1. Running application program

**import** java.io.\*;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**import** java.util.Calendar;

**import** java.util.StringTokenizer;

**public** **class** running\_processes

{

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

{

//database connection

String url = "jdbc:mysql://localhost:3306/sd";

String user = "root";

String password = "";

// Setup the connection with the DB

Connection conn = DriverManager.*getConnection*(url, user, password);

//accepts input parameters at runtime.

PreparedStatement statement;

// Formatting a date into String based on specific locale provided as input.

DateFormat dateFormat = **new** SimpleDateFormat("dd-MM-yyyy ");

//get current date where the Date is a class

//To create a new object of Date class by using the new keyword at the time of declaring the variable.

Date date = **new** Date();

String date1= dateFormat.format(date);

// Calendar is an abstract class.

//Calendar.getInstance() returns an implementation class java.util.GregorianCalendar.

Calendar cal = Calendar.*getInstance*();

// SimpleDateFormat class comes with a variety of choices for formatting dates and times.

SimpleDateFormat sdf = **new** SimpleDateFormat("HH:mm:ss");

String time1= sdf.format(cal.getTime()) ;

String a[]=**new** String[10];

//Cmd command to get the list of all running application

String[] test={"tasklist","/FO","csv"};

// ProcessBuilder instance manages a collection of process attributes.

ProcessBuilder builder = **new** ProcessBuilder(test);

// redirectErrorStream() method tells whether process builder merges standard error and standard output.

builder.redirectErrorStream(**true**);

Process p = builder.start();

BufferedReader r = **new** BufferedReader(**new** InputStreamReader(p.getInputStream()));

String line;

**int** i=0;

**while** (**true**) {

line = r.readLine();

//Break String into Tokens.

StringTokenizer st = **new** StringTokenizer(line,",");

i=0;

//check whether there are more tokens within the tokenizer's string.

**while** (st.hasMoreTokens() && i<4)

{

// return the next token

a[i]=st.nextToken();

i++;

System.***out***.println(a[i]);

System.***out***.println(i);

String sql = "INSERT INTO rp (NAME,PID,SESSION,SESSIONH,TIME,DATE)Values "

+ "('"+a[0]+"','"+a[1]+"','"+a[2]+"','"+a[3]+"','"+time1+"','"+date1+"')";

// PreparedStatements can use variables and are more efficient

statement = conn.prepareStatement(sql);

//Execute query

**int** row = statement.executeUpdate();

}

}

}

}

1. Installed application program

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.sql.\*;

**public** **class** application

{

**public** **static** **void** main(String[] args) **throws** IOException, SQLException

{

// Cmd command to fetch the installed application name

String[]test={"wmic","product","get","name"};

//Cmd command to fetch the version name of a particular application

String[]test1={"wmic","product","get","version"};

// ProcessBuilder instance manages a collection of process attributes.

ProcessBuilder builder = **new** ProcessBuilder(test);

ProcessBuilder builder1 = **new** ProcessBuilder(test1);

// redirectErrorStream() method tells whether process builder merges standard error and standard output.

builder.redirectErrorStream(**true**);

builder1.redirectErrorStream(**true**);

Process p = builder.start();

Process p1 = builder1.start();

BufferedReader r = **new** BufferedReader(**new** InputStreamReader(p.getInputStream()));

BufferedReader r1 = **new** BufferedReader(**new** InputStreamReader(p1.getInputStream()));

// strings to take the application name and version

String line,line1;

//Database connection

String url = "jdbc:mysql://localhost:3306/sd";

String user = "root";

String password = "";

// Setup the connection with the DB

Connection conn = DriverManager.*getConnection*(url, user, password);

**while** (**true**)

{

// read the data coming with the help of the process builder class

line = r.readLine();

line1=r1.readLine();

//checking if the data has finished or not

**if** ((line == **null**)&&(line1==**null**))

{

**break**;

}

//else continue writing it into the database

System.***out***.print(line +"--");

System.***out***.println(line1);

String sql = "INSERT INTO ds (Name, Version)Values ('"+line+"','"+line1+"')";

**try**

{

// PreparedStatements can use variables and are more efficient

PreparedStatement statement = conn.prepareStatement(sql);

//Execute query

**int** row = statement.executeUpdate();

**if** (row > 0)

{

System.***out***.println("uploaded");

}

}

**catch**(Exception e)

{

}

}

}

}