**Listening with Sockets**

**.NET Framework 4**

Listener or server sockets open a port on the network and then wait for a client to connect to that port. Although other network address families and protocols exist, this example shows how to create remote service for a TCP/IP network.

The unique address of a TCP/IP service is defined by combining the IP address of the host with the port number of the service to create an endpoint for the service. The [Dns](http://msdn.microsoft.com/en-us/library/system.net.dns.aspx) class provides methods that return information about the network addresses supported by the local network device. When the local network device has more than one network address, or if the local system supports more than one network device, the **Dns** class returns information about all network addresses, and the application must choose the proper address for the service. The Internet Assigned Numbers Authority (Iana) defines port numbers for common services (for more information, see www.iana.org/assignments/port-numbers). Other services can have registered port numbers in the range 1,024 to 65,535.

The following example creates an [IPEndPoint](http://msdn.microsoft.com/en-us/library/system.net.ipendpoint.aspx) for a server by combining the first IP address returned by **Dns** for the host computer with a port number chosen from the registered port numbers range.

**C#**

IPHostEntry ipHostInfo = Dns.Resolve(Dns.GetHostName());

IPAddress ipAddress = ipHostInfo.AddressList[0];

IPEndPoint localEndPoint = new IPEndPoint(ipAddress, 11000);

After the local endpoint is determined, the [Socket](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.aspx) must be associated with that endpoint using the [Bind](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.bind.aspx) method and set to listen on the endpoint using the [Listen](http://msdn.microsoft.com/en-us/library/system.net.sockets.socket.listen.aspx) method. **Bind** throws an exception if the specific address and port combination is already in use. The following example demonstrates associating a **Socket** with an **IPEndPoint**.

**C#**

listener.Bind(localEndPoint);

listener.Listen(100);

The **Listen** method takes a single parameter that specifies how many pending connections to the **Socket** are allowed before a server busy error is returned to the connecting client. In this case, up to 100 clients are placed in the connection queue before a server busy response is returned to client number 101.