

# TcpClient 类

.NET Framework 4

为 TCP 网络服务提供客户端连接。

## 继承层次结构

System.Object  
System.Net.Sockets.TcpClient

命名空间：[System.Net.Sockets](#)  
程序集：System ( 在 System.dll 中 )

## 语法

C#

```
public class TcpClient : IDisposable
```

TcpClient 类型公开以下成员。

## 构造函数

显示: ☒ 继承 ☒ 保护

	名称	说明
	<a href="#">TcpClient()</a>	初始化 TcpClient 类的新实例。
	<a href="#">TcpClient(AddressFamily)</a>	使用指定的族初始化 TcpClient 类的新实例。
	<a href="#">TcpClient(IPEndPoint)</a>	初始化 TcpClient 类的新实例，并将其绑定到指定的本地终结点。
	<a href="#">TcpClient(String, Int32)</a>	初始化 TcpClient 类的新实例并连接到指定主机上的指定端口。

# 属性

显示: ☒ 继承 ☒ 保护










	名称	说明
	Active	获取或设置一个值，该值指示是否已建立连接。
	Available	获取已经从网络接收且可供读取的数据量。
	Client	获取或设置基础 Socket。
	Connected	获取一个值，该值指示 TcpClient 的基础 Socket 是否已连接到远程主机。
	ExclusiveAddressUse	获取或设置 Boolean 值，该值指定 TcpClient 是否只允许一个客户端使用端口。
	LingerState	获取或设置有关关联的套接字的延迟状态的信息。
	NoDelay	获取或设置一个值，该值在发送或接收缓冲区未满时禁用延迟。
	ReceiveBufferSize	获取或设置接收缓冲区的大小。
	ReceiveTimeout	获取或设置在初始化一个读取操作以后 TcpClient 等待接收数据的时间量。
	SendBufferSize	获取或设置发送缓冲区的大小。
	SendTimeout	获取或设置 TcpClient 等待发送操作成功完成的时间量。

页首

# 方法


显示: ☒ 继承 ☒ 保护


	名称	说明
	BeginConnect(IPAddress, Int32, AsyncCallback, Object)	开始一个对远程主机连接的异步请求。 远程主机由 IPAddress 和端口号 (Int32) 指定。
	BeginConnect(IPAddress[], Int32, AsyncCallback, Object)	开始一个对远程主机连接的异步请求。 远程主机由 IPAddress 数组和端口号 (Int32) 指定。
	BeginConnect(String, Int32, AsyncCallback, Object)	开始一个对远程主机连接的异步请求。 远程主机由主机名 (String) 和端口号 (Int32) 指定。

	<a href="#">Close</a>	释放此 TcpClient 实例，并请求关闭基础 TCP 连接。
	<a href="#">Connect(IPEndPoint)</a>	使用指定的远程网络终结点将客户端连接到远程 TCP 主机。
	<a href="#">Connect(IPAddress, Int32)</a>	使用指定的 IP 地址和端口号将客户端连接到 TCP 主机。
	<a href="#">Connect(IPAddress[], Int32)</a>	使用指定的 IP 地址和端口号将客户端连接到远程 TCP 主机。
	<a href="#">Connect(String, Int32)</a>	将客户端连接到指定主机上的指定端口。
	<a href="#">Dispose</a>	释放由 TcpClient 占用的非托管资源，还可以另外再释放托管资源。
	<a href="#">EndConnect</a>	异步接受传入的连接尝试。
	<a href="#">Equals(Object)</a>	确定指定的 <a href="#">Object</a> 是否等于当前的 <a href="#">Object</a> 。（继承自 <a href="#">Object</a> 。）
	<a href="#">Finalize</a>	TcpClient 类使用的可用资源。（重写 <a href="#">Object.Finalize()</a> 。）
	<a href="#">GetHashCode</a>	用作特定类型的哈希函数。（继承自 <a href="#">Object</a> 。）
	<a href="#">GetStream</a>	返回用于发送和接收数据的 <a href="#">NetworkStream</a> 。
	<a href="#">GetType</a>	获取当前实例的 <a href="#">Type</a> 。（继承自 <a href="#">Object</a> 。）
	<a href="#">MemberwiseClone</a>	创建当前 <a href="#">Object</a> 的浅表副本。（继承自 <a href="#">Object</a> 。）
	<a href="#">ToString</a>	返回表示当前对象的字符串。（继承自 <a href="#">Object</a> 。）

[页首](#)

# 显式接口实现

显示: ☒ 继承 ☒ 保护 

	名称	说明
	<a href="#">IDisposable.Dispose</a>	基础结构。释放由 TcpClient 占用的所有资源。

[页首](#)

# 备注

TcpClient 类提供了一些简单的方法，用于在同步阻止模式下通过网络来连接、发送和接收流数据。

为使 TcpClient 连接并交换数据，使用 TCP [ProtocolType](#) 创建的 [TcpListener](#) 或 [Socket](#) 必须侦听是否有传入的连接请求。可以使用下面两种方法之一连接到该侦听器：

- 创建一个 TcpClient，并调用三个可用的 [Connect](#) 方法之一。
- 使用远程主机的主机名和端口号创建 TcpClient。此构造函数将自动尝试一个连接。

#### 注意

如果要在同步阻止模式下发送无连接数据报，请使用 [UdpClient](#) 类。

#### 对继承者的说明

要发送和接收数据，请使用 [GetStream](#) 方法来获取一个 [NetworkStream](#)。调用 [NetworkStream](#) 的 [Write](#) 和 [Read](#) 方法与远程主机之间发送和接收数据。使用 [Close](#) 方法释放与 TcpClient 关联的所有资源。

## 示例

下面的代码示例建立 TcpClient 连接。

C#

```
static void Connect(String server, String message)
{
    try
    {
        // Create a TcpClient.
        // Note, for this client to work you need to have a TcpServer
        // connected to the same address as specified by the server, port
        // combination.
        Int32 port = 13000;
        TcpClient client = new TcpClient(server, port);

        // Translate the passed message into ASCII and store it as a Byte array.
        Byte[] data = System.Text.Encoding.ASCII.GetBytes(message);

        // Get a client stream for reading and writing.
        // Stream stream = client.GetStream();

        NetworkStream stream = client.GetStream();

        // Send the message to the connected TcpServer.
        stream.Write(data, 0, data.Length);

        Console.WriteLine("Sent: {0}", message);

        // Receive the TcpServer.response.
```

```

// Buffer to store the response bytes.
data = new Byte[256];

// String to store the response ASCII representation.
String responseData = String.Empty;

// Read the first batch of the TcpServer response bytes.
Int32 bytes = stream.Read(data, 0, data.Length);
responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes);
Console.WriteLine("Received: {0}", responseData);

// Close everything.
stream.Close();
client.Close();
}
catch (ArgumentNullException e)
{
    Console.WriteLine("ArgumentNullException: {0}", e);
}
catch (SocketException e)
{
    Console.WriteLine("SocketException: {0}", e);
}

Console.WriteLine("\n Press Enter to continue...");
Console.Read();
}

```

## 版本信息

.NET Framework

受以下版本支持：4、3.5、3.0、2.0、1.1、1.0

.NET Framework Client Profile

受以下版本支持：4、3.5 SP1

## .NET Framework 安全性

- [SocketPermission](#)  
用于建立传出的连接或接受传入的请求的权限。

## 平台

Windows 7, Windows Vista SP1 或更高版本, Windows XP SP3, Windows XP SP2 x64 Edition, Windows Server 2008 ( 不支持服务器核心 ), Windows Server 2008 R2 ( 支持 SP1 或更高版本的服务器核心 ), Windows Server 2003 SP2

.NET Framework 并不是对每个平台的所有版本都提供支持。有关支持的版本的列表，请参见[.NET Framework 系统要求](#)。

## 线程安全

此类型的任何公共 **static**（在 Visual Basic 中为 **Shared**）成员都是线程安全的。但不保证所有实例成员都是线程安全的。

## 请参见

参考

[System.Net.Sockets 命名空间](#)

[TcpListener](#)

[NetworkStream](#)

[Socket](#)

[ProtocolType](#)

[IPEndPoint](#)

[Connect](#)

[Write](#)

[Read](#)

其他资源

[TCP/UDP](#)

---

## 社区附加资源