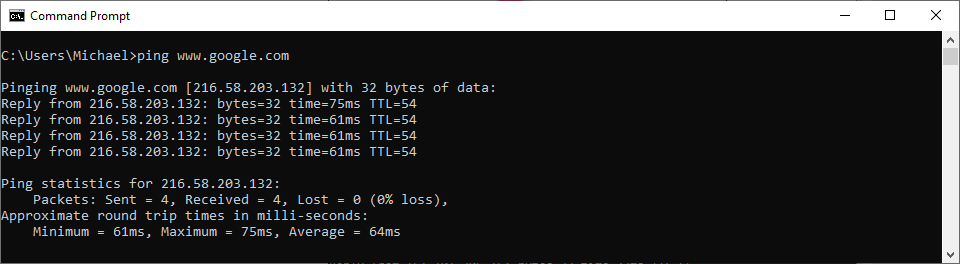
**Windows Network Diagnostic Commands**

Network Diagnostic Commands:

* ping
* ipconfig
* netstat
* tracert
* getmac
* hostname
* route print

­**ping**



**Syntax**  
 ping <target\_name>

**Output**  
 Ping time for each packet with ping time and TTL(Time To Live). After ping tests diagnostics is given of the connection.

Pings target\_name with 4 packets of data each of 32 bytes (By default).

­**ipconfig**

**Syntax**  
 ipconfig

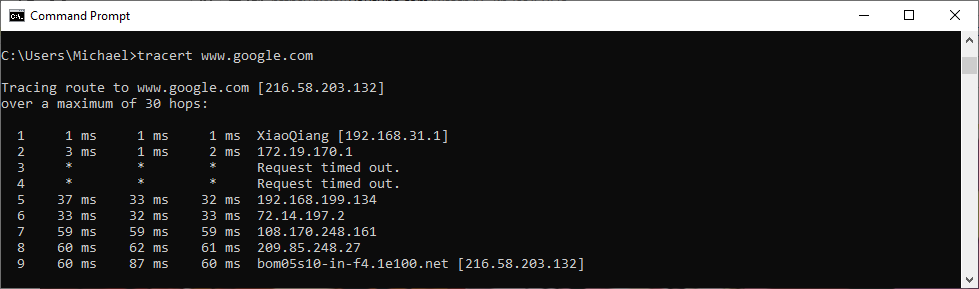
**Output**  
 Displays the network connections and displays their corresponding IP address, subnet mask and default gateway.

­**netstat**

**Syntax**  
 netstat

**Output**  
 Displays the active connections, with the Protocol, Listening address and Foreign address and its current state.

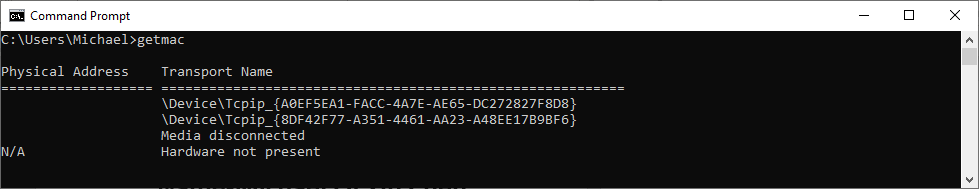
**tracert**



**Syntax**  
 tracert <target\_name>

**Output**  
 Sends a packet with a TTL of 1 and increments it by 1 on every subsequent transmission until destination responds or maximum TTL is reached.

**getmac**



**Syntax**  
 getmac

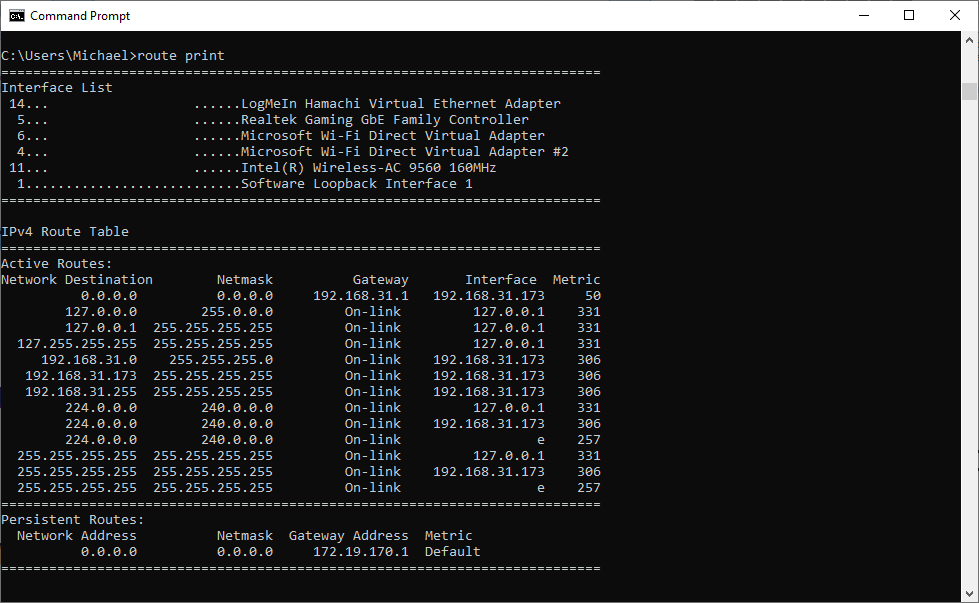
Output:  
 Displays the MAC addresses for all network adapters.

**hostname**

**Syntax**  
 hostname

**Output**  
 Displays the name of the current host.

**route print**



**Syntax**  
 route print

**Output**  
 Displays the local active and persistent IP routing table for IPv4 and IPv6. Displays Network Destination, Subnet Mask, Gateway, Interface and Metric associated with that route.

­**netstat State Descriptions**

|  |  |
| --- | --- |
| **State** | **Description** |
| CLOSED | Indicates that the server has received an ACK signal (to acknowledge receipt of a packet) from the client and the connection is closed. |
| CLOSE\_WAIT | Indicates that the server has received the first FIN signal (to acknowledge there is no more data to be sent) from the client and the connection is in the process of closing. |
| ESTABLISHED | Indicates that the server received the SYN signal (synchronize, this signal is only sent in the first packet) from the client and the session is established. |
| FIN\_WAIT\_1 | Indicates that the connection is still active but not currently being used. |
| FIN\_WAIT\_2 | Indicates that the client just received acknowledgement of the first FIN signal from the server. |
| LAST\_ACK | Indicates that the server is in the process of sending it's own FIN signal. |
| LISTENING | Indicates that the server is ready to accept a connection. |
| SYN\_RECEIVED | Indicates that the server just received a SYN signal from the client. |
| SYN\_SEND | Indicates that this connection is open and active. |
| TIME\_WAIT | Indicates that the client recognizes the connection as active, but not currently being used. |

**Socket**

* **Sever program**:

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author DELL

\*/

import java.io.\*;

import java.net.\*;

public class SocketDay1\_server {

public static void main(String argv[]) throws Exception

{

String clientSentence;

String capitalizedSentence;

ServerSocket skt = new ServerSocket(24150);

while(true)

{

Socket connectionSocket = skt.accept();

BufferedReader inFromClient = new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));

DataOutputStream outToClient = new DataOutputStream(connectionSocket.getOutputStream());

clientSentence = inFromClient.readLine();

System.out.println("Recieved and sending: " + clientSentence);

capitalizedSentence = clientSentence.toUpperCase()+'\n';

outToClient.writeBytes(capitalizedSentence);

}

}

}

* **Client program:**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author DELL

\*/

import java.io.\*;

import java.net.\*;

public class SocketDay1\_client {

public static void main(String argv[]) throws Exception

{

String sentence;

String modifiedSentence;

Socket clientSocket = new Socket("192.168.43.108",35999);

DataOutputStream outToServer = new DataOutputStream(clientSocket.getOutputStream());

BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

sentence = "Hi, from client";

outToServer.writeBytes(sentence+'\n');

modifiedSentence = inFromServer.readLine();

System.out.println("From server: " + modifiedSentence);

clientSocket.close();

}

}

* **Output:**

//Run time error

