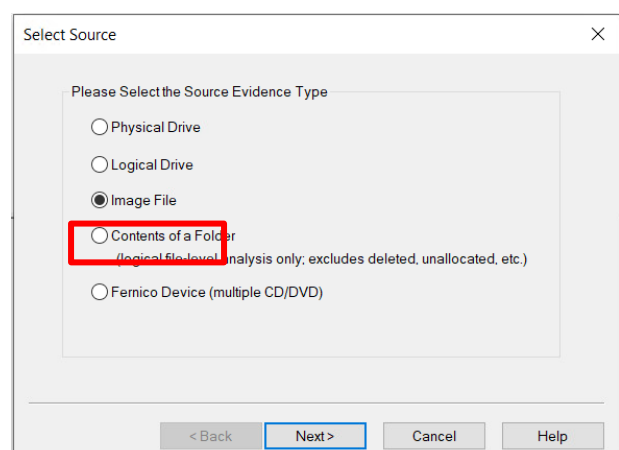
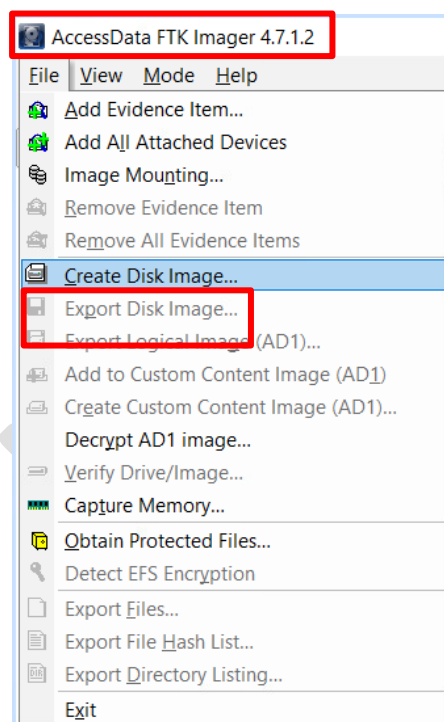
Q: You may have a tool that will not work with .001 images (DD) images, but the tool can handle E01 files. How can you convert the DD image into an E01 image?

Answer: DD images are simple, bit-for-bit copies of a disk. They lack the built-in metadata that some tools rely on for features like case Information (examiner details, case numbers, etc.), hash values, compression. Segmentation, Proprietary Features and tool Complexity.

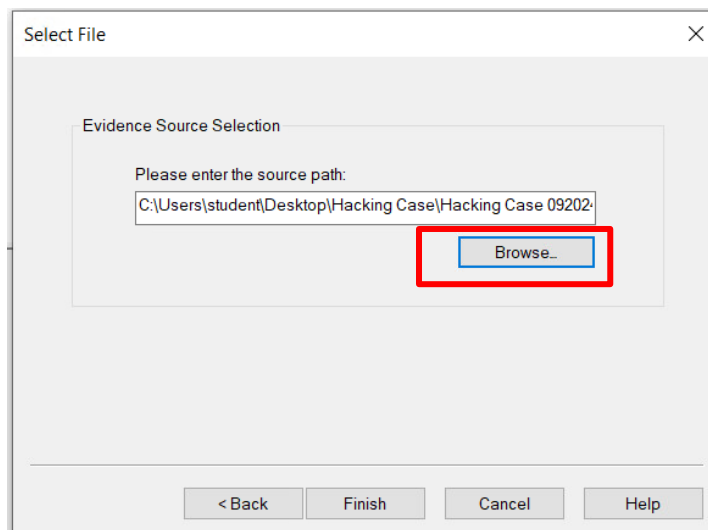
- To convert a DD image file to a E01 file, **FTK imager** is used. **FTK Imager** is a popular, free tool from AccessData that can create, convert, and analyze forensic images.

Steps:

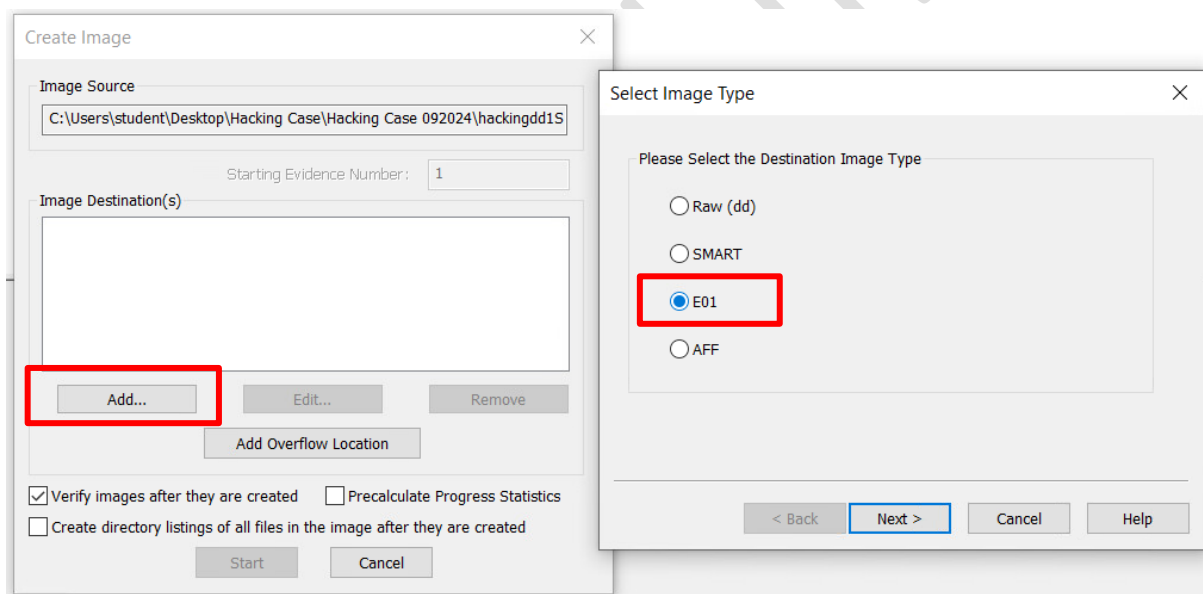
1. Open FTK Imager and select "Create Disk Image", select the disk source and it type and click next.



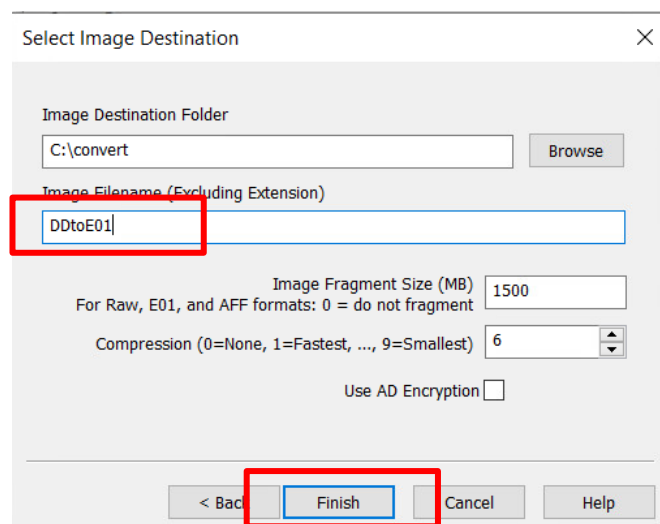
2. Choose the source .001 image file.



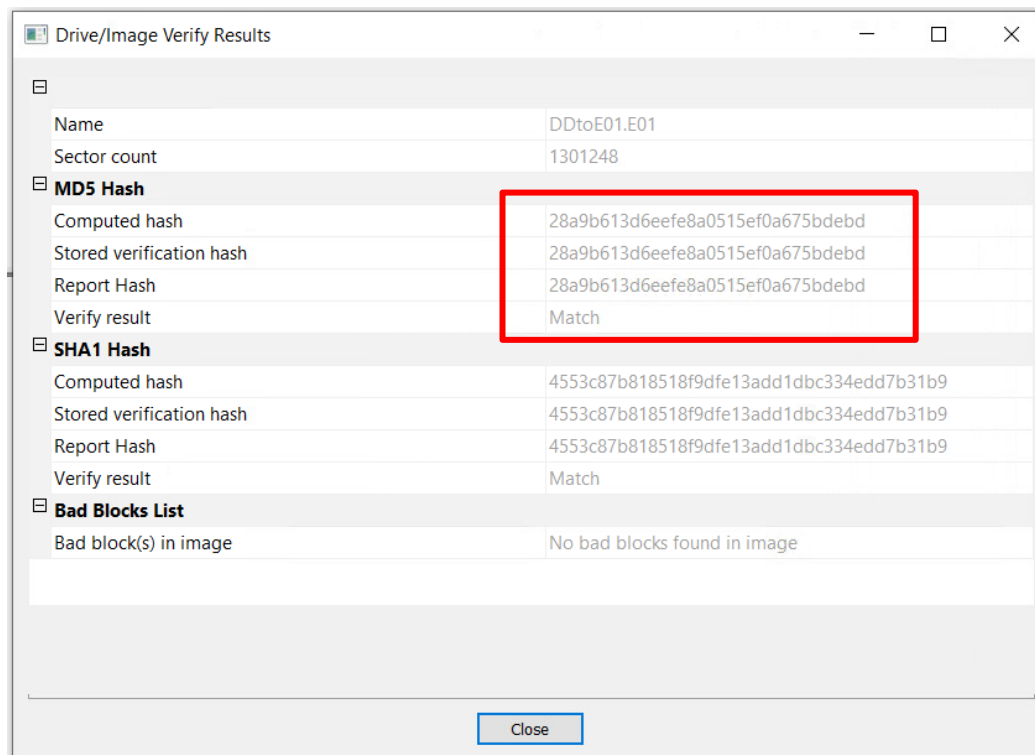
3. In the "Image Destination" section, select "E01" as the output format.



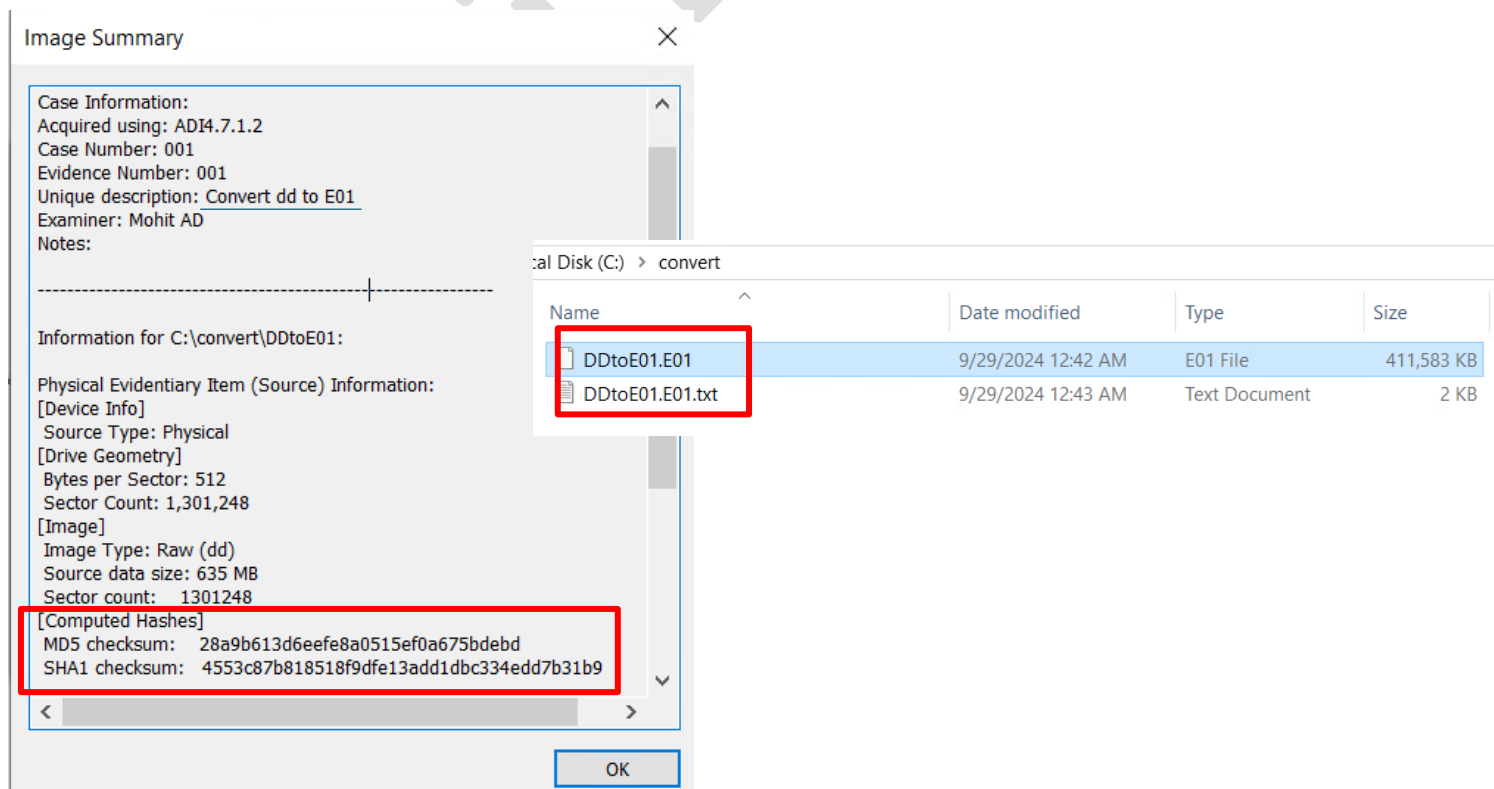
4. Configure any additional options (compression level, case information, etc.) if desired and click finish.



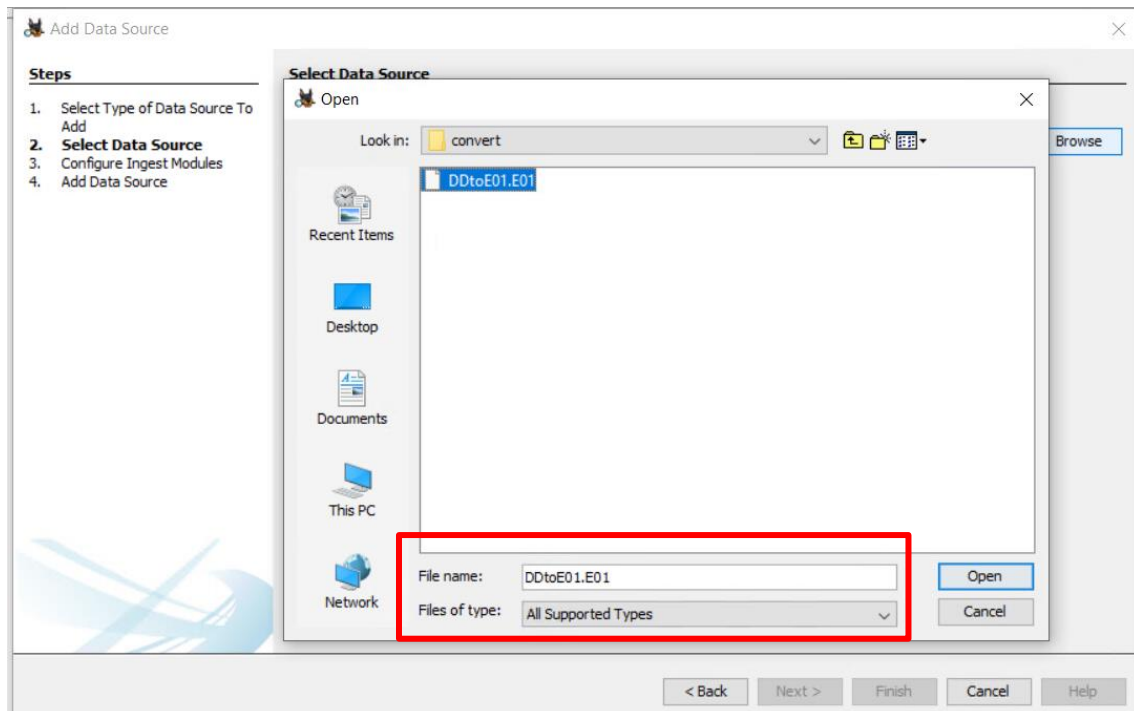
5. Click "Create Image" to start the conversion process. Once the conversion is complete, the **Verify Results** dialog box will appear.



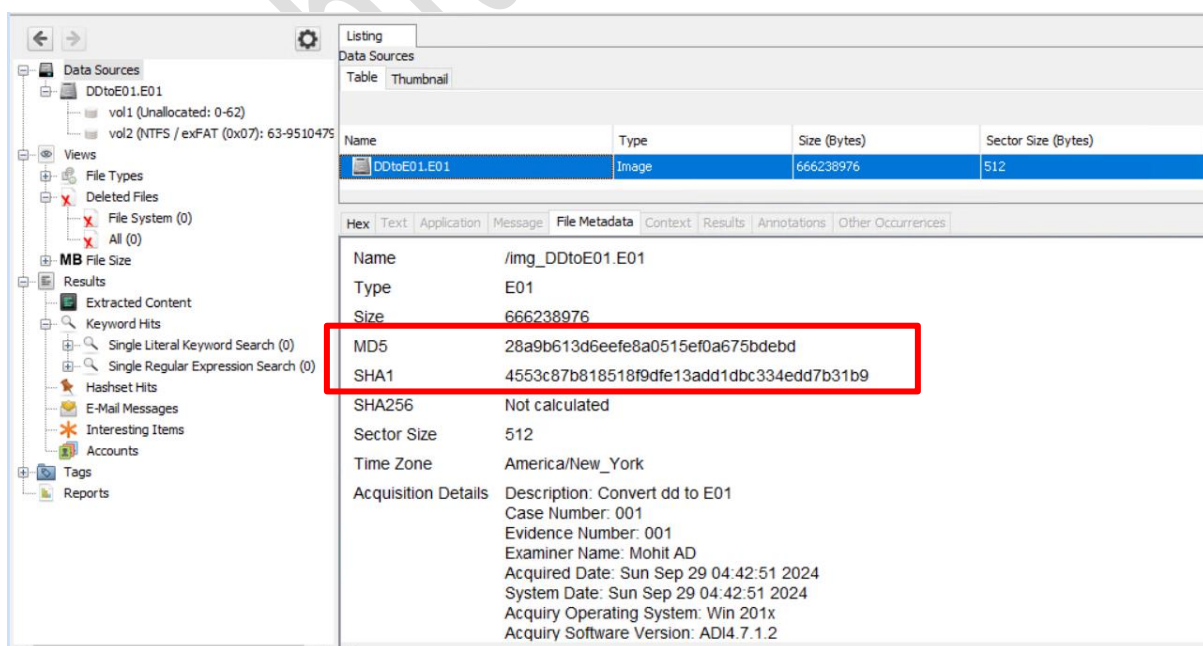
6. After the conversion, FTK will generate an **Image Summary** mentioning all the details like the case information, image information, and hash values.



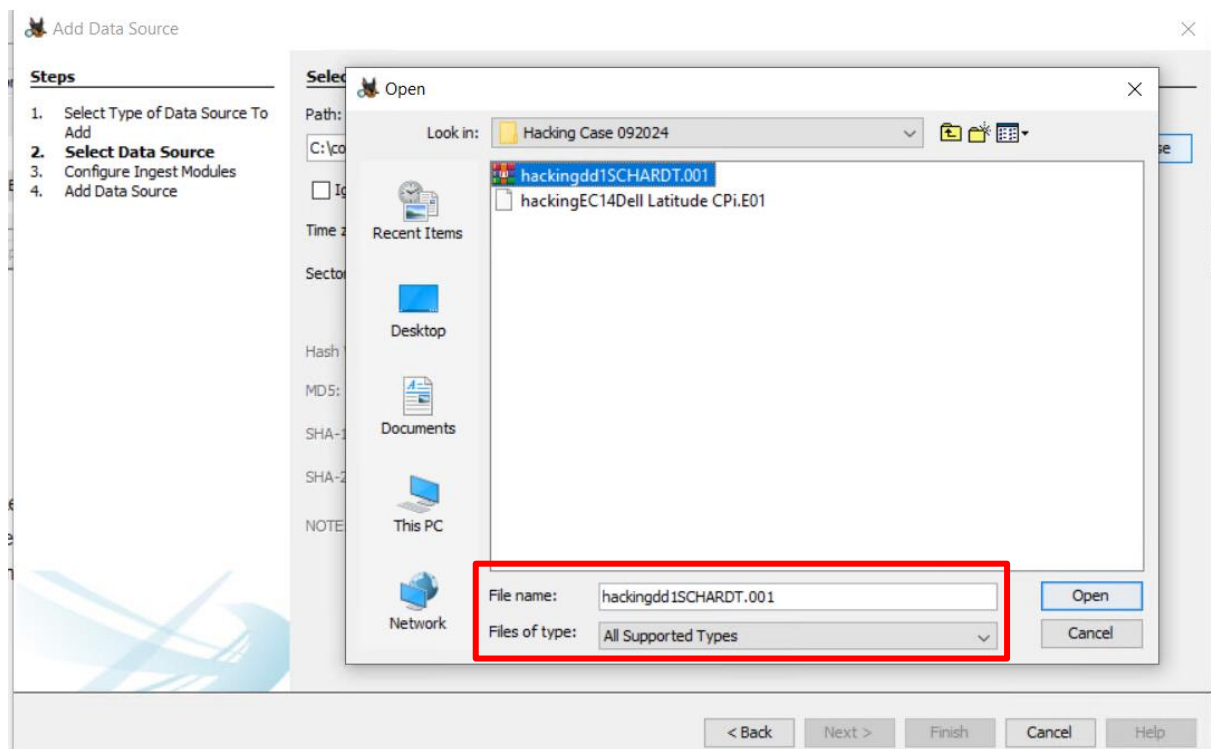
- Now to verify whether the image is converted properly, and the contents are same as in the previous version, it is loaded into Autopsy for verification.



- After loading the case in Autopsy, check the file contents and the calculated hash value. It is found to be same as the previous version.



9. Also to match the contents and the hash value, load the original DD image into Autopsy.



10. As seen in the image, the fact is verified, and we have successfully converted a DD image file into a E01 image file using FTK Imager.

