

[Réduire tout](#) [Code : C#](#)

Bibliothèque de classes .NET Framework

UdpClient, classe

[Membres](#) [Exemple](#) [Voir aussi](#) [Commentaires](#)

Fournit les services réseau du protocole UDP (User Datagram Protocol).

Espace de noms : [System.Net.Sockets](#)

Assembly : System (dans System.dll)

Syntaxe

C#

```
public class UdpClient : IDisposable
```

Notes

La classe **UdpClient** fournit des méthodes simples pour l'envoi et la réception de datagrammes UDP sans connexion en mode blocage synchrone. UDP étant un protocole de transport sans connexion, il n'est pas nécessaire d'établir une connexion d'hôte distant avant d'envoyer et de recevoir des données. Vous pouvez toutefois également établir un hôte distant par défaut en utilisant l'une des méthodes suivantes :

- Créez une instance de la classe **UdpClient** en utilisant comme paramètres le nom d'hôte distant et le numéro de port.
- Créez une instance de la classe **UdpClient** puis appelez la méthode [Connect](#).

Vous pouvez utiliser n'importe quelle méthode d'envoi fournie dans **UdpClient** pour envoyer des données à un périphérique distant. Utilisez la méthode [Receive](#) pour recevoir des données d'hôtes distants.

Remarque :

N'appellez pas la méthode [Send](#) en utilisant un nom d'hôte ou [IPEndPoint](#) si vous avez déjà spécifié un hôte distant par défaut. Si vous le faites, **UdpClient** lèvera une exception.

Les méthodes **UdpClient** permettent en outre d'envoyer et de recevoir des datagrammes multicast. Utilisez la méthode [JoinMulticastGroup](#) pour abonner **UdpClient** à un groupe multicast. Utilisez la méthode [DropMulticastGroup](#) pour annuler l'abonnement de **UdpClient** à un groupe multicast.

Exemples

L'exemple suivant établit une connexion **UdpClient** à l'aide du nom d'hôte `www.contoso.com` sur le port 11 000. Un petit message de chaîne est envoyé à deux ordinateurs hôtes distants séparés. La méthode [Receive](#) bloque l'exécution jusqu'à la réception d'un message. En utilisant le [IPEndPoint](#) passé à [Receive](#), l'identité de l'hôte qui répond est révélée.

C#

[Copier le code](#)

```
// This constructor arbitrarily assigns the local port number.
UdpClient udpClient = new UdpClient(11000);
try{
    udpClient.Connect("www.contoso.com", 11000);

    // Sends a message to the host to which you have connected.
    Byte[] sendBytes = Encoding.ASCII.GetBytes("Is anybody there?");

    udpClient.Send(sendBytes, sendBytes.Length);

    // Sends a message to a different host using optional hostname and port parameters.
    UdpClient udpClientB = new UdpClient();
    udpClientB.Send(sendBytes, sendBytes.Length, "AlternateHostMachineName", 11000);

    //IPEndPoint object will allow us to read datagrams sent from any source.
    IPEndPoint RemoteIpEndPoint = new IPEndPoint(IPAddress.Any, 0);

    // Blocks until a message returns on this socket from a remote host.
    Byte[] receiveBytes = udpClient.Receive(ref RemoteIpEndPoint);
    string returnData = Encoding.ASCII.GetString(receiveBytes);

    // Uses the IPEndPoint object to determine which of these two hosts responded.
    Console.WriteLine("This is the message you received " +
        returnData.ToString());
    Console.WriteLine("This message was sent from " +
        RemoteIpEndPoint.Address.ToString() +
        " on their port number " +
        RemoteIpEndPoint.Port.ToString());

    udpClient.Close();
    udpClientB.Close();
}
```

```
    }  
    catch (Exception e ) {  
        Console.WriteLine(e.ToString());  
    }  
}
```

✚ Autorisations

- [SocketPermission](#)
Établir une connexion sortante ou accepter une demande entrante.

✚ Hiérarchie d'héritage

[System.Object](#)

System.Net.Sockets.UdpClient

✚ Sécurité des threads

Tous les membres **static** (**Shared** en Visual Basic) publics de ce type sont thread-safe. Il n'est pas garanti que les membres d'instance soient thread-safe.

✚ Plateformes

Windows Vista, Windows XP SP2, Windows XP Media Center Edition, Windows XP Professionnel Édition x64, Windows XP Starter Edition, Windows Server 2003, Windows Server 2000 SP4, Windows Millennium Edition, Windows 98, Kit de développement logiciel (CE) Windows, Windows Mobile pour Smartphone, Windows Mobile pour Pocket PC

Le .NET Framework et le .NET Compact Framework ne prennent pas en charge toutes les versions de chaque plateforme. Pour obtenir la liste complète des versions prises en charge, consultez [Configuration requise du .NET Framework](#).

✚ Informations de version

.NET Framework

Pris en charge dans : 3.5, 3.0 SP1, 3.0, 2.0 SP1, 2.0, 1.1, 1.0

.NET Compact Framework

Pris en charge dans : 3.5, 2.0, 1.0

✚ Voir aussi

Concepts

[TCP/UDP](#)

Référence

[Membres UdpClient](#)

[System.Net.Sockets, espace de noms](#)

[TcpClient](#)

Envoyer [des commentaires](#) à propos de cette rubrique à Microsoft.