TestPingCSharpV2

Software Documentation

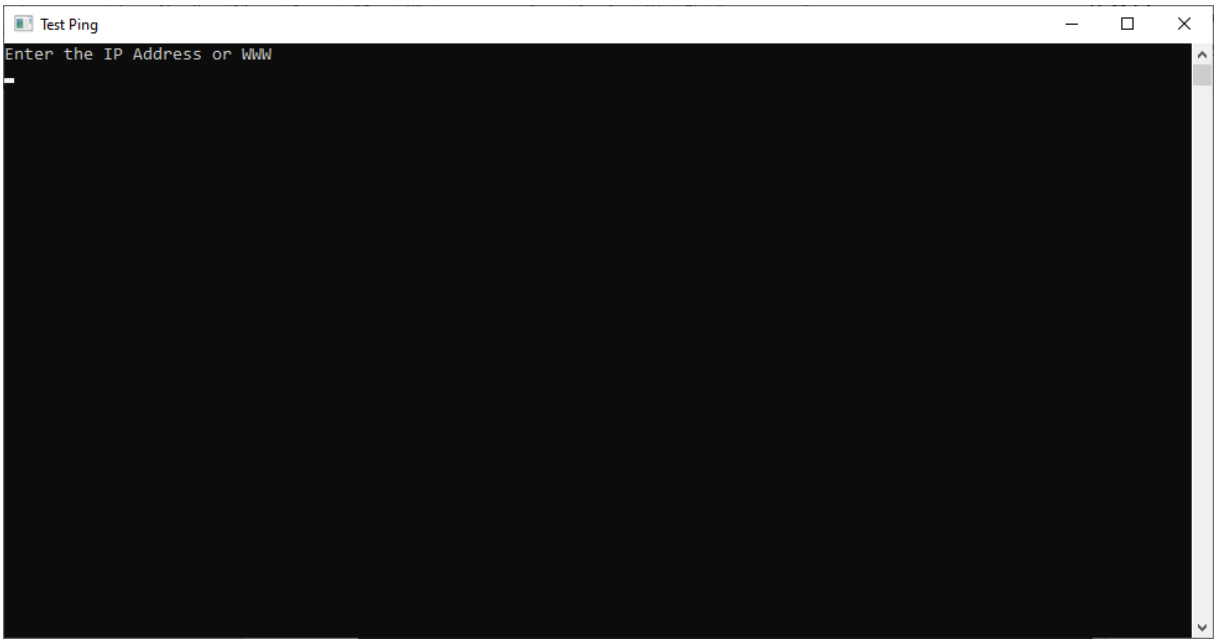
Author: matiwa

Table of contents

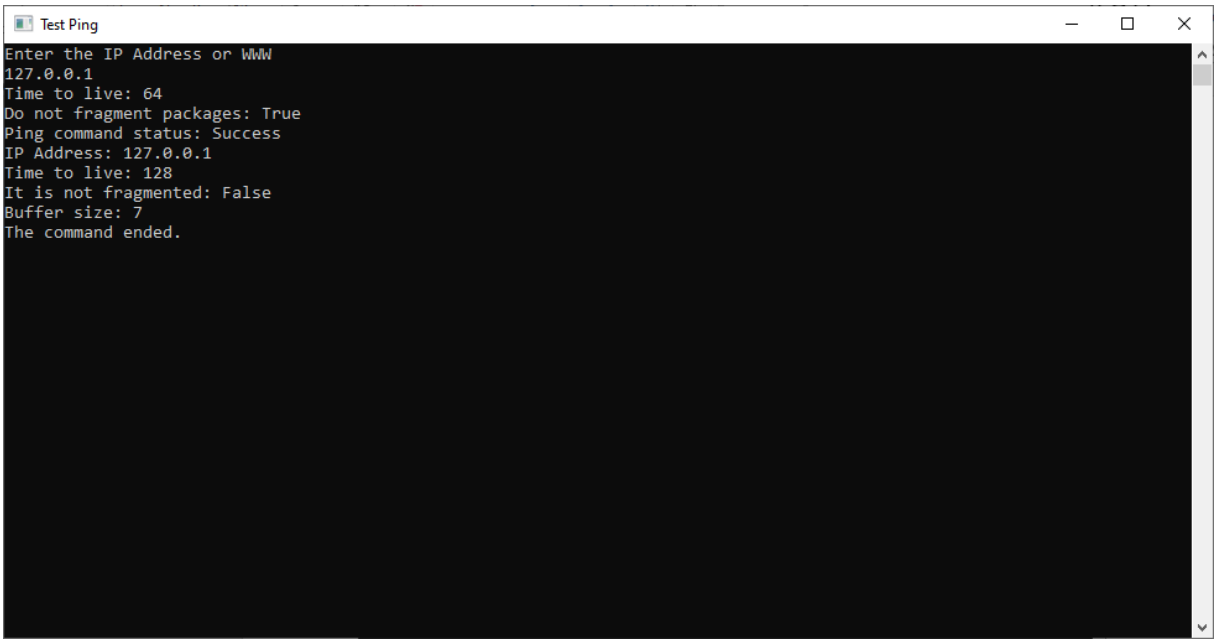
Introduction

This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. This application is used to diagnose network connections by Ping command.

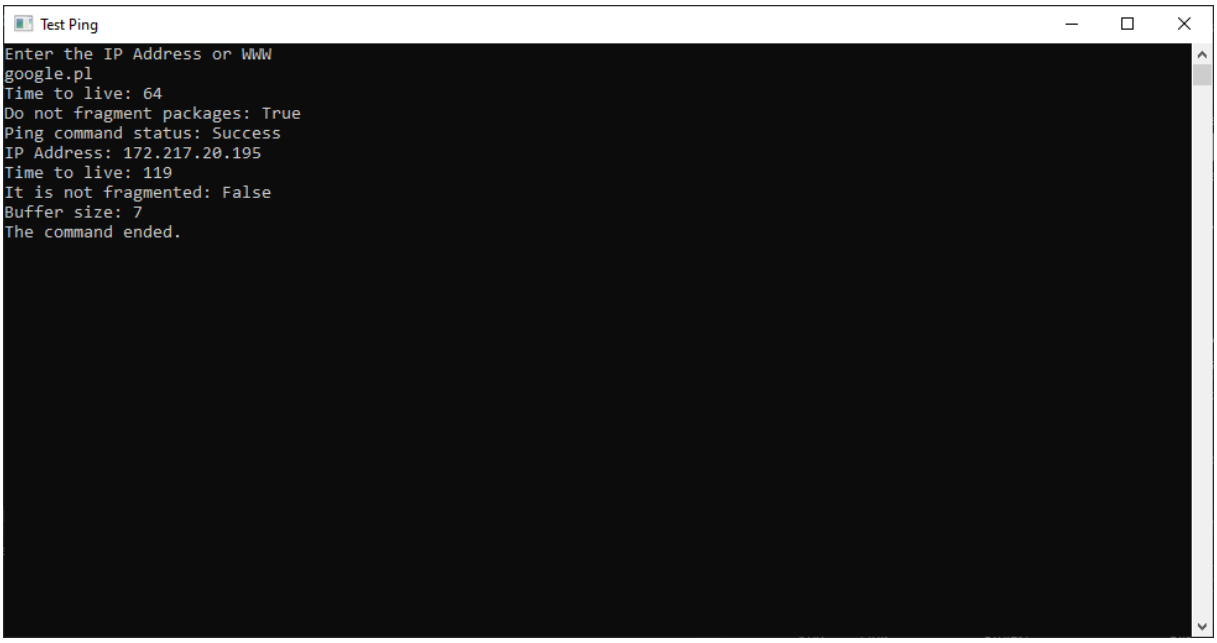
Describing of the application’s operation



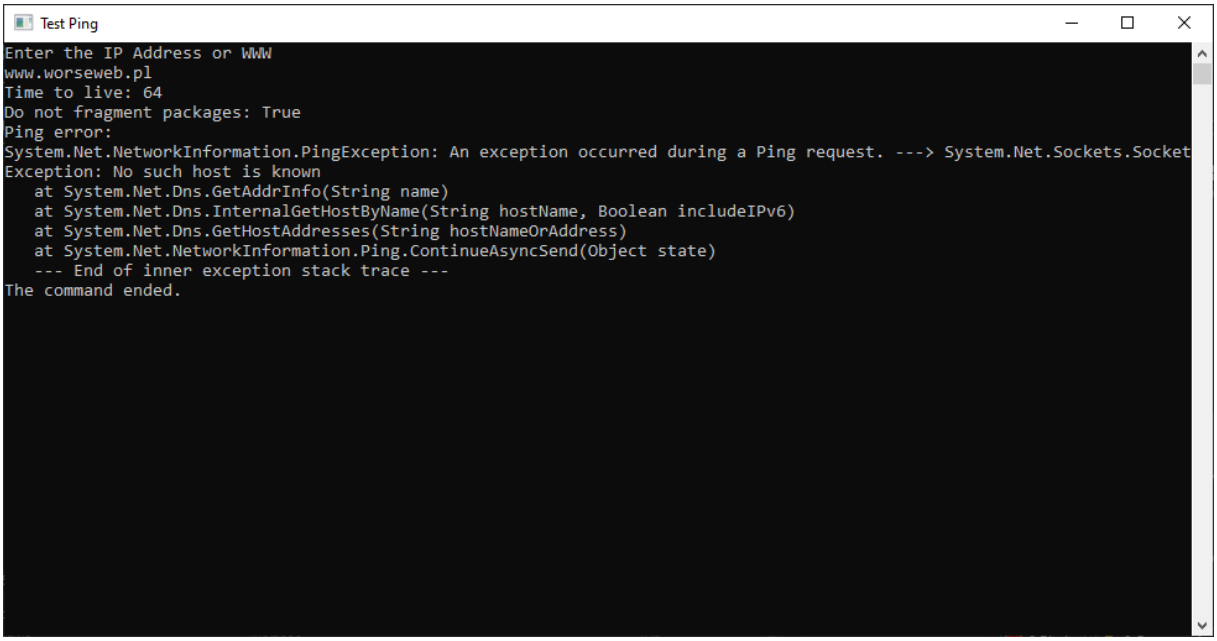
Drawing 1: The beginning of the application's operation [own study]



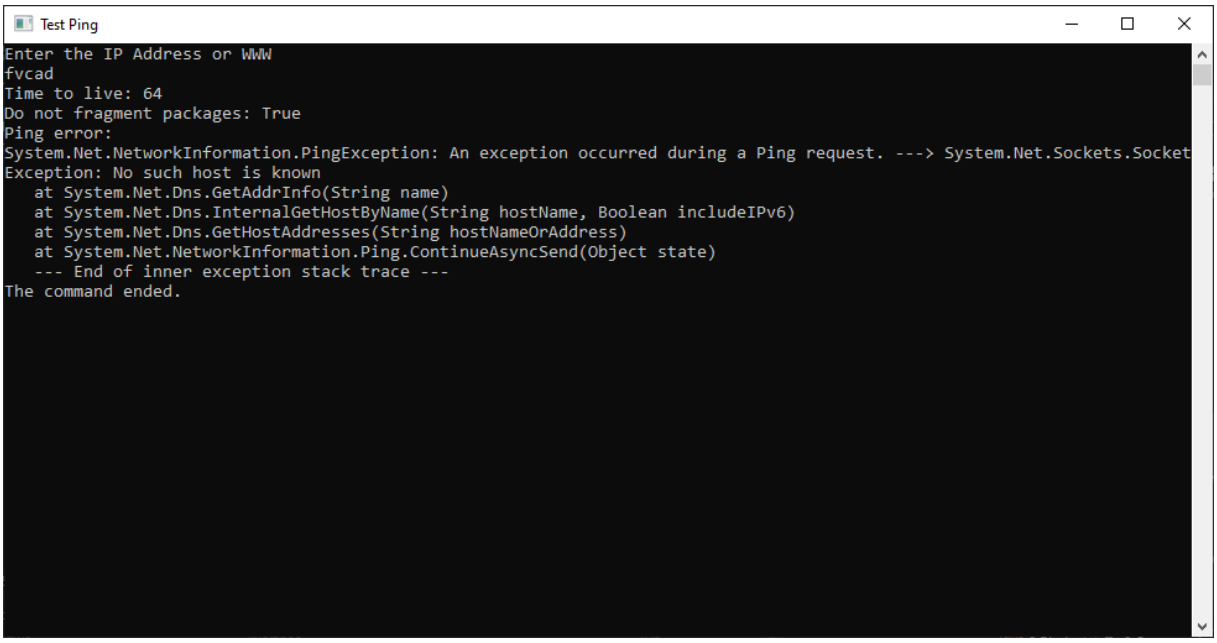
Drawing 2: The effect after entering the IPv4 address of the localhost[own study]



Drawing 3: The effect after entering the website [own study]



Drawing 4: Effect after entering a wrong address [own study]



Drawing 5: Effect after entering random characters [own study]

Drawing 6: Effect after typing nothing [own study]

After starting the program, the user is asked to enter an IP address or a website. Accepts the input by pressing the Enter key. If there is a network connection and the recipient information is correct, "Success" will be displayed next to "Ping command status". Otherwise, when there is a wrong address, random characters (not IP Address and WWW) or missing connection, there will be a message containing error details. It is possible that the user is staring at nothing and typing nothing, then the message "There is no IP Address or WWW" will appear. There are certainly errors that the developer did not discover while working on the application.

What is needed for use?

The application does not require installation. It only needs the Windows operating system.

Algorithm used

The basic form of the algorithm can be deduced from the previous section. It only needs the Windows operating system. In summary, the application is equivalent to the Ping command in the Windows Command Prompt.

Interface description

Drawing 7: Graphical interface [own study]

The interface is typical for a Console Application.

Source code description

The project was made in the C# programming language, in the Visual Studio Community 2017 programming environment. All work was done on the Windows 10 operating system. The application's source code looks like this.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Net;
```

```
using System.Net.NetworkInformation;

namespace TestPingCSharpV2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Title = "Test Ping";
            Console.WriteLine("Enter the IP Address or WWW");
            string adresWWW = Console.ReadLine();

            if (adresWWW.Length > 0)
            {
                try
                {
                    AutoResetEvent waiter = new AutoResetEvent(false);

                    Ping pingSender = new Ping();

                    pingSender.PingCompleted += new
PingCompletedEventHandler(PingCompletedCallback);

                    string data = "Message";
                    byte[] buforBajtow = Encoding.ASCII.GetBytes(data);

                    int timeout = 3600;

                    PingOptions opcje = new PingOptions(64, true);

                    Console.WriteLine("Time to live: " + opcje.Ttl.ToString());
                    Console.WriteLine("Do not fragment packages: " +
opcje.DontFragment.ToString());

                    pingSender.SendAsync(adresWWW, timeout, buforBajtow, opcje,
waiter);

                    waiter.WaitOne();
                    Console.WriteLine("The command ended.");
                }catch(Exception)
                {
                    Console.WriteLine("Wrong IP Address or WWW");
                }
            }
            else if (adresWWW.Length == 0) Console.WriteLine("There is no IP Address
or WWW");
            Console.ReadKey();
        }

        public static void PingCompletedCallback(object sender, PingCompletedEventArgs
e)
        {
            if (e.Cancelled)
            {
                Console.WriteLine("Cancelled Ping.");
                ((AutoResetEvent)e.UserState).Set();
            }

            if (e.Error != null)
            {
                Console.WriteLine("Ping error:");
                Console.WriteLine(e.Error.ToString());
```

```
                ((AutoResetEvent)e.UserState).Set();
            }

            PingReply odpowiedz = e.Reply;
            WyswietlOdpowiedzPing(odpowiedz);
            ((AutoResetEvent)e.UserState).Set();
        }

        public static void WyswietlOdpowiedzPing(PingReply odpowiedz)
        {
            if (odpowiedz == null) return;

            Console.WriteLine("Ping command status: "+odpowiedz.Status.ToString());
            if (odpowiedz.Status == IPStatus.Success)
            {
                Console.WriteLine("IP Address: " + odpowiedz.Address.ToString());
                Console.WriteLine("Time to live: " +
odpowiedz.Options.Ttl.ToString());
                Console.WriteLine("It is not fragmented: " +
odpowiedz.Options.DontFragment.ToString());
                Console.WriteLine("Buffer size: " +
odpowiedz.Buffer.Length.ToString());
            }
        }
    }
}
```

Listing 1: Source code [own study]

List of drawings

List of listings