# Index

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Practical name** | **Page No.** |
| 1 | Message encryption over a channel | [2](#_bookmark0) |
| 2 | Custom logging module | [7](#_bookmark1) |
| 3 | Searching files in a given directory | [10](#_bookmark2) |
| 4 | Searching a word in a given file | [13](#_bookmark3) |
| 5 | Virus that eats disk space | [16](#_bookmark4) |
| 6 | Disk backup | [19](#_bookmark5) |
| 7 | Forensic images of a disk | [28](#_bookmark6) |
| 8 | Retrieving deleted files | [35](#_bookmark7) |
| 9 | Registry Editor | [43](#_bookmark8) |

**Practical 1**

**Aim:** Create a Java file to send an encrypted message from the sender end and decrypt it at the receiver’s end.

## Source Code:

**Sender,java:**

import java.io.DataOutputStream; import java.io.IOException; import java.net.Socket;

import java.net.UnknownHostException;

import java.util.Random;

import java.util.Scanner;

/\*\*

* Sen er: Sends an encryp ed essage and ge era ed ey
* to t e eceiver.
* U es Soc ets for communication.

\*/

public class Sender {

/\*\*

* @param args Command li e argu ents

\*/

public static void main(String[] args) {

int counter = 0;

String cipherText = "", key = "";

Random random = ew Random();

Scan er scanner = ew Scan er(Sys em.in);

try {

Soc et socket = ew Soc et("localhost", 6017);

DataOutputSt eam dataOutputStream = ew DataOutputSt eam(socket.getOutputStream()); Sys em.out.println("Enter message: "); String message = scanner.nextLine();

/\*

* Co e for encryption.
* Working:
* 1. Ge era e an array of n ( ength of t e essage) random numbers.
* 2. Add t e co ePoints of t e essage with t e array eq enti lly.
* 3. Append t e typecas ed charac er to t e cip er ext.

\*/

int[] keyArray = ew int[message.length()];

for (char messagePart : message.toCharArray()) { keyArray[counter] = random.nextInt(50);

key += In eger.valueOf(keyArray[counter]) + ":"; cipherText += (char)(messagePart + keyArray[counter]); counter +;

}

Sys em.out.println("Message: " + message);

Sys em.out.println("Generated key: " + key);

Sys em.out.println("Encrypted message: " + cipherText);

dataOutputStream.writeUTF(cipherText); dataOutputStream.writeUTF(key);

scanner.close(); dataOutputStream.flush(); dataOutputStream.close(); socket.close();

}

catch (UnknownHostException e) {

Sys em.err.println("Error: Host not found."); e.printStackTrace();

}

catch (IOException e) {

Sys em.err.println("IOError: Some I/O operations could not be performed.");

e.printStackTrace();

}

}

}

## Receiver.java:

import java.io.DataInputStream; import java.io.IOException; import java.net.ServerSocket; import java.net.Socket;

/\*\*

* Receiver: Receives an encryp ed essage and ey om t e en er
* and ecrypts it.
* U es Soc ets for communication.

\*/

public class Receiver {

public static void main(String[] args) {

String message = "";

int counter = 0;

try {

ServerSoc et serverSocket = ew ServerSoc et(6017); Soc et socket = serverSocket.accept();

DataInputSt eam dataInputStream = ew

DataInputSt eam(socket.getInputStream());

cipherText = dataInputStream.readUTF(); key = dataInputStream.readUTF();

*String String*

/\*

* Co e for ecryption.
* Working:
* 1. Split t e ey string using t e ':' elimi er and convert it into an in eger.
* 2. Subtract t e array v l es om t e co ePoints eq enti lly.
* 3. Append t e typecas ed charac er to t e essage.

\*/

int[] keyArray = ew int[cipherText.length()]; for (String keyPart : key.split(":")) { keyArray[counter] = In eger.parseInt(keyPart); message += (char)(cipherText.charAt(counter) - keyArray[counter]);

counter +;

}

System.out.println("Ciphertext: " + cipherText);

System.out.println("Key: " + key);

System.out.println("Message: " + message);

dataInputStream.close(); socket.close(); serverSocket.close();

}

catch (IOException e) {

Sys em.err.println("IOError: Some I/O operations could not be performed");

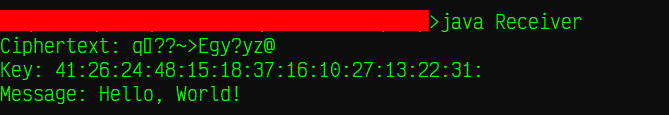
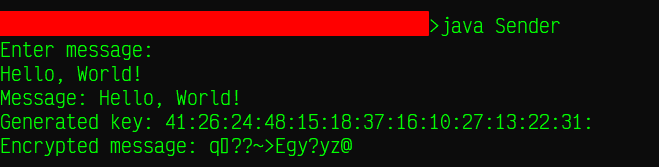
e.printStackTrace();

}

}

}

## Output:



**Sender**

**Receiver**

# Practical 2

**Aim:** Create a Java file to create a logger.

## Source Code:

import java.io.FileWriter; import java.io.IOException; import java.time.LocalDateTime; import java.util.Random;

import java.time.format.DateTimeFormatter;

public class CustomLogger {

F leWri er fileWriter;

public CustomLogger(String filePath, boo ean appendMode) {

try {

fileWriter = ew F leWri er(filePath, appendMode);

}

catch (IOException e) {

Sys em.err.println("IOError: File could not be opened"); e.printStackTrace();

}

}

void writeLog(String message, String intensity) { datetime = Da eTi eFormat er.ofPattern("yyyy/MM/dd

*public String*

HH:mm:ss").format(Loc lDa eTi e.now()); try {

fileWriter.write(datetime + "\t\t" + message + "\t\t" +

intensity + "\n"); fileWriter.flush();

}

catch (IOException e) {

Sys em.err.println("IOError: Log could not be written"); e.printStackTrace();

}

}

public void close() { try { fileWriter.close();

} catch (IOException e) {

Sys em.err.println("IOError: File could not be closed"); e.printStackTrace();

}

}

public static void main(String[] args) {

CustomLogger customLogger = ew CustomLogger("log.txt", tr e);

String[] intensity = {"INFO", "WARNING", "ERROR", "CRITICAL"};

Random random = ew Random();

for (int i = 0; i < 10; i +) { customLogger.writeLog("Log " + i, intensity[random.nextInt(4)]);

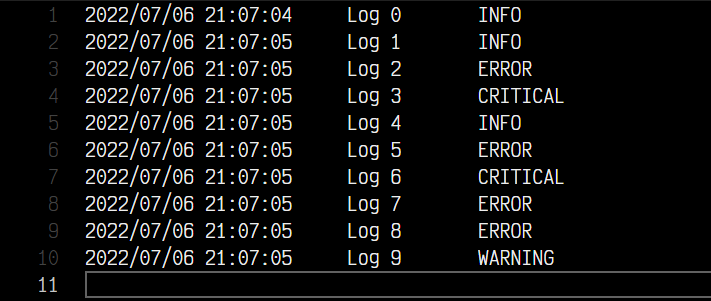
}

customLogger.close();

}

}

**Output:**



# Practical 3

**Aim:** Create a Java file to search for files in a given directory.

## Source Code:

import java.io.File;

import java.util.Scanner;

public class DirectorySearcher {

priva e String directoryPath;

/\*\*

* @param di ectoryPath Absolu e path of t e di ectory
* C ea es a di ectorySearc er object with a speci ied di ectory path.

\*/

public DirectorySearcher(String directoryPath) {

this.directoryPath = directoryPath;

}

/\*\*

* @param il er F l er to be appl ed
* Searc es t e di ectory for i ena es starting with given
* il er.Igno es subdi ector es.

\*/

public void search(String filter) { F le file = ew F le(directoryPath); F le[] fileArray = file.listFiles();

for (F le file2 : fileArray) { if (file2.isDirectory()) { contin e;

}

if (file2.getName().startsWith(filter)) {

Sys em.out.println(file2.getName());

}

}

}

/\*\*

* @param args Command li e argu ents
* Driver co e.

\*/

public static void main(String[] args) {

Scan er scanner = ew Scan er(Sys em.in);

Sys em.out.println("Enter a directory > ");

String directoryPath = scanner.nextLine();

Di ectorySearc er directorySearcher = ew Di ectorySearc er(directoryPath);

Sys em.out.println("Enter filter > ");

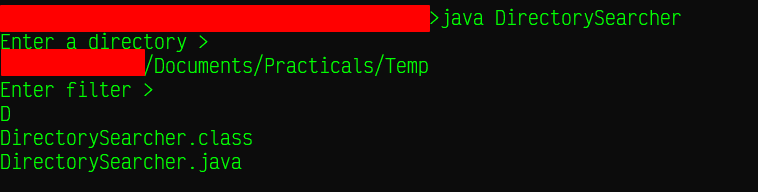
String filter = scanner.nextLine(); directorySearcher.search(filter);

scanner.close();

}

}

**Output:**



# Practical 4

**Aim:** Create a Java file to search for files in a given directory.

## Source Code:

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class FileSearcher {

priva e String absFileName;

public FileSearcher(String absFileName) {

this.absFileName = absFileName;

}

public boo ean search(String word) {

boo ean found = f l e;

try {

F le file = ew F le(absFileName); Scan er scanner = ew Scan er(file);

wh le (scanner.hasNext()) { if(scanner.nextLine().indexOf(word) = -1) { found = tr e;

}

}

scanner.close();

} catch (F leNotFoundException e) {

Sys em.out.println("File not found.");

e.printStackTrace();

}

eturn found;

}

public static void main(String[] args) {

Scan er scanner = ew Scan er(Sys em.in);

Sys em.out.println("Enter a file name > ");

String fileName = scanner.nextLine();

F leSearc er fileSearcher = ew F leSearc er(fileName);

Sys em.out.println("Enter a word filter > "); String word = scanner.nextLine(); scanner.close();

boo ean found = fileSearcher.search(word);

if (found) {

Sys em.out.println("Word found");

} el e {

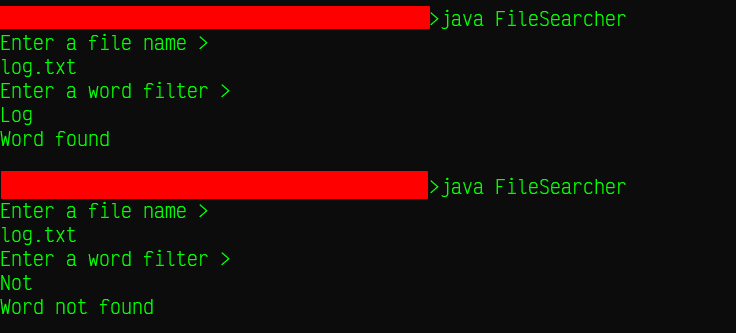
Sys em.out.println("Word not found");

}

}

}

**Output:**



# Practical 5

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

## Source Code:

import java.io.FileWriter;

import java.io.IOException;

public class VirusExample {

/\*\*

* @param args Command-li e argu ents.
* @throws IOException if i e cannot be ope ed.

\*

* C ea es a i e na ed i e1. ll in append mo e and epea e ly
* appends "Virus" into it.

\*/

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

F leWri er fileWriter = ew

F leWri er("D:/VirusFiles/file1.dll", tr e); wh le (tr e) {

fileWriter.write("Virus");

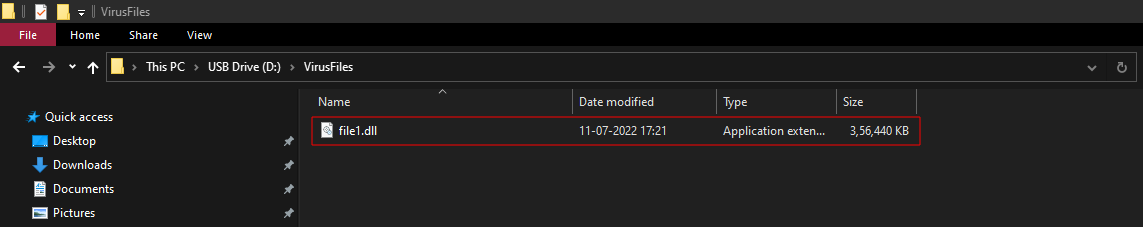
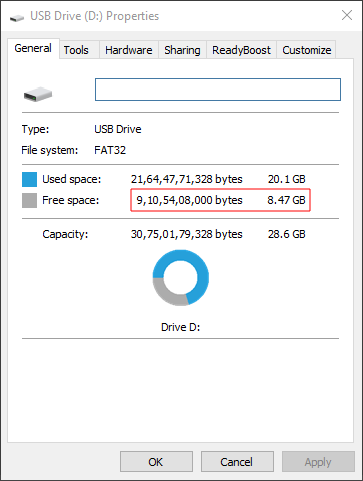
}

}

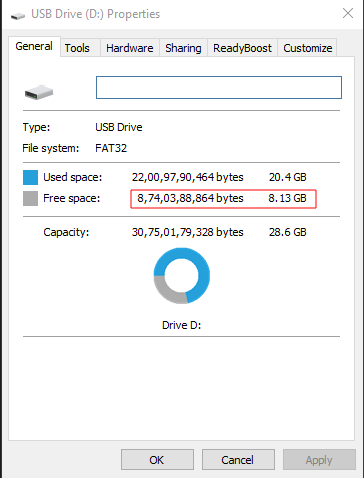
}

## Output:

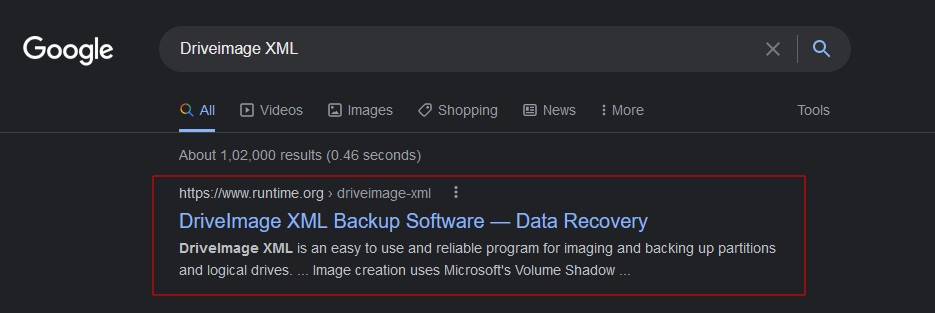
* Before:



* Generated file:
* After:



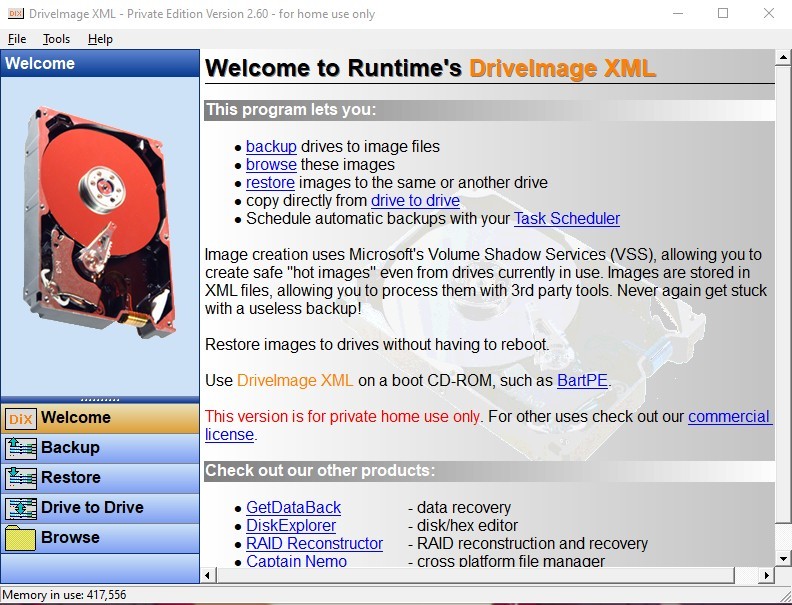
# Practical 6



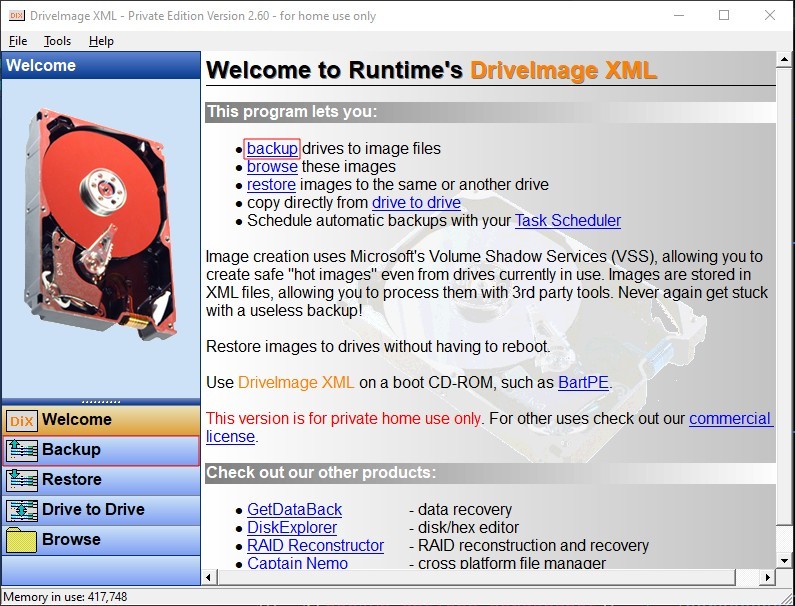
**Aim:** Create a backup of a disk using DriveImage XML.

## Procedure:

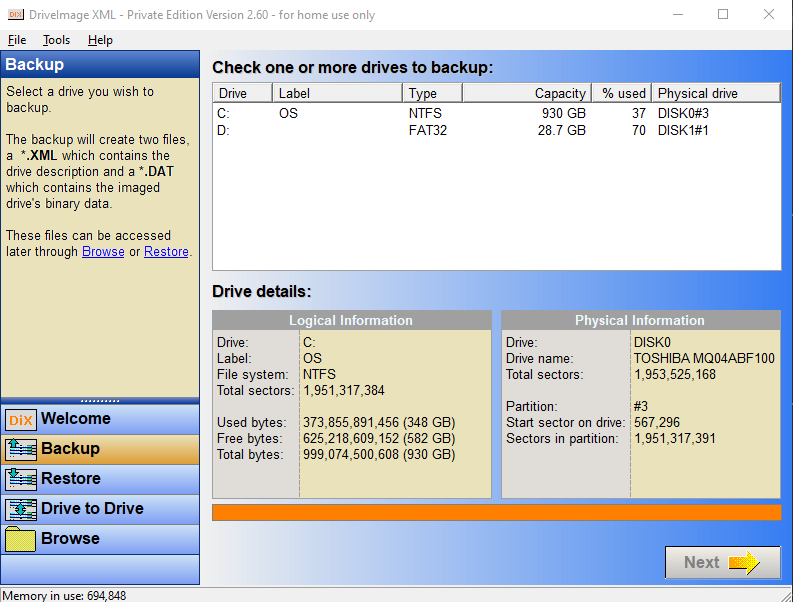
* Download and install DriveImage XML from this [link](https://www.runtime.org/driveimage-xml.htm). A quick web search should lead you to this website:
* After opening DriveImage XML, you will be presented with this screen:



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

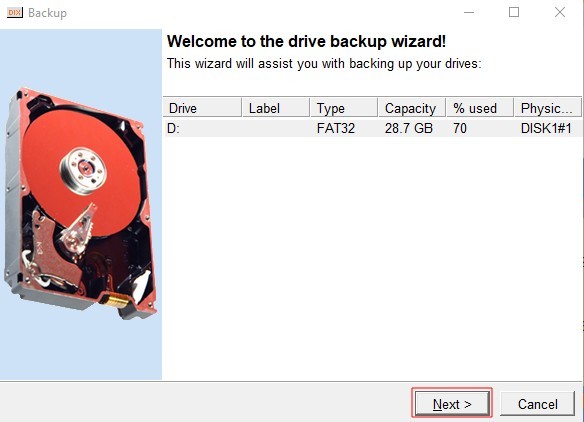
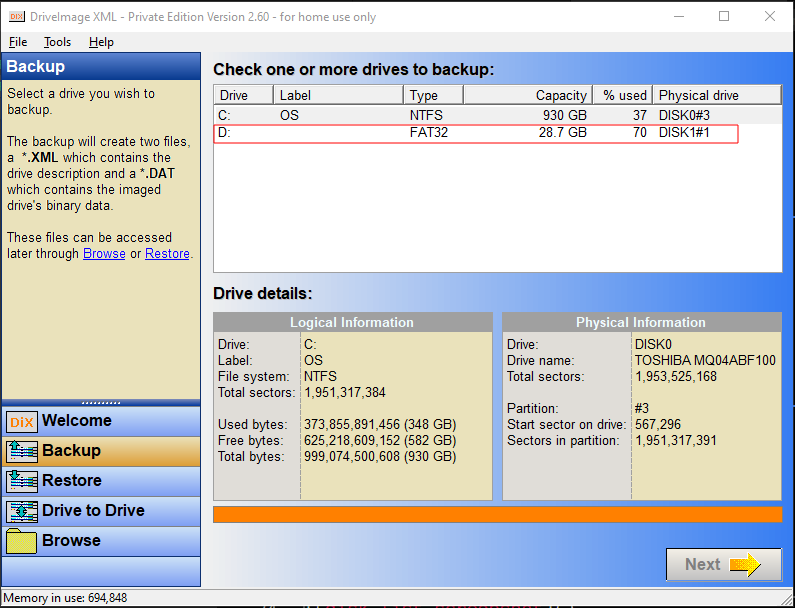


* After clicking on either of the two options listed above, it should show you a list of all the disk(s) present on your system:

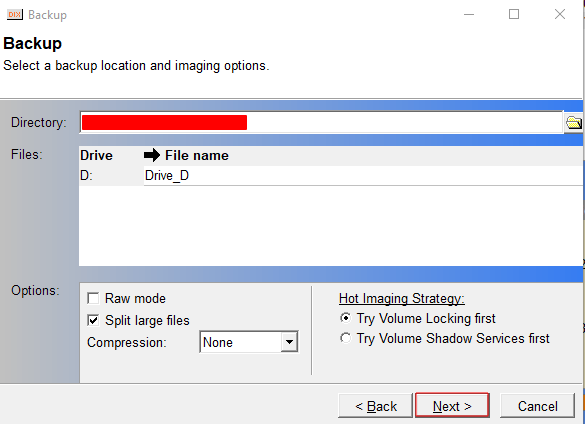


* Choose one (or multiple) disk(s) to image. In this exercise, Disk D is chosen for creating a backup. After clicking on "Next", the

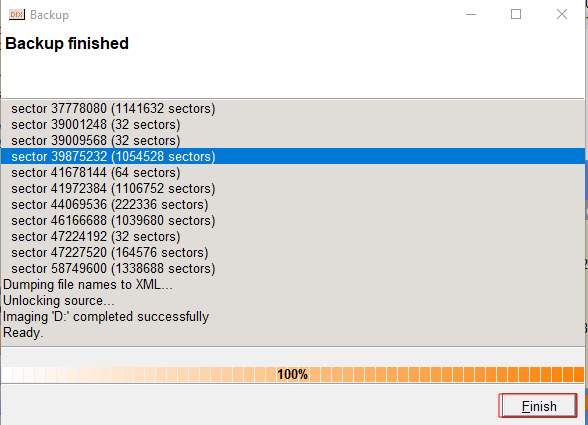
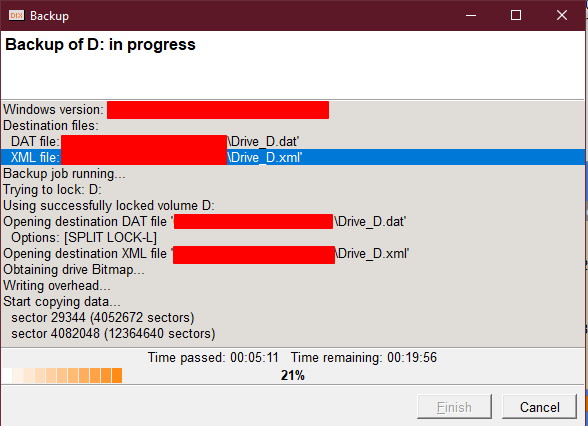
Backup wizard will be displayed. After confirming your selection, click on Next:



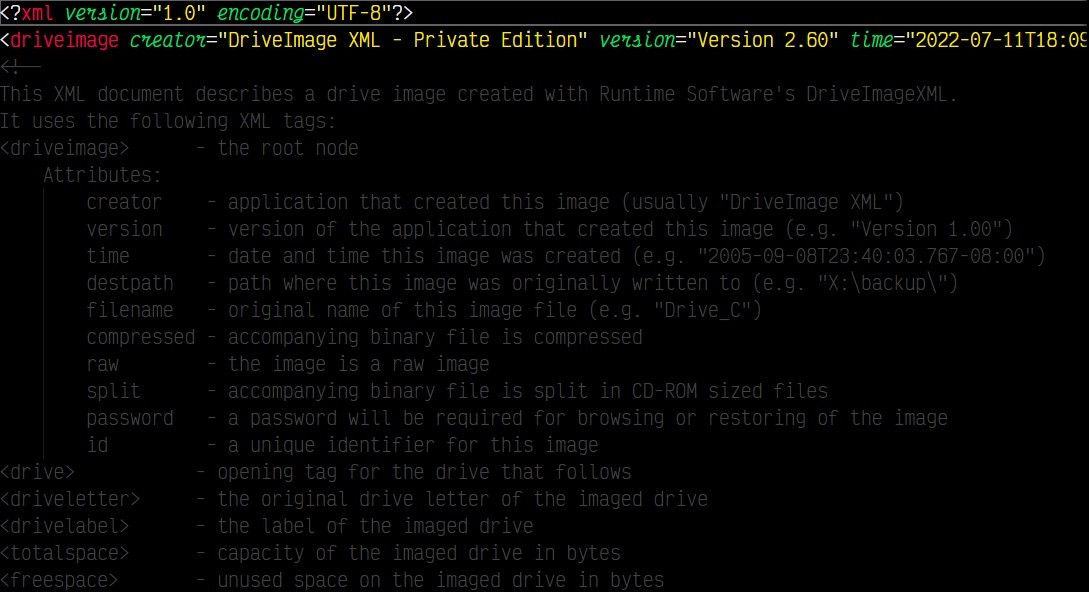
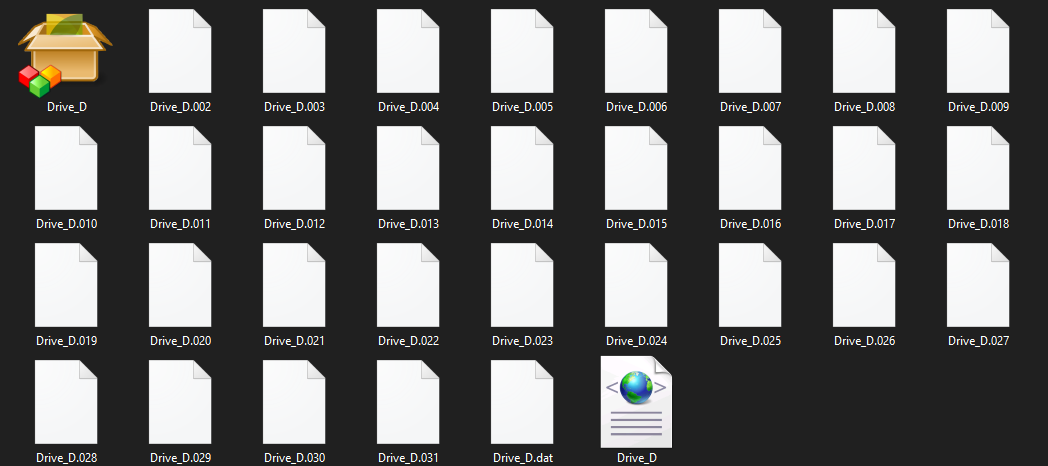
* Confirm other details such as Output location and other settings and when comfortable, click on Next.



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

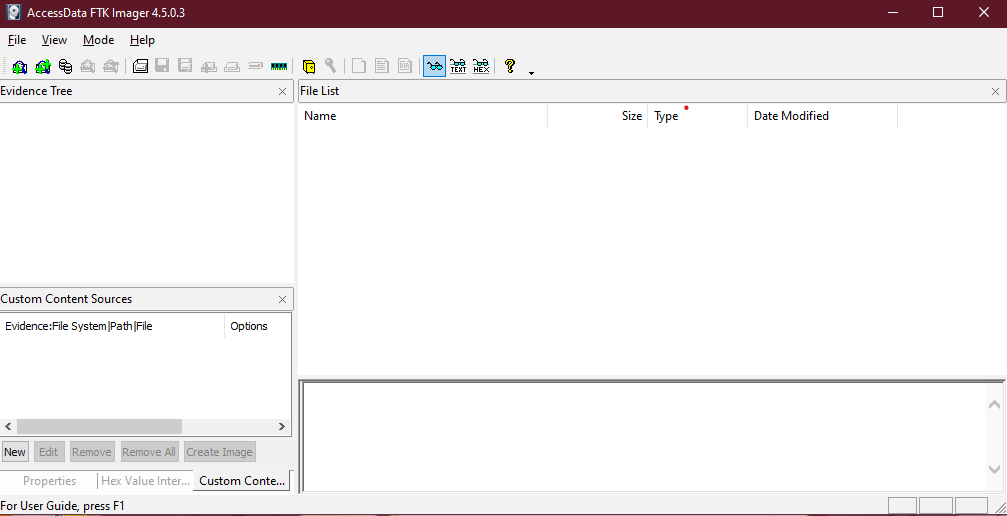


* The following files will be generated in the destination folder.



* The generated XML file has the following text:

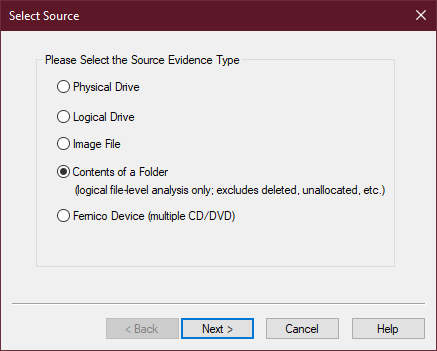
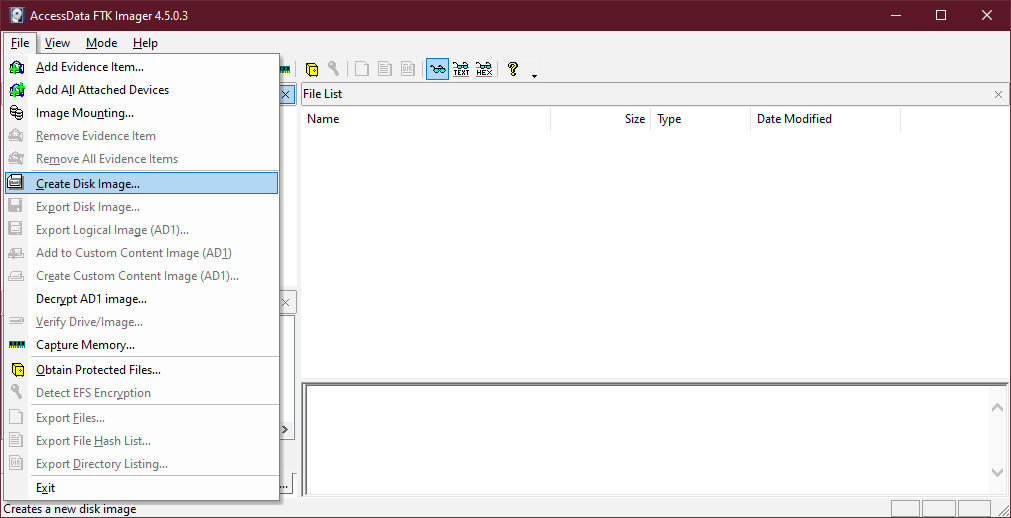
# Practical 7



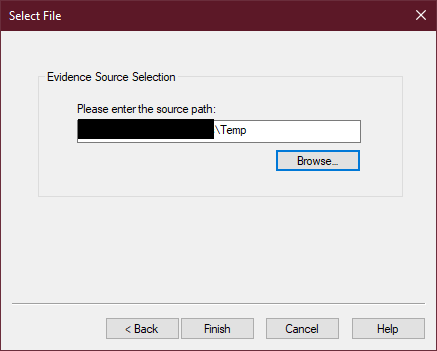
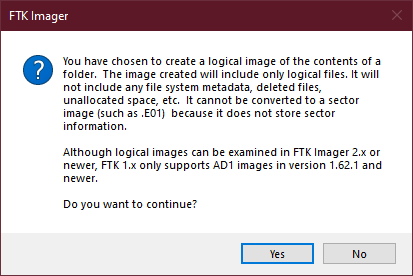
**Aim:** Create a forensic image of a digital device from volatile data such as memory.

## Procedure:

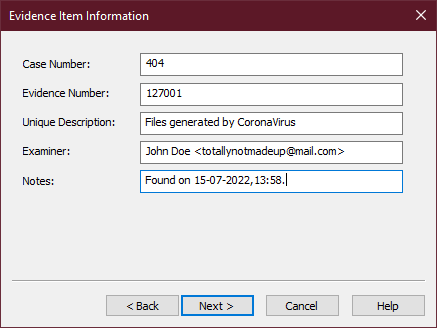
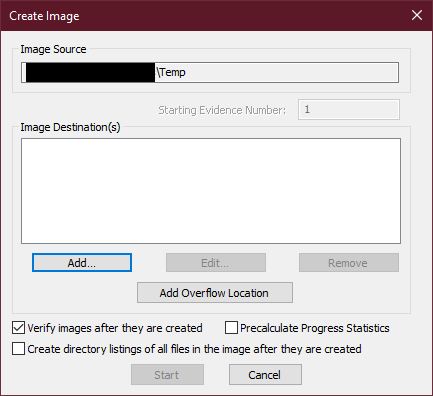
* Download and install AccessData® FTK® Imager from this [link](https://accessdata.com/product-download-page). Launching the application will display a screen similar to this:
* Now, navigate to File > Create Disk Image....



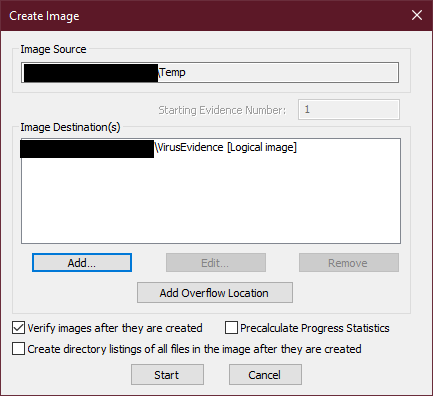
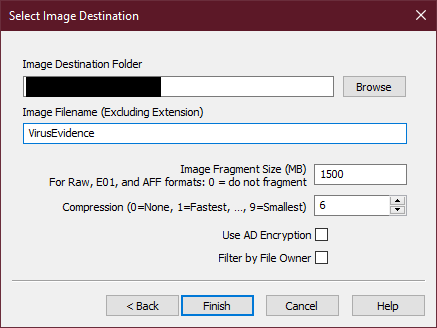
* This should bring up a new window. Select the Contents of a Folder option for the source. Click on Next.
* The generated warning window can be ignored. Simply click on Next.



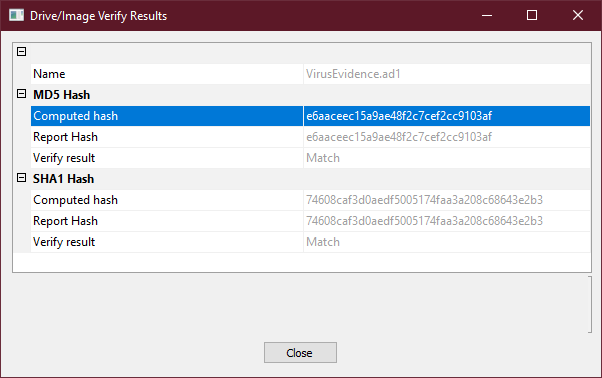
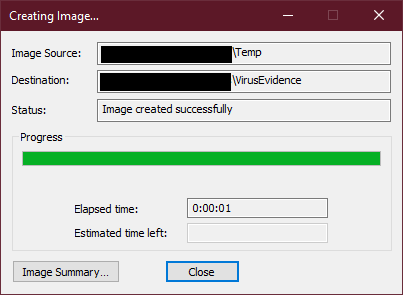
* The window will now ask for a source location. Enter the location of your choice and click on Finish.
* Now, a new dialog box will appear. Confirm your source selection and then click on the Add... to add a new destination.



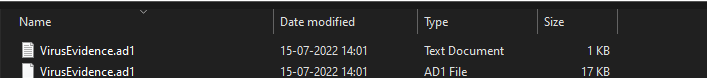
* A new window will appear which will ask for information about this particular item. Fill it and then click on Next.
* Select the destination of your choice and provide the filename of the (soon to be) generated image file(s). Click on Finish.



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



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



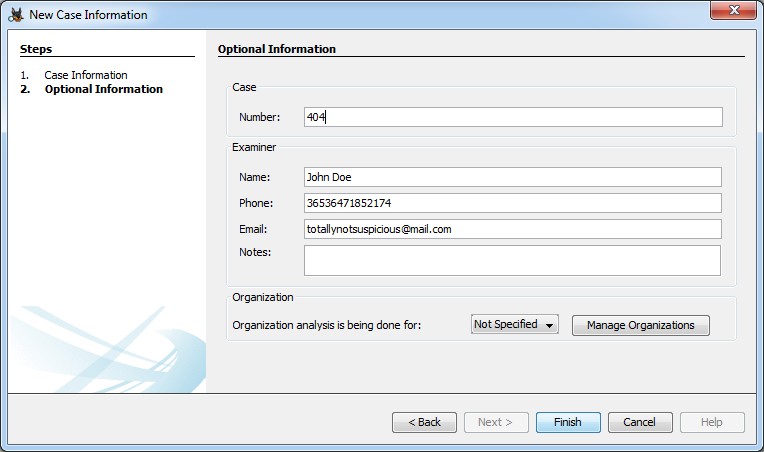
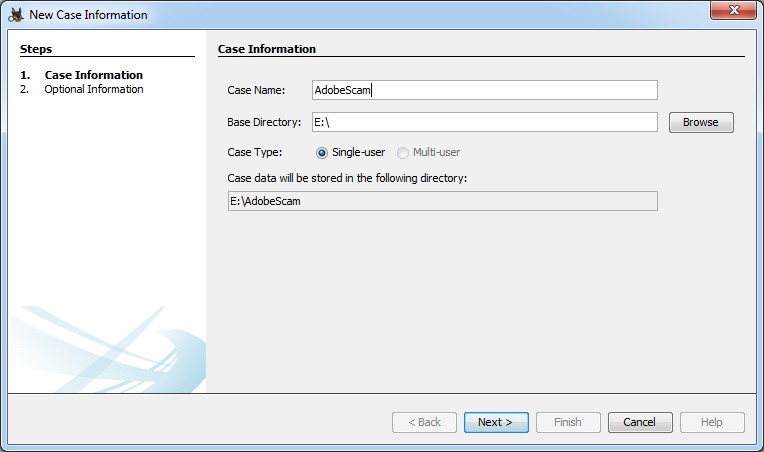
# Practical 8



**Aim:** Retrieve deleted files from a computer.

## Procedure:

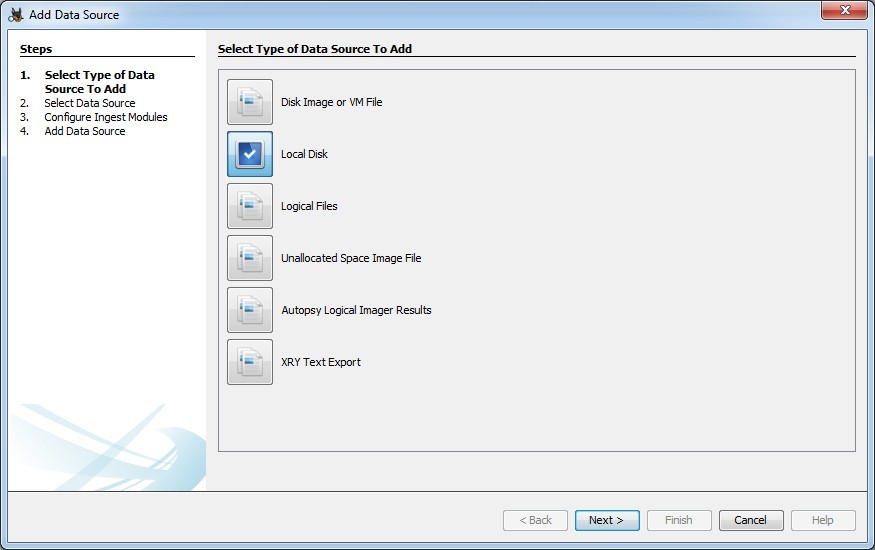
* Download and install Autopsy® from this [link](https://www.autopsy.com/download/). Running the application should present you this window:
* Click on New Case. It should present you this window asking for case name and the directory to store case-related data.



* 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

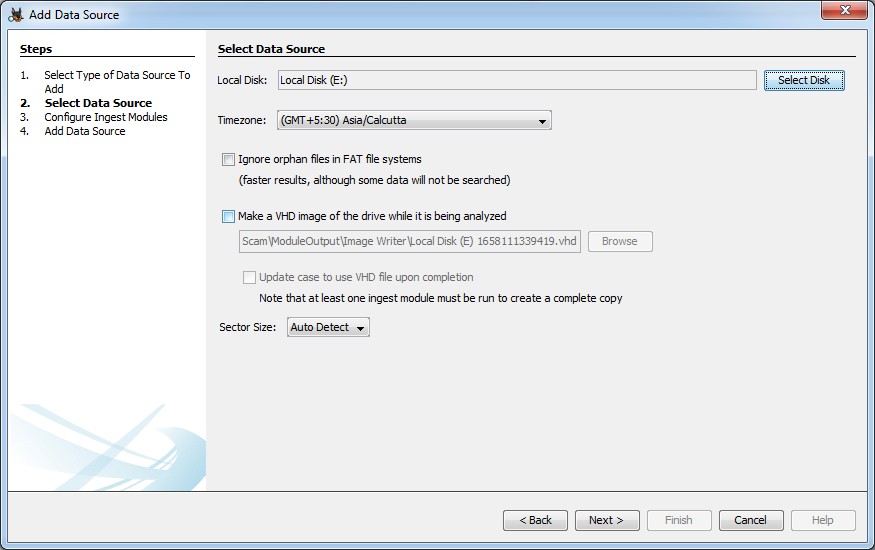
Finish when you're done.

* 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

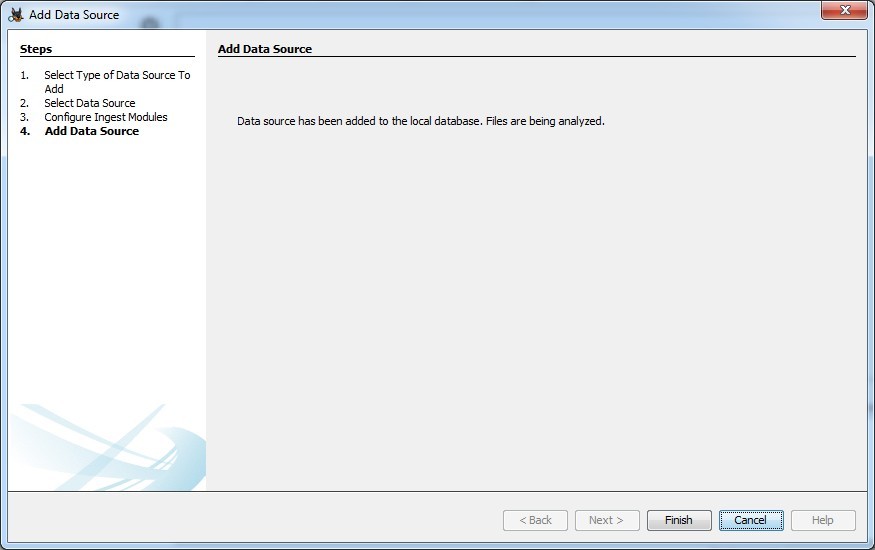
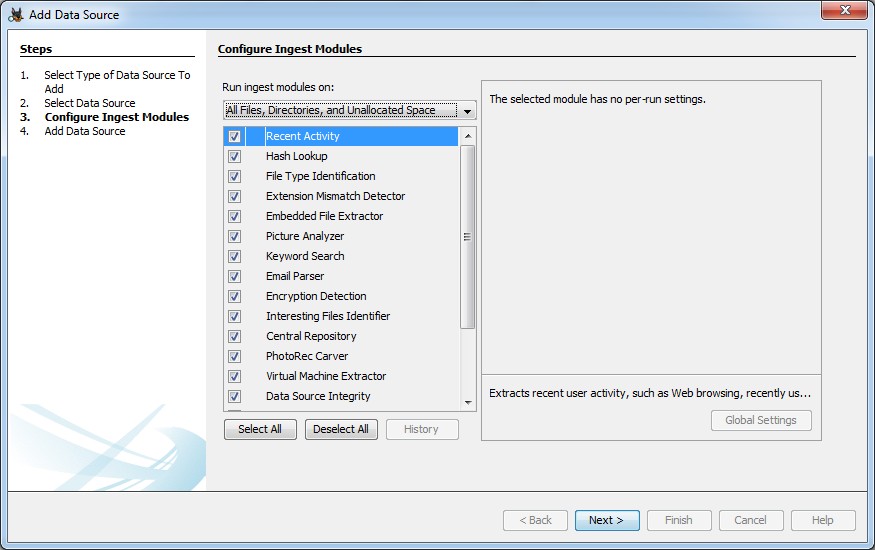


to be added and click on Next.

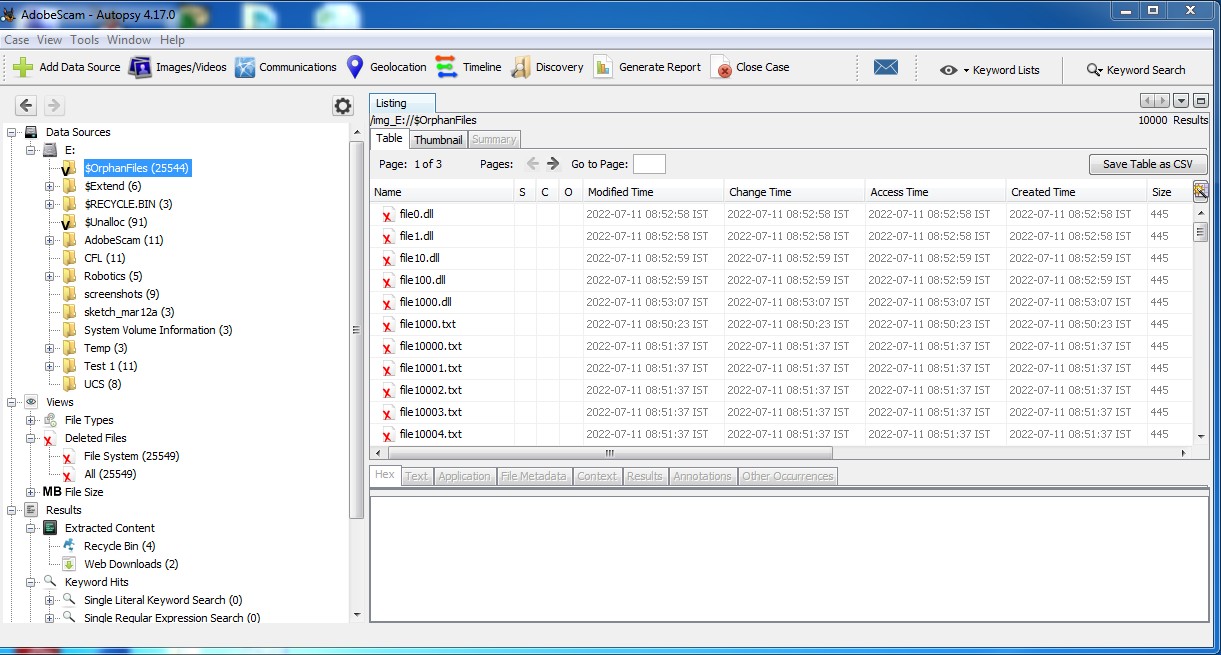
* A new section named Select Data Source should now be active. Select the disk of your choice and click on Next.



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

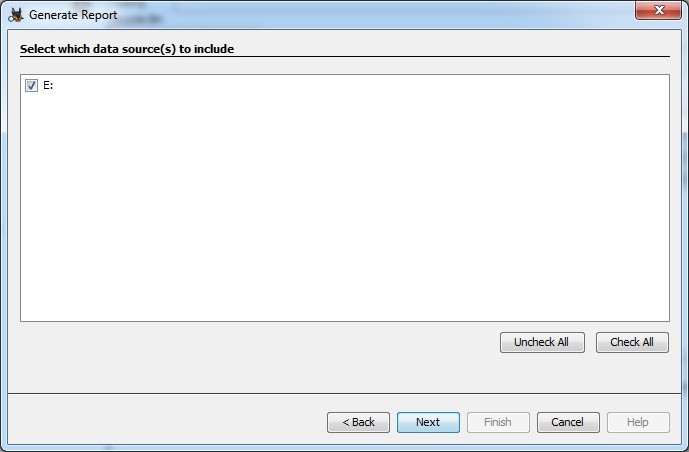


* Autopsy® will now try to process the data source. This process may take some time depending on the size of the disk and its contents. After completion, you will see all the information it has

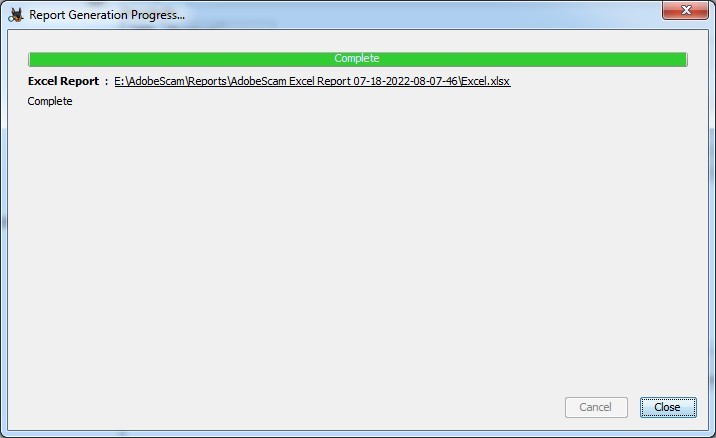
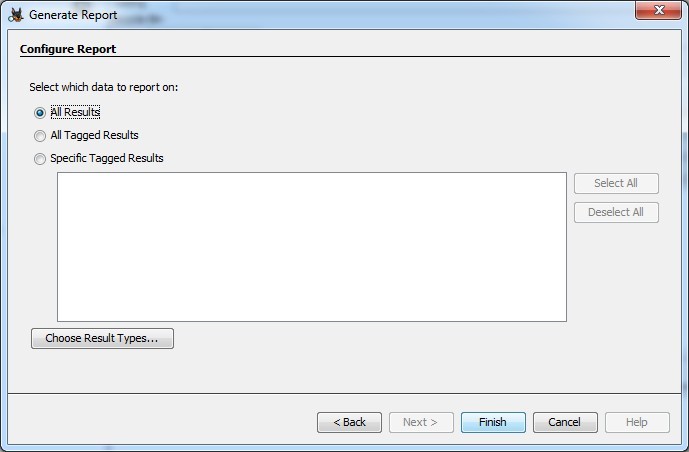


gathered ordered as a tree. Now, navigate to Data Sources > {Disk of your choice} > $OrphanFiles. It will show all the deleted files. You can retrieve it by right clicking the file(s) and selecting Export. It will ask for a location to restore the file.

* 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.



* Select the data sources to be included and click on Next.
* Select the data which should be reported and click on Finish. The report will be generated.



# Practical 9

**Aim:** Use the registry to obtain information.

## Theory:

The Windows Registry Editor(regedit) was launched in 1992 with Microsoft Windows 3.1. The registry is the backbone of the OS and is critical for system performance. It enables administrators and advanced users to keep the registry operational and make root and administrative level changes such as setting up access permissions or changing the hardware and software level configuration.

## Features:

1. **System Performance:**
   * If a key inside the registry becomes corrupt or faulty, it can cause system to crash or other performance issues.
   * Using Registry Editor we can edit/update the key.

## Configuration settings:

* + The automatic type startup programs display or desktop setting can be configured using regedit.

## Registry cleaning:

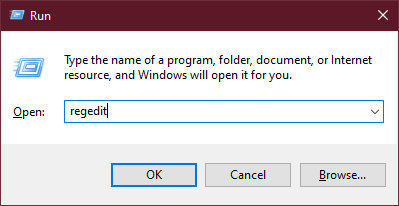
* + Entries inside the registry can sometimes break. To fix broken entries, a registry cleaner is required.
  + Unlike standard configuration files, entries inside the Registry cannot be opened or cleaned via standard text editor.

## Registry errors:

* + Certain events can disturb the hierarchy and cause errors.
  + The regedit tool can be used to fix the hierarchical structure

of the registry.

## Finding Strings:



* + regedit can be helpful when searching for specific strings in keys, values (names & values).

## Remote editing of registry:

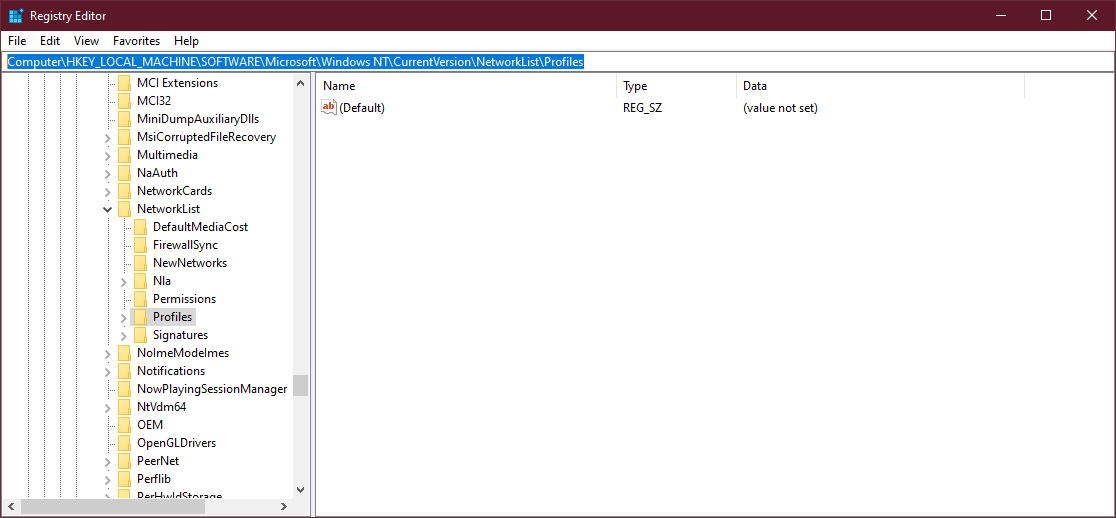
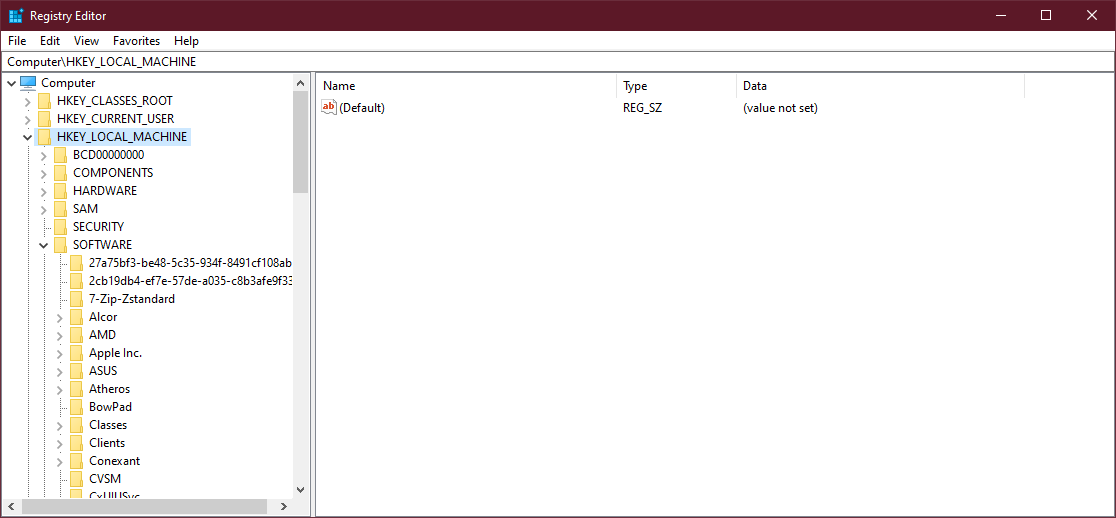
* + regedit can be used for remote editing of another computer's registry on the same network.

## Modification of key:

* + Registry key can be modified, renamed or deleted by regedit.

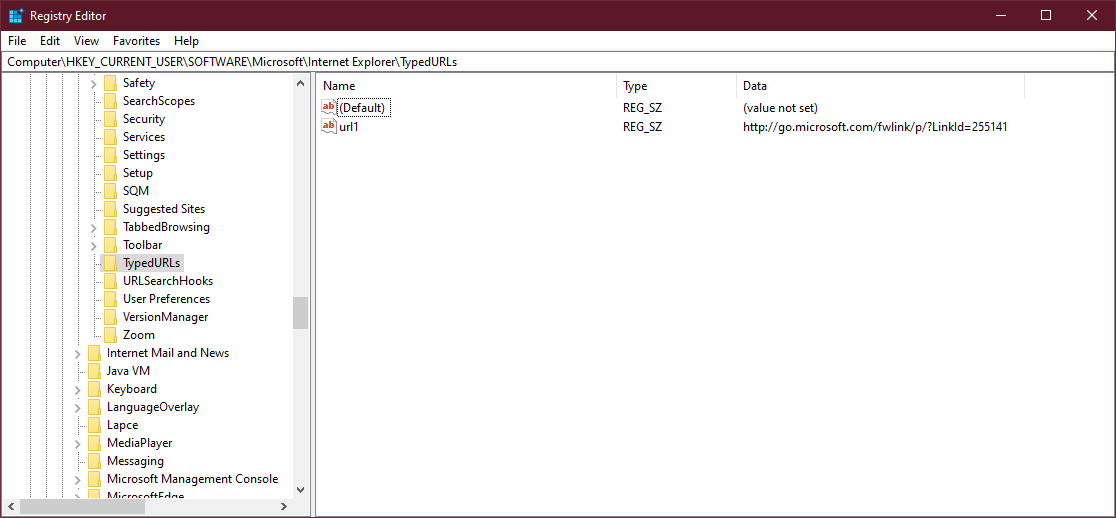
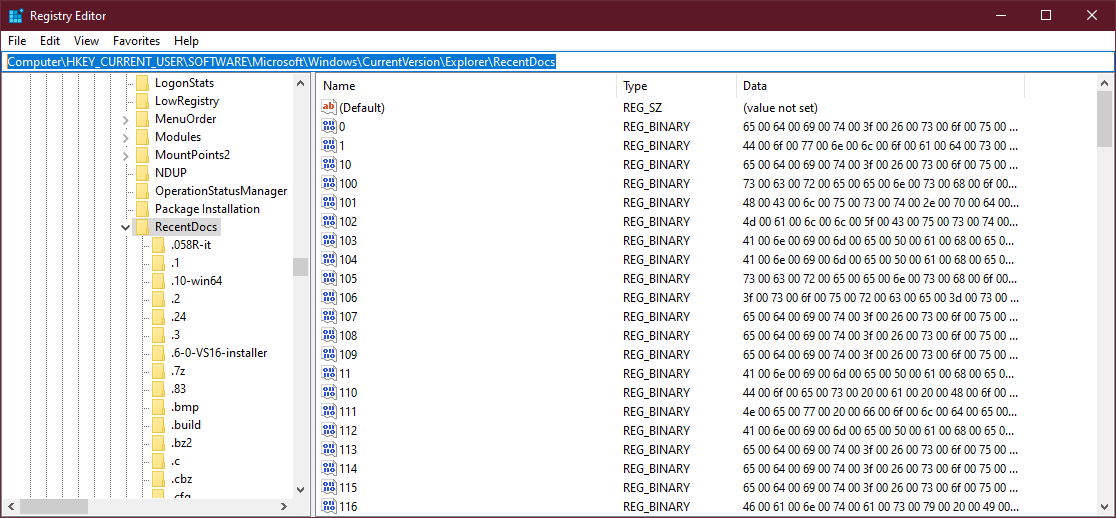
## Procedure:

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

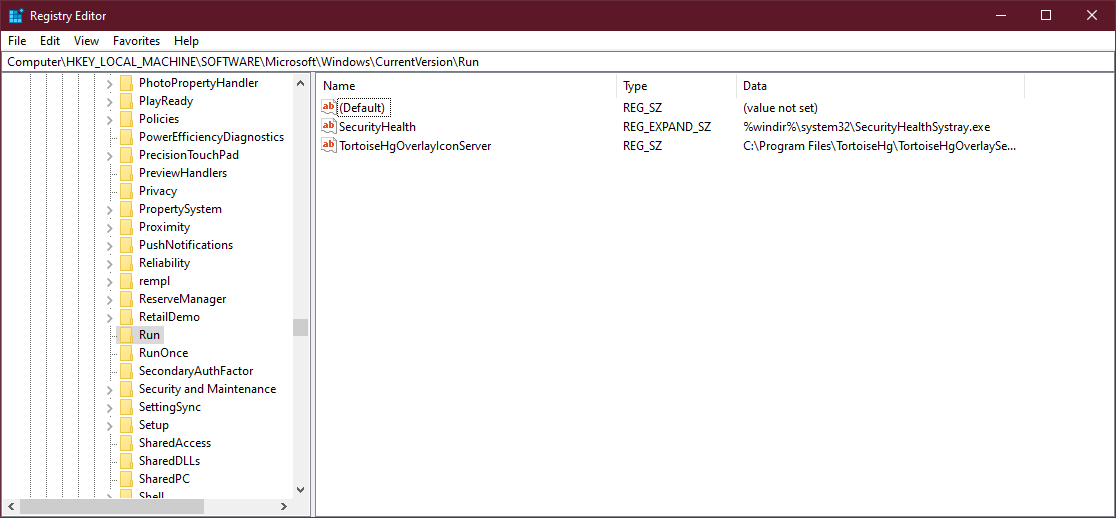
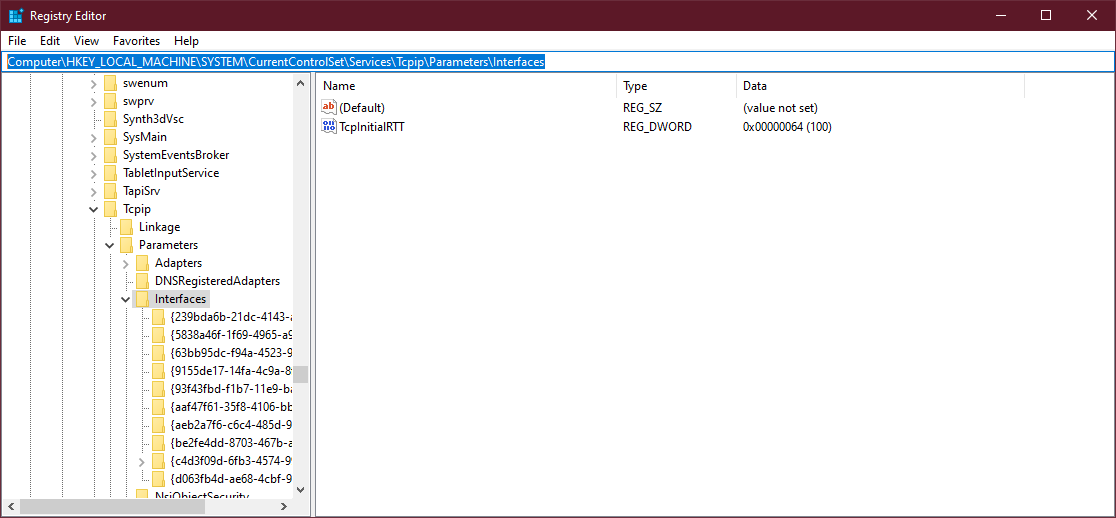


## Locations:

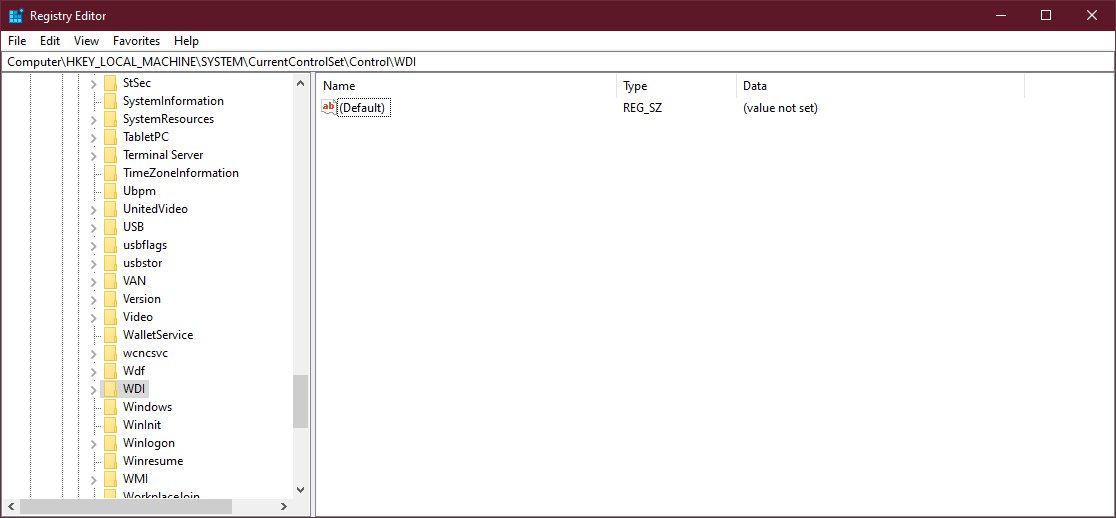
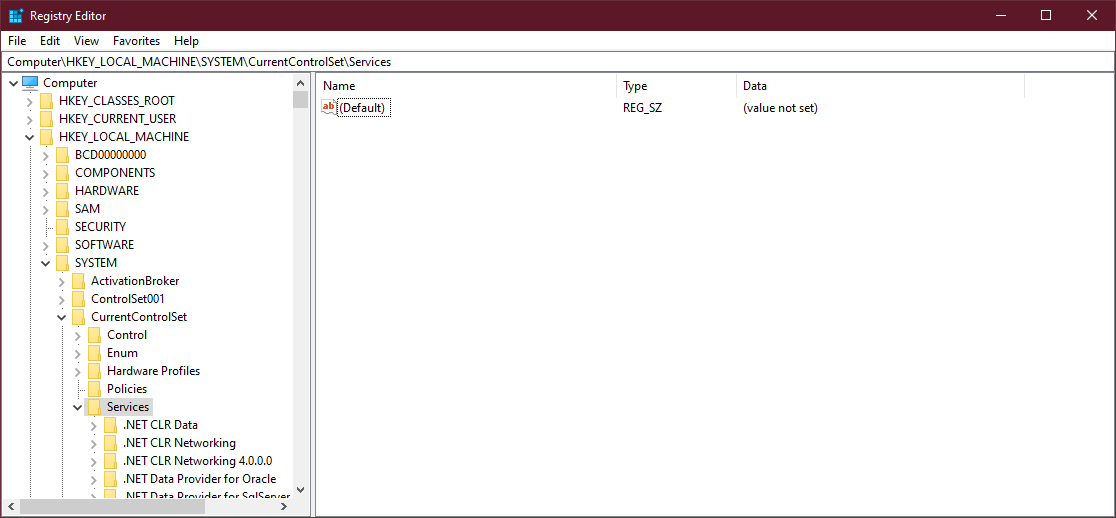
* Wireless Evidences: Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\ Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles
* Recent Documents key: Computer\HKEY\_CURRENT\_USER\SOFTWARE\ Microsoft\Windows\CurrentVersion\Explorer\RecentDocs



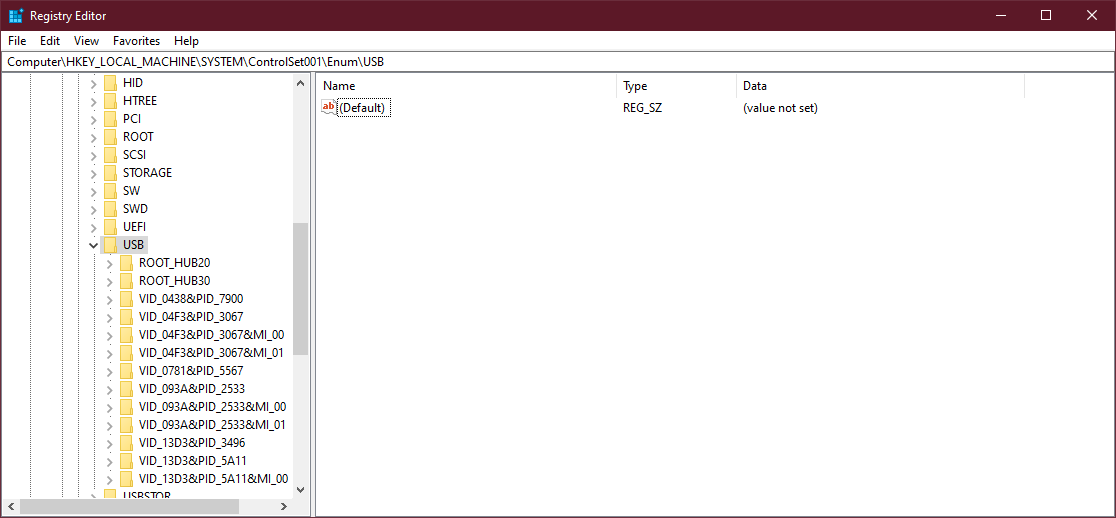
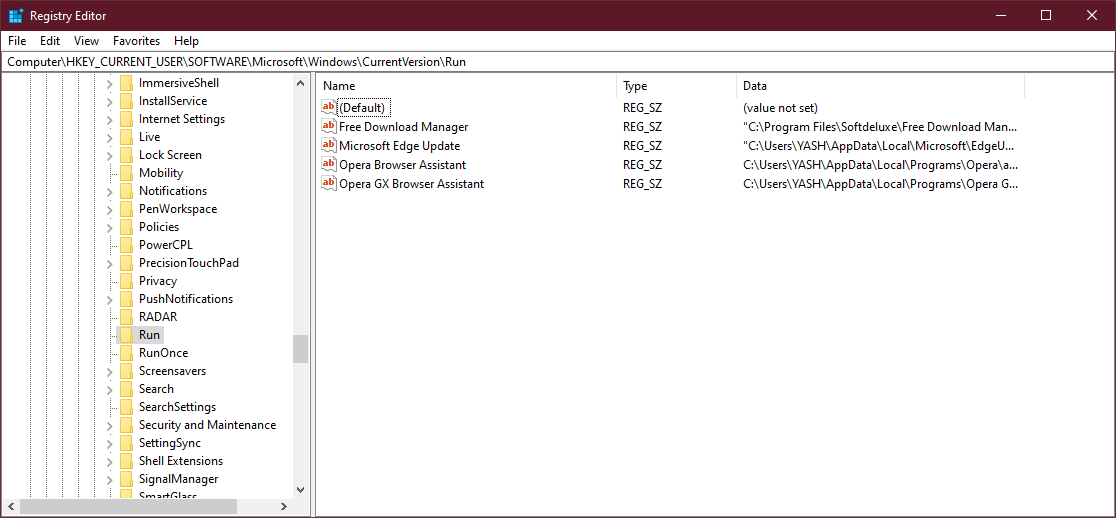
* Typed URLs key: Computer\HKEY\_CURRENT\_USER\SOFTWARE\ Microsoft\Internet Explorer\TypedURLs
* IP address: Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\ CurrentControlSet\Services\Tcpip\Parameters\Interfaces



* 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

