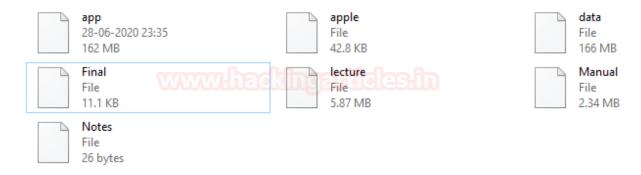
# Forensic Investigation: Examining Corrupted File Extension

July 8, 2020 By Raj Chandel

In this article, we will learn how we can Examine Corrupted File Extension to identify the basic file header in a Forensic Investigation.

### Let's understand this with the following Scenario

In this Scenario, a forensic investigator has gone for an investigation and found out a suspicious folder where no file has any kind of file extension. Now, what will he do to proceed in his forensic investigation?



**Objective:** Learn to use various techniques in Forensic Investigation to analyse and examine the various file headers

- Examining Corrupted File Extension using Windows Platform.
- Examining Corrupted File Extension using Linux Platform.

## **Table of Content**

#### **Cheatsheet for Hex File Header**

### **Examining Corrupted File Extension using Windows Platform**

- File #1: app
- File #2: apple
- File #3: data
- File #4: Final
- File #5: lecture
- File #6: Manual
- File #7: Notes
- Recovered all files successfully

### **Examining Corrupted File Extension using Linux Platform**

• Analyze in Linux with file command

Analyze in Linux

### **Cheatsheet for Hex File Header**

We all know that the hex file header is used to identify any file by examining the first 4 or 5 bytes of its hexadecimal content.

We have created our very own cheat sheet to examine these values more appropriately; Which contains all the basic files extensions and its 4 to 5 bytes starter hexadecimal value along with its ASCII translation.

File	STARTER	STARTER ASCII
Extensions	HEX VALUE	TRANSLATION
exe	4D 5A 50 00	MZP
exe 📆	4D 5A 80 00	Classiff MZ
dll	4D 5A 90 00	MZ
png	89 50 4E 47	PNG
mp3	49 44 33 2E	ID3
mp3	49 44 33 03	ID3
docx	50 4B 03 04 14 00 06	PK
zip	50 4B 03 04	PK
rar	52 61 72 21	<u>Rar</u> !
PDF	25 50 44 46	%PDF
jpg	FF D8 FF E0	ÿØÿà
jpeg	FF D8 FF FE	JFIF
Linux bin	7F 45 4C 46	ELF
ani	52 49 46 46	RIFF
gif	47 49 46 38 39 61	GIF89a
gif	47 49 46 38 37 61	GIF87a
cab	4D 53 43 46	MSCF
au	2E 73 6E 64	<u>Snd</u>
bmp	42 4D F8 A9	ВМ
bmp	42 4D 62 25	BMp%
bmp VVV	42 4D 76 03	BMv
OFT	4F 46 54 32	OFT2
sfw	43 57 53 06/08	cws

# **Examining Corrupted File Extension using Windows Platform**

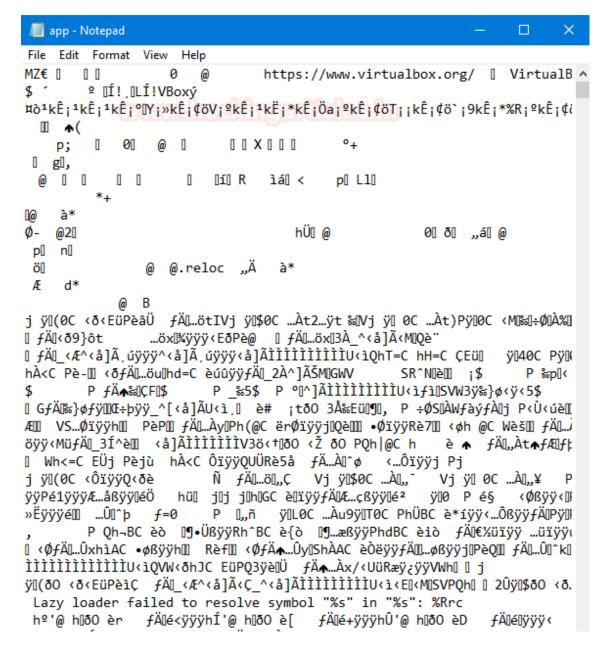
As per the given scenario, the first thing which comes into our mind that let's check these files in the command prompt [cmd]. Nevertheless, nothing is visible to the investigator.

Now Let's try to examine each file we found this folder and try to restore them in their original format.

```
C:\Users\SKS19\Desktop\forensics>dir
 Volume in drive C has no label.
 Volume Serial Number is C296-F770
 Directory of C:\Users\SKS19\Desktop\forensics
07-07-2020
            20:44
                     <DIR>
            20:44
07-07-2020
                     CDTRN
                        170,617,352 app
            23:36
28-06-2020
03-07-2020
            19:14
                             43,853 apple
            21:14
03-07-2020
                        174,786,466 data
03-07-2020
            19:53
                             11,403 Final
03-07-2020
           19:58
                          6,162,016 lecture
15-04-2020
           21:44
                          2,456,722 Manual
03-07-2020
           19:12
                                 26 Notes
                            354,077,838 bytes
               7 File(s)
               2 Dir(s) 361,035,194,368 bytes free
C:\Users\SKS19\Desktop\forensics>_
```

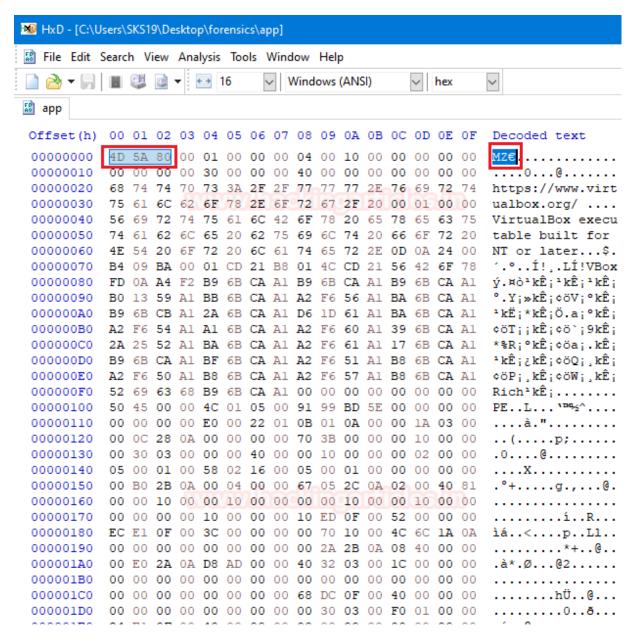
## File #1: app

The first file, which we got is app. The first thing that comes into our mind is to open this file with the help of notepad. We are doing it to show you guys that the file is in an unreadable format.



Now, we examine hexadecimal values of these files with Hexadecimal editor. We can use any kind of hexadecimal editor, that can show us a hexadecimal value of any file. After opening that file, we need to examine its stating hexadecimal value to know about the file type.

So, I have used Hex Workshop which you can download from here.



After, analyzing its starting bytes with our cheat sheet. We come to know that it is a **.exe file** with its **ASCII translation MZ**. MZ is the initials of Mark Zbikowski, he is the designer of the DOS executable file format. We have successfully investigated the first file as a .exe file.

Now, we have two methods to rename that file extension.

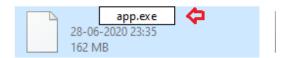
**Method 1:** With the help of the command line.

Follow this command to rename this file extension.

```
rename *app. *.exe
```

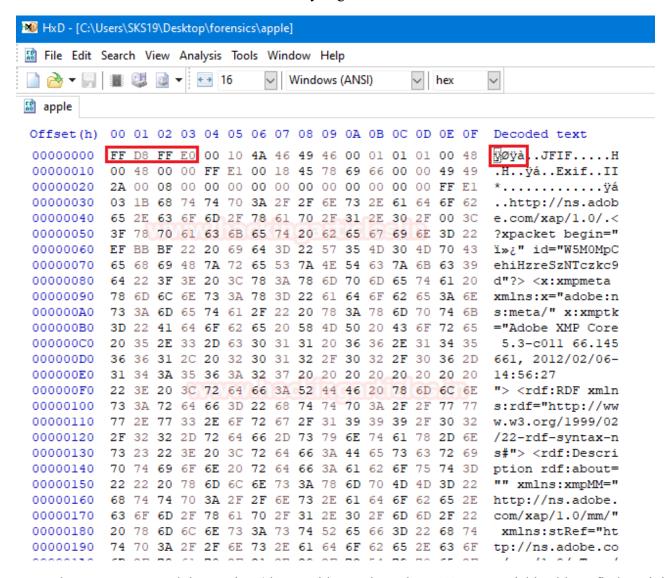
This command helps us to select only the app file to rename only this file extension. Because others are yet to be examined.

**Method 2:** We can simply change it directly by renaming the file name and providing it with an extension which we already find above.



## File #2: apple

Now, it's time to examine the second file all we know about that file is its name apple. Straight away we opened that file in the hexadecimal editor. To start analyzing its hexadecimal values.



As we have to try to match its starting 4 bytes with our cheat sheet. We were quickly able to find out it is a .jpg file with ASCII translation ÿØÿà.

Now, just rename this file with the help of this command.

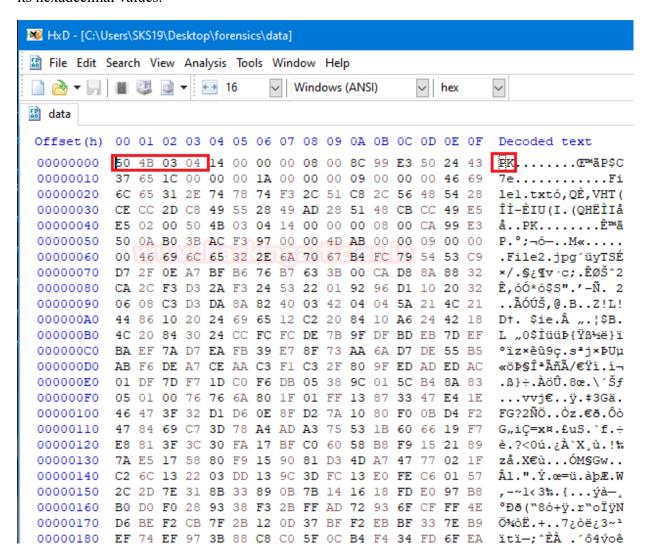
```
rename *apple. *.jpg
```

This command will only change the apple file to a .jpg file. Because others are yet to be examined.

```
C:\Users\SKS19\Desktop\forensics>rename *apple. *.jpg
```

### File #3: data

Time to examine the third file which name is data. We are opening that file into a hexadecimal editor, to examine its hexadecimal values.



Now, try to match it first 4 bytes with our cheat sheet which we provide above. In a few moments, we find out that it is a **.zip** file with **ASCII translation PK**.

Change the file name and provide it with an extension with the help of rename command.

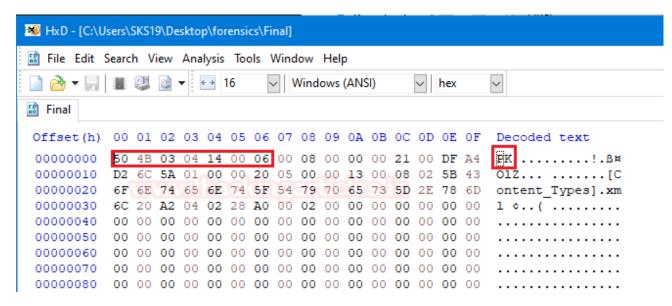
```
rename *data. *.zip
```

As we know it will only make changes in data and change it into a .zip file.

```
C:\Users\SKS19\Desktop\forensics>rename *data. *.zip
```

### File #4: Final

Here comes the fourth file which name is Final. Now, open that file in a hexadecimal editor to analyse its hexadecimal values.



After opening that file, try to match its first seven bytes with our cheat sheet. In a few moments, we found out that its values match with a .docx file. So, it is a .docx file with ASCII translation **PK**.

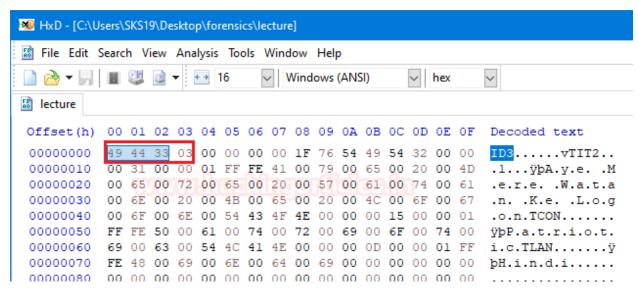
Just change its name and provide it with a .docx extension with the help of [rename] command.

```
rename *Final. *.docx
```

C:\Users\SKS19\Desktop\forensics>rename \*Final. \*.docx

## File #5: lecture

The fifth file named as a lecture; we try to open that file in a hexadecimal editor. To analyse its hexadecimal values, which helps us to identify its file type.



Now, try to match its first four bytes. In a moment we found out that it is a .mp3 file with an ASCII translation ID3. Just provide that file a .mp3 extension with the help of [rename] command.

```
rename *lecture. *.mp3
```

```
C:\Users\SKS19\Desktop\forensics>rename *lecture. *.mp3
```

### File #6: Manual

The second last file in that folder named Manual. Open that file in a hexadecimal editor to examine its hexadecimal values.

```
M HxD - [C:\Users\SKS19\Desktop\forensics\Manual]
File Edit Search View Analysis Tools Window Help
                                  Windows (ANSI)
Manual
Offset(h)
           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
                                                             Decoded text
           25 50 44 46 2D 31 2E 37
                                   OD OA 25 B5 B5 B5 OD
                                                             %PDF-1.7..%µµµµ.
 00000010
           0A 31
                 20
                    30 20 6F 62 6A 0D 0A 3C 3C 2F 54 79 70
                                                             .1 0 obj..<</Typ
 00000020
          65 2F 43 61 74 61 6C 6F 67 2F 50 61 67 65 73 20
                                                             e/Catalog/Pages
 00000030
          32 20 30 20 52 2F 4C 61 6E 67 28 65 6E 2D 55 53
                                                             2 0 R/Lang(en-US
          29 20 2F 53 74 72 75 63 74 54 72 65 65 52 6F 6F
                                                             ) /StructTreeRoo
          74 20 31 38 36 37 20 30 20 52 2F 4D 61 72 6B 49
 00000050
                                                            t 1867 0 R/MarkI
 00000060
           6E 66 6F 3C 3C 2F 4D 61 72 6B 65 64 20 74 72 75
                                                             nfo<</Marked tru
 00000070
          65 3E 3E 2F 4D 65 74 61 64 61 74 61 20 31 31 36
                                                             e>>/Metadata 116
 08000000
          39 37 20 30 20 52 2F 56 69 65 77 65 72 50 72 65
                                                             97 0 R/ViewerPre
 00000090
          66 65 72 65 6E 63 65 73 20 31 31 36 39 38 20 30
                                                            ferences 11698 0
           20 52 3E 3E 0D 0A 65 6E 64 6F 62 6A 0D 0A 32 20
 000000A0
                                                              R>>..endobj...2
           30 20 6F 62 6A 0D 0A 3C 3C 2F 54 79 70 65 2F 50
 000000B0
                                                             0 obj..<</Type/P
          61 67 65 73 2F 43 6F 75 6E 74 20 31 39 33 2F 4B
 000000C0
                                                             ages/Count 193/K
 000000D0
          69 64 73 5B 20 33 20 30 20 52 20 32 36 20 30 20
                                                             ids[ 3 0 R 26 0
 000000E0 52 20 35 30 20 30 20 52 20 35 31 20 30 20 52 20
                                                            R 50 0 R 51 0 R
 000000F0
           35 32 20 30 20 52 20 35 33 20 30 20 52 20 35 34
                                                             52 0 R 53 0 R 54
 00000100
          20 30 20 52 20 35 35 20 30 20 52 20 35 36 20 30
                                                              0 R 55 0 R 56 0
 00000110
           20 52 20 35 37 20 30 20 52 20 35 38 20 30 20 52
                                                              R 57 0 R 58 0 R
           20 35 39 20 30 20 52 20 36 30 20 30 20 52 20 36
                                                              59 0 R 60 0 R 6
 00000120
           31 20 30 20 52 20 36 32 20 30 20 52 20 36 33 20
 00000130
                                                             1 0 R 62 0 R 63
           30 20 52 20 36 34 20 30 20 52 20 36 35 20 30 20
 00000140
                                                             0 R 64 0 R 65 0
              20 26 26 20 20 20 52 20 26 27 20 20
```

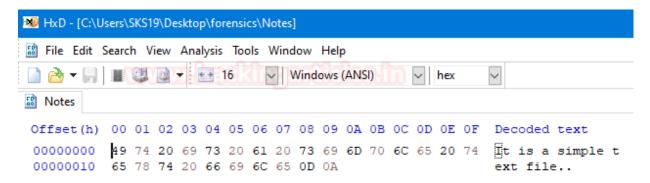
Now, try to match its four bytes with our cheat sheet. Then we come to know that it is a .pdf file with ASCII translation %PDF. Change its name and provide .pdf extension to it, with the help of rename command.

```
rename *Manual. *.pdf
```

C:\Users\SKS19\Desktop\forensics>rename \*Manual. \*.pdf

### File #7: Notes

Finally, we have reached to the file in the folder named Notes. Straight away we opened that file in a Hexadecimal editor to examine its hexadecimal values.



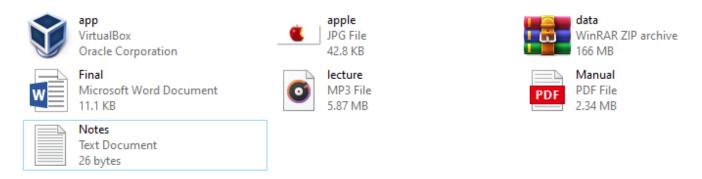
After opening it is saying that "It is a simple text file." so, we provided a .txt extension with the help of [rename] command.

```
rename *Notes. *.txt
```

```
C:\Users\SKS19\Desktop\forensics>rename *Notes. *.txt
```

#### Recovered all File successfully

Now, look at the folder which itself saying that we have recovered all the files successfully.



And we can also see that these files were recovered in the command prompt along with its original extension, with the help of [dir] command.

```
C:\Users\SKS19\Desktop\forensics>dir
Volume in drive C has no label.
Volume Serial Number is C296-F770
Directory of C:\Users\SKS19\Desktop\forensics
           21:51
07-07-2020
                     <DIR>
                     <DIR>
07-07-2020
           21:51
28-06-2020
           23:36
                        170,617,352 app.exe
03-07-2020
           19:14
                             43,853 apple.jpg
03-07-2020 21:14
                        174,786,466 data.zip
03-07-2020 19:53
                             11,403 Final.docx
03-07-2020
           19:58
                          6,162,016 lecture.mp3
15-04-2020
           21:44
                          2,456,722 Manual.pdf
03-07-2020 19:12
                                 26 Notes.txt
               7 File(s)
                            354,077,838 bytes
               2 Dir(s) 361,037,729,792 bytes free
```

## **Examining Corrupted File Extension using Linux Platform**

Now suppose in your investigation, you are in the same scenario where the file extension is missing but this time the Victim machine operates on Linux Environment and you are not allowed to copy this folder on another machine. Then How would you handle this situation?

## **Analysis using the File command**

The file command is a Linux utility that analyzes each argument in an attempt to classify it. Hence, we can examine this forensic investigation in a Linux environment with the help of file command.

We are using the [ls] command to show you guys, these are the same files and the same scenario that we already explained above.

```
root@kali:/home/kali/Desktop/forensics# ls
app apple data Final lecture Manual Notes
```

We just need to use [file] along with the file name, to know about the originality of that file. Pick the first file and use this command. It shows that it is an MS Windows executable file.

file app

```
root@kali:/home/kali/Desktop/forensics# file app
app: PE32 executable (GUI) Intel 80386, for MS Windows
```

Let us try the same technique with the second file named apple. Apply [file] command and provide its file name. It shows that it is a jpeg image along with its internal pieces of information.

file apple

```
root@kali:/home/kali/Desktop/forensics# file apple
apple: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72×72
, segment length 16, Exif Standard: [TIFF image data, little-endian, dirent
ries=0], baseline, precision 8, 1024×440, components 3
```

This article will help us to identify the true identity of a file during a Forensic Investigation in both the Windows and Linux environments.