

One Page Series: Windows XP Service Pack 2 -- Enabling IPv6 With Teredo

Before Beginning

This handout is for users running **Windows XP with Service Pack 2 with all current patches from Microsoft installed**. If you **do not** have all applicable service packs and patches from Microsoft installed, do not attempt to enable IPv6. Backup your system first, then run Microsoft Update (see <http://windowsupdate.microsoft.com/>). Rerun Microsoft Update until you have no remaining patches to install. After you have all applicable patches installed, for your own protection, **do one last final backup before enabling IPv6**. You should also make sure that your system is free of any viruses or other malware before proceeding.

Assumptions

Many users running Windows XP will do so from behind a hardware firewall/broadband router such as the very popular Linksys WRT54G. That device (and most other consumer hardware firewalls/broadband routers) do not do IPv6 when running their as-shipped factory firmware. While technically proficient users can reflash their Linksys device using a third party IPv6-aware replacement image (such as <http://www.dd-wrt.com/>), we believe that most users will likely not follow that path.

Thus, to meet the needs of typical Windows users we need NAT traversal. We'll illustrate configuring a Windows XP system to access IPv6 via Teredo, an IPv6-enabling protocol which will tunnel IPv6 traffic through non-IPv6-aware hardware firewall devices.

Enabling IPv6 on Windows XP Using Teredo

To enable IPv6, begin by putting up a Windows command prompt terminal window

Start ==> Programs ==> Accessories ==> Command Prompt

C:> netsh interface ipv6 install

(if you ever want to remove, change install to uninstall)

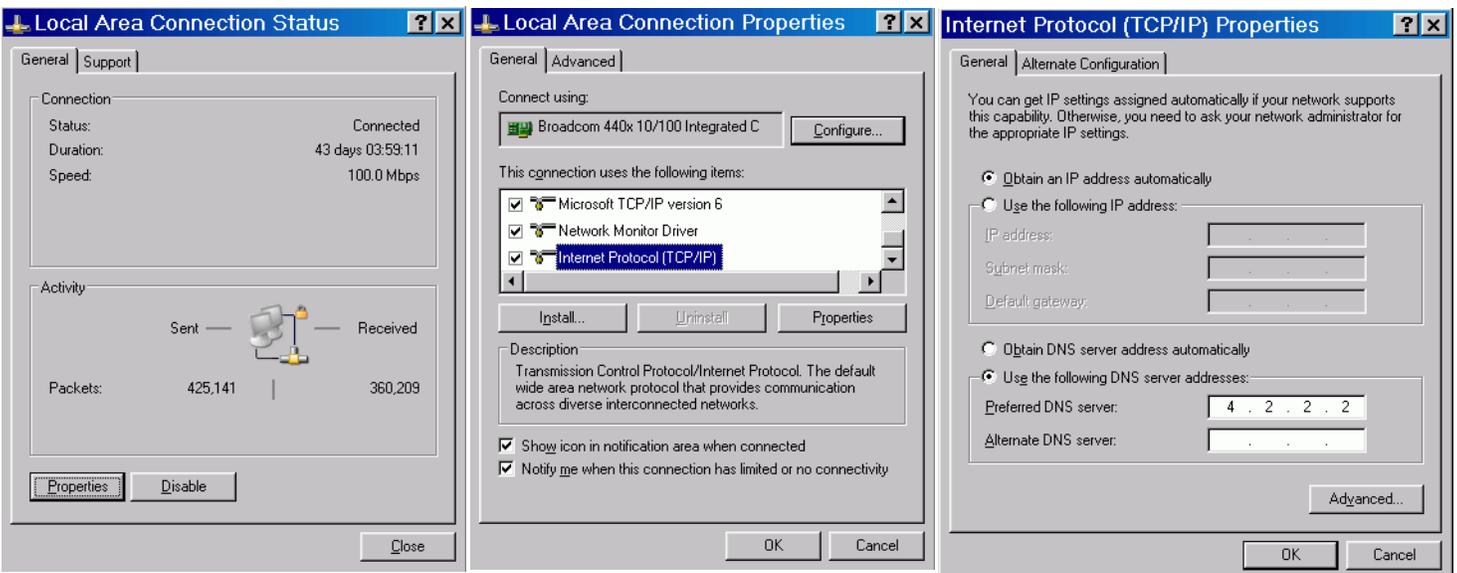
C:> netsh interface ipv6 set teredo client (member of a domain? use enterpriseclient not client) (to disable, use disabled, not client)

C:> ping ipv6.google.com (no reply? you probably need to configure an alternative IPv6 aware name server)

Configuring An IPv6 Aware Name Server, If Needed

Windows XP Service Pack 2 needs to one or more IPv6-**aware** name servers defined (an IPv6-aware name server is one which can return AAAA records for IPv6-connected fully qualified domain names). Note that Windows XP does **not** support name servers **accessible** only via IPv6. You may already have received a suitable name server as part of your existing IPv4 configuration, but if not, you will want to add an IPv6-aware name server. To manually configure an IPv6-aware name server, do:

Start ==> Settings ==> Control Panel ==> Network Connections ==> Local Area Connection



Click Properties.

Come down to Internet Protocol (TCP/IP) and select that item. Now click Properties.

Click Use the following DNS server address and enter the name of the IPv6 aware name server you want to use (such as 4.2.2.2).

Click OK, OK and Close. You should now have a manually-configured IPv6-aware name server.

Enabling the Windows Software Firewall for Teredo

By default, Teredo is not subject to the Windows software firewall that ships with Windows. To enable/confirm IPv6 firewall coverage:

Start ==> Settings ==> Control Panel ==> Windows Firewall ==> Advanced tab
Local Area Connection should be checked (and should stay checked)
Select Local Area Connection, then click on Settings
If Teredo is listed as an accessible service (e.g., it is checked), uncheck it
Click OK, OK

If you are using a 3rd party Windows software firewall instead of the integrated Microsoft Firewall, consult your vendor for advice with respect to using that firewall in a Teredo-enabled IPv6 environment.

Confirming You Have IPv6 Connectivity

To confirm that you have IPv6 connectivity, try tracing to an IPv6 only destination such as `ipv6.google.com`

Start ==> Programs ==> Accessories ==> Command Prompt
C:\> `tracert6 ipv6.google.com`

If you see a series of hops with IPv6 addresses, you have IPv6 connectivity. You may also find it interesting to see the path your traffic takes when using Teredo for IPv6 connectivity.

Technical Stuff: Protocol Preferences (Most Regular Users Should Skip This Section)

Assume you now have at least IPv4 connectivity as well as IPv6 connectivity via Teredo (you may also have native IPv6 connectivity, IPv6 connectivity via 6to4, IPv6 connectivity via ISATAP, IPv6 connectivity via a configured tunnel, etc.).

If a host is available via IPv4 and at least one IPv6 connection modality, what's the right order (or precedence) with which traffic to that dual stack host should be tried? Should traffic go over IPv4 whenever possible, or over IPv6 whenever possible?

Obviously IPv4 is the old-and-proven familiar/safe/production quality choice, **but** the object of this whole exercise **is** to try IPv6, right? On the other hand, some IPv6 connectivity (such as Teredo, as we've configured here) may be less direct/more circuitous than production IPv4 connections, with greater latency and maybe even some packet loss. Do we really want to force as much traffic as possible over that sort of disadvantaged route?

In IPv6, where a single computer may have multiple IPv6 addresses assigned, protocol preferences also influence which of multiple IPv6 address will be used when composing outbound IPv6 traffic.

To see the current policy, in a terminal window use the command:

```
C:\> netsh interface ipv6 show prefixpolicy
```

For example, a typical default prefix policy (matching well with RFC3484 recommendations) might look like:

Precedence	Label	Prefix	
5	5	2001::/32	<== Teredo-tunnelled IPv6 connectivity (least preferred)
10	4	::ffff:0:0/96	<== IPv4 connectivity (represented here in IPv6 mapped format)
20	3	::/96	<== deprecated; see RFC4291 at section 4
30	2	2002::/16	<== 6to4 IPv6 connectivity
40	1	::/0	<== native IPv6 connectivity
50	0	::1/128	<== IPv6 localhost (like 127.0.0.1 in IPv4) (most preferred)

While theoretically you can change those preferences with netsh commands, in practice it appears difficult to force IPv6 to be preferred over IPv4 in practice under Windows XP Service Pack 2 when a hostname resolves to both an IPv6 and an IPv4 address. This is actually probably a good thing in practice, although it means that if you want to insure that traffic to a given host goes via IPv6, you may need an IPv6-only address for that host (e.g., `www.ipv6.example.com` is a common format for IPv6-only web servers)