HEXAGO Gateway6™

Gateway6™ Client Guide

Gateway6™ Client Guide

Part number HEX-DC-0005-07.

Table of Contents

About This Guide

This document describes how to configure and use the Gateway6 Client software. It also provides various deployment scenarios and describes advanced features you may wish to implement.

Gateway6 Documents

This table presents the Gateway6 documentation package.

Title	Content
Gateway6 HexOS Release Notes	Provides information on HexOS releases, such as new features, bug fixes and changes.
Gateway6 Client Release Notes	Provides information on Gateway6 Client releases, such as new features, bug fixes and changes.
Gateway6 Client with HAP6 Release Notes	Provides information on Gateway6 Client with HAP6 releases, such as new features, bug fixes and changes. This document, together with the *Gateway6 HexOS Release Notes* and the *Gateway6 Client Release Notes*, should be read first.
Gateway6 Documentation Guide	Describes the Gateway6 documentation package, introduces the HexOS software, and describes the CLI (Command Line Interface) command modes and basic features.
Gateway6 Quick Setup Guide	Provides hardware installation procedures and minimal software configuration procedures.
Gateway6 HexOS Configuration Guide	Shows you how to configure Gateway6 using the CLI.
Gateway6 HexOS Command Reference	Describes the four main types of Gateway6 commands: ▸ Management and protocol-independent commands ▸ Interface and access list commands ▸ Tunnel broker commands ▸ Logging and troubleshooting commands
Gateway6 Client Guide (This document)	Explains how to configure and use the Gateway6 Client.
Gateway6 Client with HAP6 Guide	Explains how to configure and use the Home Access Platform.

Obtaining Documentation

The Gateway6 documents are supplied as Portable Document Format (PDF) files on the Gateway6 Software & Documentation CD-ROM. Printed copies of these documents are also available.

The Software & Documentation CD-ROM also contains HexOS image files, Gateway6 Client software, the latest documentation updates, as well as HexOS software and copyright information. You can also find these items on the Hexago corporate Web site.

Revision

See the *Gateway6 HexOS Release Notes* to learn which software version corresponds to this document. The revision number of this document is 07.

This section gives an overview of the relationship between the Gateway6 Client and the Gateway6 server (ofter referring simply as the Gateway6), and how they interact.

TSP is a protocol that establishes and maintains static data tunnels. Located on the host computer (local node), the Gateway6 Client connects to the Gateway6 server and obtains tunnel-related information using the TSP protocol. Upon receiving the information for the tunnel, the Gateway6 Client creates a static tunnel on the local operating system.

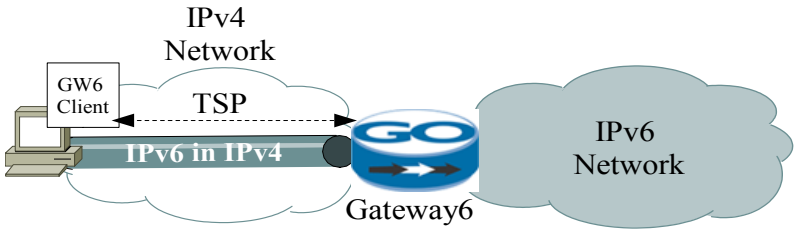


Figure 1 - Gateway6 Client and Gateway6

The Gateway6 Client source code is mostly identical for all client platforms. Creating the static tunnel, on the other hand, is dependent on the operating system and is completed by means of a script executed by the Gateway6 Client. These scripts are located in the `template` subdirectory of the Gateway6 Client installation directory.

Client Components

Figure 1 below shows the various components of the Gateway6 Client, as well as how they interact with the local node's operating system and the Gateway6.

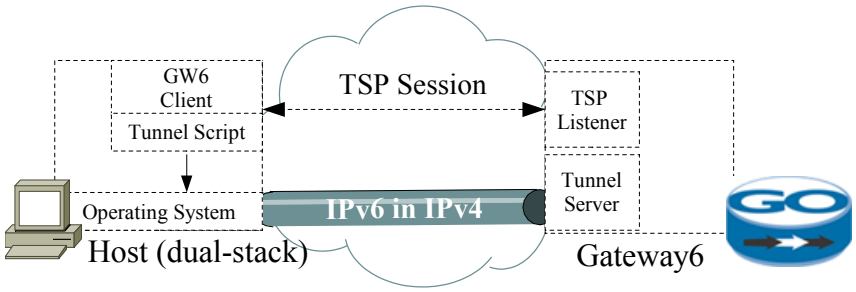


Figure 2 - Gateway6 Client components

The script executed by the Gateway6 Client to configure the tunnel interface is customized for each supported target operating system in order to handle its unique characteristics. On Unix systems, it is a shell script, while for Microsoft Windows, it takes the form of a batch file (`*.bat`). Separating the binary from the script simplifies adding new operating systems, as has been demonstrated by the community contributions for many different environments.

Packaging

The Gateway6 Client is available in several ways:

- It is part of certain operating system distributions, such as Linux or FreeBSD
- It can be downloaded from the Web site of the tunnel broker service, such as Freenet6 (http://www.go6.net)
- It is included on the Gateway6 installation CD-ROM
- It can be obtained directly from Hexago (http://www.hexago.com)

Multi-Site Operation

You can use the Gateway6 Client to connect to a single Gateway6 server or to multiple servers in different locations. This flexibility serves two purposes: it offers a better quality of service by enabling users to connect to the server located closest to them, and it provides redundancy in the event that one site becomes unavailable.

TSP announces multiple sites using a mechanism called a *broker list* (if there are several sites) or *broker redirection* (if there is only one site to announce). The Gateway6 Client that receives a broker list will use echo messages to test which sites are available, along with their respective topological distance. The client then connects to each broker in the list, starting with the closest one, until a successful connection is established.

In the example presented in Figure 3 below, the Gateway6 Client connects to GW6 #1 initially in order to retrieve the broker list. It then sends out echo requests to all brokers in the list and orders them according to their proximity. Based on the round-trip time results returned by the Gateway6 units, GW6 #2 is the first to be contacted by the client, followed by GW6 #1 and GW6 #3. The client connects to GW6 #2 and successfully negotiates a tunnel. If the GW6 #2 ever becomes unavailable, the client will attempt to connect to GW6 #1.

A given Gateway6 unit can act as both the Master and a tunnel server at the same time. A Master Gateway6 would redirect connections onto itself using a second IP address.

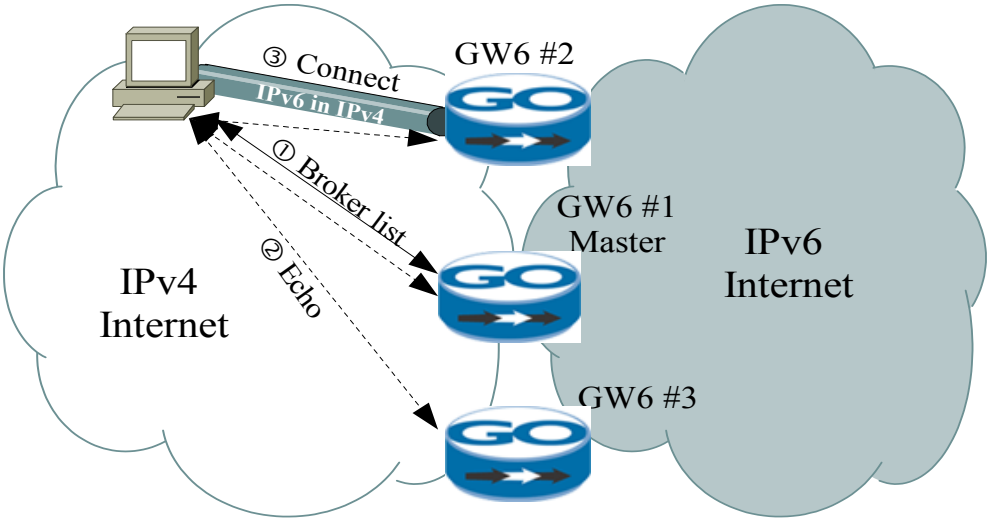


Figure 3 - Gateway6 multi-site deployment

Configuring the Gateway6 Client

The Gateway6 Client is configured using values stored in a file called `gw6c.conf`.

A graphical user interface (GUI) designed for easy configuration management and service status reporting is available for the Microsoft Windows operating system. See the "Operating System Specifics" section on page 27 for more information regarding the Windows version of the Gateway6 Client Utility.

`gw6c.conf` Configuration File

The `gw6c.conf` configuration file is a raw text file whose comments are identified by the "#" character. Each statement adheres to the syntax `variable = value`, similar to the `rc.conf` files in FreeBSD. The configuration statements presented below have been sorted into three categories: basic configuration, router configuration (when a prefix is delegated) and advanced configuration. Many parameters are optional; most users will only need to modify the basic configuration statements.

Basic Configuration Statements

Variable	Default Value	Possible Values	Description
`userid`	`A` `<none>`	`string`	The user identification string. Not required if connecting anonymously (`auth_method=anonymous`)
`passwd`		`string`	The password associated with `userid`.

`server`	`broker.freenet6.net`	`ip_address`
		`hostname`
		`ip_address:port`
		`hostname:port`
			The broker's IP address or hostname (full domain name, when appropriate). A TSP port number can be specified as well.

Router Mode Configuration Statements

Variable	Default Value	Possible Values	Description
`host_type`	`host`	`host` `router`	Specifies whether the Gateway6 Client is a **host** or a **router**. In **router mode**, the Gateway6 Client receives a prefix when `prefixlen` is set.
`prefixlen`	`48`	`<48-64>`	The length of the prefix required by the Gateway6 Client.
`if_prefix`		string interface ID (Windows)	The interface in the Gateway6 Client's operating system that is used to send router advertisements with the prefix received from Gateway6. On Windows, this may be an interface ID. (Use the *netsh interface ipv6 show interface* command to list IPv6 interfaces with their IDs.)

Variable	Default Value	Possible Values	Description
`dns_server`		string	The fully-qualified domain name of the DNS server that handles reverse DNS delegation of the prefix. Multiple servers can be specified by separating them with the colon character `:` .

Advanced Configuration Statements in `gw6c.conf`

Variable	Default Value	Possible Values	Description
`gw6_dir`	Gateway6 Client installation directory	string	The Gateway6 Client installation directory containing the `template` subdirectory where the configuration scripts are stored.

`auth_method`	`anonymous`	`any`
		`digest-md5`
		`passdss-3des-1`
		`plain`
		`anonymous`
			The type of authentication used for the TSP session. **Digest-md5 and passdss-3des-1 are** the most secure, since they protect the passwords. **Plain** sends the userid and password without any protection. **Anonymous** does not send a userid or password. With **any**, the Gateway6 Client uses the most secure mode based on its capabilities, as well as the broker's authentication capabilities. The **any** value is recommended when not connecting anonymously.

`client_v4`	`auto`	`auto` *IPv4 address*	The IPv4 address used by the Gateway6 Client as its tunnel endpoint source address. When this parameter is set to **auto**, the Gateway6 Client uses the first IPv4 address assigned by the operating system. Leaving this setting to **auto** is recommended.
`client_v6`	`auto`	`auto` *IPv6 address*	The IPv6 address used by the Gateway6 Client as its tunnel endpoint. This parameter is used only when requesting a V4V6 tunnel. Leaving this setting to `auto` is recommended.

Variable	Default Value	Possible Values	Description

`template`		`checktunnel`
		`darwin`
		`freebsd`
		`linux`
		`netbsd`
		`openbsd`
		`windows`
		`cisco`
			The script file used to create the tunnel. The value of this parameter is the name of the script file (located in the `template` directory) that will be called by the Gateway6 Client at the end of the TSP session to create the tunnel.
			When compiling the Gateway6 Client, the `template` variable is assigned the correct value for the operating system used to complete the compilation.

`auto_retry_connect`	`Yes`	`Yes` `No`	Specifies whether the Gateway6 Client should try reconnecting after a disconnection provoked by a keepalive timeout.
`retry_delay`	`30`	`number`	When the tunnel is disconnected because of an error, the number of seconds to wait before making another attempt to connect to the tunnel broker.

`tunnel_mode`	`v6anyv4`	`v6v4`
		`v6udpv4`
		`v6anyv4`
		`v4v6`
			The tunnel encapsulation mode, as described on page 16 in the “Tunnel Encapsulation Modes” section. Note that not all encapsulation modes are available on all platforms. Refer to the platform-specific sections for details.

`if_tunnel_v6v4`	*Automatically set during installation*	String interface ID (Windows)	The tunnel interface on the client's operating system that is used for IPv6-in-IPv4 encapsulation.
`if_tunnel_v4udpv4`	*Automatically set during installation*	String interface ID (Windows)	The tunnel interface on the client's operating system that is used for IPv6-in-UDP-in-IPv4 encapsulation.
`if_tunnel_v4v6`	*Automatically set during installation*	String interface ID (Windows)	The tunnel interface on the client's operating system that is used for IPv4-in-IPv6 encapsulation.
`proxy_client`	`no`	`yes` `no`	When set to **yes**, the Gateway6 Client is a proxy for the tunnel endpoint, as opposed to the tunnel endpoint itself. This parameter is useful when creating a tunnel for an external router.

Variable	Default Value	Possible Values	Description
keepalive	yes	yes no	When set to **yes**, the Gateway6 Client sends packets to keep the tunnel active. This is especially useful for environments with Network Address Translation (NAT) since they must retain their NAT mapping in order to have a sustainable tunnel over UDP. It can also be used to monitor the tunnel on both sides.
keepalive_interval	30	number	This interval, expressed in seconds, must be shorter than the NAT mapping timeout for UDP. The Gateway6 Client sends a keepalive packet to the broker at every keepalive_interval. The broker may force a higher value than what is entered here, depending on the load from a provider that is expected on the broker. Recommended values are between 30 and 110 seconds.
log_file	0 Windows: 1	0, 1, 2, 3	Specifies the log verbosity level in the file.
log_stderr	1 Windows: 0	0, 1, 2, 3	Specifies the log verbosity level printed to the standard error.
log_console	0	0, 1, 2, 3	Specifies the log verbosity level sent to the console.
log_syslog	0	0, 1, 2, 3	Specifies the log verbosity level sent to syslog.
log_filename	gw6c.log	String filename	Specifies the filename to use for logging purposes when log_file=[level] is encountered in the configuration.
log_rotation	yes	yes no	Specifies whether the log file should use the rotation feature. When enabled, the contents of the file are moved to a backup file before the log file reaches the size specified in the log_rotation_size variable. The backup file name contains the timestamp.
log_rotation_size	32	16, 32, 128, 1024	Directive controlling the size the log file must reach before its contents are moved to an archive file.

Variable	Default Value	Possible Values	Description
syslog_facility	USER	USER, LOCAL [0-7]	Indicates the facility when the log directive is set to syslog. (Unix platforms only)
last_server	Tsp-last-server.txt	File name	Name of the text file containing the address of the last broker to which a connection was successfully established.
always_use_same_server	no	yes no	Determines whether the client will always attempt to connect to the broker specified with the last_server directive (if any).
broker_list	tsp-broker-list.txt	File name	Name of the text file where the broker list, received in a redirection instruction, is saved.
hap6_web_enabled	no	yes no	Specifies whether or not the Home Web feature is enabled for use.
hap6_proxy_enabled	no	yes no	Specifies whether or not the Home Access feature is enabled for use.
hap6_document_root	None	A full directory path	If the Home Web feature has been enabled for use (hap6_web_enabled=yes), this statement MUST indicate the full path leading to the directory that serves as the document root. The folder specified here will become the root folder for tree structure of files that will comprise the Web site available on the client host.

The template variable contains the name of the script file executed upon conclusion of the TSP session. On Unix-based operating systems (BSD, Linux, MacOS X), the **.sh** extension is appended to the file name before the script is executed; the **.bat** extension is appended instead for Microsoft Windows. How to customize the script is discussed in the "Configuring the Gateway6 Client" section on page 53 of this document.

Last Server File `tsp-last-server.txt`

The Last Server file stores the name or address of the last server to which a successful connection was established. This parameter is used to reconnect if the always_use_same_server variable has been set to *yes*, usually in cases where there is a static host or router for which the same IPv6 address and prefix are desired.

Broker List File `tsp-broker-list.txt`

The Broker List file is generated automatically when the client receives a list of brokers from a TSP server. The information contained in this file is used in multiple-site configurations to announce the available brokers to clients.

Tunnel Encapsulation Modes

Tunnel encapsulation modes keywords, as defined by the `tunnel_mode` variable of the `gw6c.conf` configuration file, are listed in the table below. When `v6anyv4` is sent by the Gateway6 Client, the broker tests to determine whether the client is located behind a NAT, then responds by setting the correct encapsulation mode.

Tunnel Mode Keyword	Description
v6v4	IPv6-in-IPv4 encapsulation, using IPv4 protocol 41. This encapsulation mode is not compatible with a NAT.
v6udpv4	IPv6-in-UDP-in-IPv4 encapsulation. This encapsulation mode is designed to work gracefully through a NAT.
v6anyv4	IPv6 in any IPv4 encapsulation. When this mode is enabled, the tunnel broker will suggest the correct encapsulation method to the Gateway6 Client based on whether or not the broker discovers a NAT in the path. If the broker finds a NAT, then **v6udpv4** is proposed to the Gateway6 Client, otherwise **v6v4** is proposed.
v4v6	IPv4-in-IPv6 encapsulation. (DSTM)

NOTE: Not all Gateway6 Client platforms support the NAT traversal feature implemented with the **v6udpv4** encapsulation mode or the IPv4-in-IPv6 encapsulation. Please refer to the *Gateway6 Client Release Notes* for platform-specific feature support.

Executing the Gateway6 Client

The command line version of the Gateway6 Client is executed manually by typing the command **gw6c.** First of all, open a command prompt and navigate to the Gateway6 Client installation directory. From there, you may launch the Gateway6 Client by typing `gw6c`. The Gateway6 Client program will either remain connected (Windows only) or continue running in the background (other operating systems) to sustain the keepalive with the tunnel broker. Keepalive packets are mandatory for IPv6-in-UDP-in-IPv4 tunnels in order to sustain the NAT mapping, but are optional for IPv6-in-IPv4 tunnels and IPv4-in-IPv6 tunnels.

A second mode of operation is available for Windows. The Gateway6 Client is configured to run by default as a service by the installer. As a service, it will automatically start when the operating system launches. The automatic reconnection feature of the Gateway6 Client Utility is now user controlled. As such, when a tunnel expires due to a keepalive timeout, a popup window is displayed allowing the user to choose whether or not to reconnect.

To start the Gateway6 Client so that it is minimized to the system notification area (system tray), simply create a shortcut in the usual manner and append the `--start-minimized` command line option to the executable. This option is particularly useful when auto-launching the client at logon from the startup directory.

To configure the Gateway6 Client service, go to the *Windows Control Panel*, open *Administrative Tools*, then choose *Services*, as shown in the section "Windows: Configuring the Gateway6 Client Service" on page 43. The graphical user interface (GUI) is a user-friendly way to configure the Gateway6 Client and manage the Gateway6 Client process.

NOTE: When debugging, the Gateway6 Client should always be executed manually.

Arguments

The command line arguments for the **gw6c** program are described in the table below.

Argument	Description

`-v`
`-vv`
`-vvv`
`-vvvv`
	Sets the verbosity level and type of debugging information sent to the screen. **-vvv** yields the most debugging information, such as the TSP XML content. **-vvvv** is very verbose and is intended to monitor the keepalive process.

`-i` *interface_name*	Sets the interface name for IPv6-in-IPv4 encapsulation.
`-u` *interface_name*	Sets the interface name for IPv6-in-UDP-in-IPv4 encapsulation.
`-s` *interface_name*	Sets the interface name to configure router advertisements of the prefix when the client is a router and a prefix has been received.
`-f` *config_filename*	Sets the configuration file.
`-r` *number_of_seconds*	Sets the retry interval when the TSP connection to the broker fails.
`-h`	Shows the current version number and list of available options.
`--register`	Registers the Gateway6 Client as a Windows service. This is completed automatically as part of the installation process. (Windows only) **NOTE:** Sending this argument will **not** start the client itself.
`--unregister`	Stops the Gateway6 Client, then unregisters it as a Windows service. (Windows only)

Troubleshooting

To troubleshoot the Gateway6 Client, use **-v** or **-vv** or **-vvv** as the command line argument. The `gw6c.log` file contains logging information for the TSP session and tunnel configuration.

This section describes scenarios typically encountered by the Gateway6 Client. The depicted configurations are equally applicable to the graphical user interface and the configuration file.

Scenario #1: Single Host on an IPv4 Network, Temporary IPv6 Address

A single node attached to the IPv4 Internet requires a temporary IPv6 address and connectivity, as shown in Figure 2 on page 9. This scenario reflects the default state of the configuration file when the Gateway6 Client software is first installed. Because it uses the anonymous authentication mode, pre-registering a username is unnecessary. Required variables for this scenario are as follows:

- **`auth_method=anonymous`**
- **`host_type=host`**

Scenario #2: Single Host on an IPv4 Network, Permanent IPv6 Address

Here, the IPv6 address is bound to a username so it can become permanent. The user must first subscribe to and obtain valid login credentials (*i.e.* a userid/password) from the tunnel broker. The userid and password are then added to the configuration file for use during authentication of the TSP session with the tunnel broker. Required variables for this scenario are as follows:

- **`auth_method=any`**
- **`userid=`*`your_username`***
- **`passwd=`*`your_password`***
- **`host_type=host`**

Scenario #3: Router on an IPv4 Network, Delegated IPv6 Prefix

In this scenario, a Gateway6 Client router (R1) is forwarding IPv6 packets between the tunnel interface to the tunnel broker and another interface, as specified by the `if_prefix` parameter of the configuration file. The Gateway6 Client has also requested an IPv6 prefix from the tunnel broker to be advertised on its attached network. Figure 4 below shows an example where R1 is a Gateway6 Client that has been authenticated by the tunnel broker, and an IPv6 prefix for the Gateway6 Client's attached network has been duly requested and received.

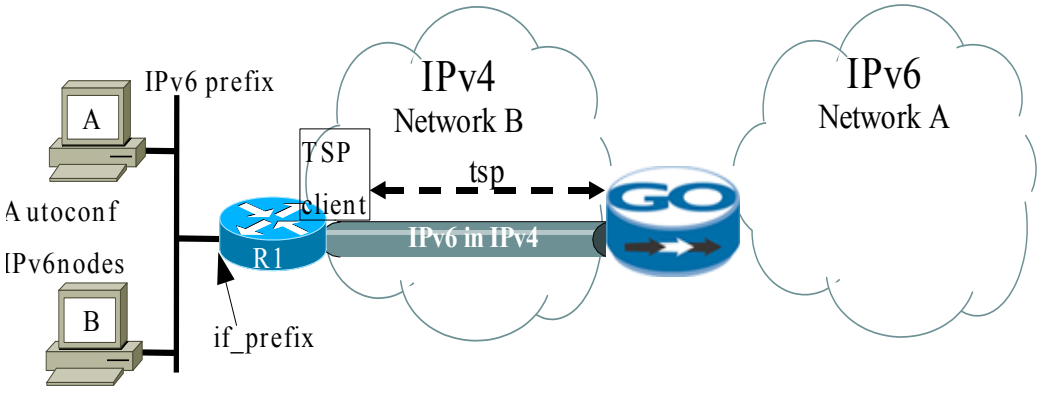


Figure 4 - Gateway6 Client as a router

Once the tunnel is successfully established, R1 advertises the received IPv6 prefix on its attached network. Nodes A and B configure themselves based on the advertised prefix. Even if R1's IPv4 address changes, the IPv6 prefix for nodes A and B will remain permanent and stable. Required variables for this scenario are as follows:

- **auth_method=any**
- **userid=*your_username***
- **passwd=*your_password***
- **host_type=router**
- **if_prefix=*interface_name***
- **prefixlen=64** (or 48 or 60, depending on how the tunnel broker is configured)

Scenario #4: Behind an IPv4 Network Address Translator (NAT)

The Gateway6 Client, which is either a host or router as described above, can be placed behind a NAT. When this is the case, IPv6-in-UDP-in-IPv4 encapsulation must be used to traverse the NAT. Figure 5 below provides an example of this type of scenario.

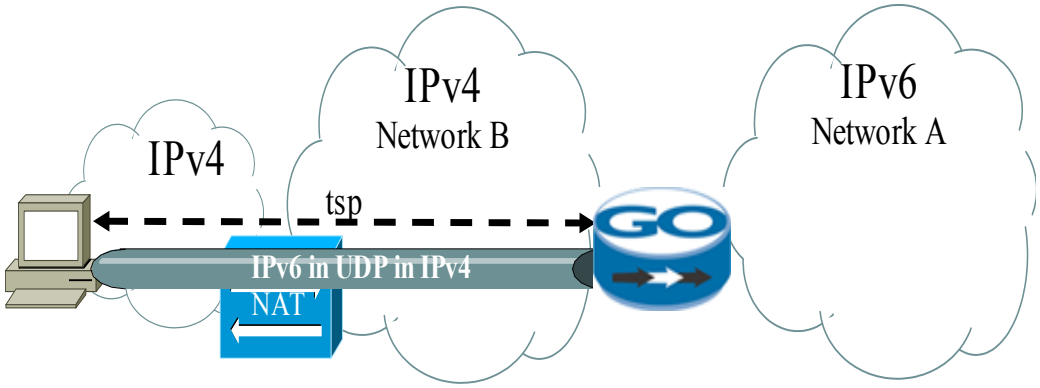


Figure 5 - Gateway6 Client behind an IPv4 NAT

It is important to understand that a Gateway6 Client cannot easily discern whether or not nodes are situated behind a NAT. For example, a node can use a public address even from behind a NAT. Moreover, a node may be using a private address space, but does not traverse a NAT in order to reach its tunnel broker. For these reasons, **v6anyv4** is the default tunnel mode used by the Gateway6 Client. The Gateway6 Client sends the tunnel request to the tunnel broker and, because the tunnel broker can verify whether or not the Gateway6 Client is located behind a NAT, it is the tunnel broker that decides which encapsulation method is appropriate for the tunnel requested by the client. This highly-flexible configuration option is recommended because it adapts to cover all possible cases, especially mobile nodes.

Required variables for this scenario are as follows:

- **auth_method=any**
- **userid=*your_username***
- **passwd=*your_password***
- **tunnel_mode=v6anyv4**
- **keepalive=yes**
- **keepalive_interval=30**
- **if_tunnel_v6udpv4=*interface_name***

The keepalive interval is used to sustain the NAT mapping. If the tunnel remains up for less than the keepalive interval, it might be because the NAT mapping lifetime is shorter than the keepalive interval. In such case, the keepalive interval should be shortened accordingly. Informal observations indicate that the NAT mapping lifetime varies from hours to only a few seconds.

Scenario #5: Mobile Node on IPv4 Networks

In this scenario, a mobile node connects to the IPv4 Internet either with or without a NAT. The configuration described in Scenario #4 with `tunnel_mode=v6anyv4` enables the mobile node to obtain the best tunnel encapsulation mode at all times.

The Gateway6 Client must be rerun whenever the mobile node's IPv4 address changes. One possibility would be to add the Gateway6 Client to the node's boot sequence; however, if the IPv4 address is changed without rebooting, the Gateway6 Client may or may not reconnect, depending on the length of time between the address changes. A safe way to ensure that the tunnel is always re-established when the IPv4 address changes is to bind the process that changes the IPv4 address with the Gateway6 Client. For example, on Unix operating systems such as Linux or FreeBSD, the DHCP client, *ISC dhclient*, can be customized to rerun the Gateway6 Client when the IPv4 address changes.

As such, the **/etc/dhclient-exit-hooks** script would contain the following commands:

```
if [ x$old_ip_address = x ] || [ x$old_ip_address != x$new_ip_address ];
then
       gw6c
fi
```

Scenario 6: Single Host on an IPv6-Only Network

A single host on a native IPv6 network requires IPv4 connectivity. Both the anonymous mode and authenticated modes can be used here, depending on whether or not a stable IPv4 address is required.

Required variables for this scenario are as follows:

Authenticated mode:

- **`auth_method=any`**
- **`userid=`*`your_username`***
- **`passwd=`*`your_password`***
- **`tunnel_mode=v4v6`**
- **`if_tunnel_v4v6=`*`interface_name`***

OR

Anonymous mode:

- **`auth_method=anonymous`**
- **`userid=`**
- ***`passwd`*`=`**
- **`tunnel_mode=v4v6`**
- **`if_tunnel_v4v6=`*`interface_name`***

This section describes the Gateway6 Client's advanced features and configuration settings.

TSP Transport and Encapsulation

The Gateway6 Client initiates the TSP session with the Gateway6 specified in the `server` parameter of the `gw6c.conf` configuration file. The Gateway6 Client first attempts to connect to the broker over UDP (thus supposing the presence of a NAT). If no connection is made, the TSP session is restarted over TCP. The presence of a NAT means that v6udpv4 requires UDP transport, since the same UDP channel is used to tunnel the IPv6 traffic upon termination of the TSP session. This approach enables the same NAT mapping to be reused, and guarantees that the tunnel for all types of NATs will be reliably established.

Tunnel Maximum Transmission Unit (MTU)

Each operating system sets the MTU of the tunnel interface to a different value. In the interests of conformity and interoperability, the TSP template script (or batch file) executed on the local operating system forces the tunnel interface's MTU to 1280 for all operating systems.

TSP Protocol Versions

Version 1.X of the TSP protocol uses TCP as the transport mechanism. Version 2.X uses either UDP or TCP. Both versions use the assigned IANA port 3653. When a Gateway6 Client connects to a broker, the Gateway6 Client advertises to the broker the latest version of the TSP protocol it supports.

When a 2.X client connects to a version 1.X broker, the Gateway6 Client first connects using UDP. Because a 1.X broker does not listen on UDP, the Gateway6 Client times out, automatically falls back to TCP, then establishes the TSP session using TCP. The entire process is quick to execute, despite the timeout.

When a 1.X client connects to a version 2.X broker, the broker seamlessly adapts to the Gateway6 Client using the 1.X TSP protocol.

At the same time, the client will advertise version 2.0.0 of the TSP protocol. If the broker is 1.X TSP protocol compliant, it will deny the Gateway6 Client's request. The Gateway6 Client then restarts the TSP session advertising the 1.X TSP protocol version.

Version 2.0 of the TSP protocol added the v6udpv4 encapsulation mode for NAT traversal, as well as a keepalive mechanism to sustain the NAT mapping while the tunnel is in use. IPv4-in-IPv6 encapsulation (v4v6) was also added in version 2.0 of the TSP protocol.

In version 2.0.1 of the TSP protocol, the broker redirection feature was added. In this version, the TSP server may respond with a list of brokers at any time in the negotiation process. The client sorts the list from the closest to the farthest, then attempts to connect to them in order. The purpose of this feature is to redirect users in the event of an error or to support multi-site operation.

Using the PASSDSS Authentication Method

The PASSDSS-3DES-1 authentication method uses a dual-authentication approach that is similar to SSH. When the Gateway6 Client connects for the first time to a broker using the PASSDSS-3DES-1 authentication mechanism, the broker sends the client its DSA public key so the client can authenticate the broker.

- If the received broker key is already present in the `gw6ckeys.pub` file, the Gateway6 Client authenticates itself to the broker as usual.
- If the received broker key is not found in the `gw6ckeys.pub` file or the file does not exist, the Gateway6 Client prompts the user to accept the broker key.
  - ☞ If the user answers *yes*, then the key is saved in `gw6ckeys.pub`.
  - ☞ If the user refuses, the TSP session is aborted.

  The next time the Gateway6 Client connects to the broker, the same key will be compared. If it is identical, the broker authentication by the client will be successful and the client will subsequently authenticate itself to the broker.

Since the broker key is associated with the IPv4 address of the broker that has been entered in the `gw6ckeys.pub` file, a change in the broker's IPv4 address will result in the user being prompted to accept the new key. The key with the broker's former IPv4 address will be purged from the `gw6ckeys.pub` file.

NOTE: In the case of the Gateway6 Client running as a Windows service, there is no way to accept user input. As such, the Gateway6 Client service automatically accepts the broker key and generates a logging record.

Lifetime of IPv4-in-IPv6 Tunnels

When requesting an IPv4 prefix with a v4v6 tunnel, the allocated IPv4 prefix lifetime is bound to the lifetime of the tunnel. The Gateway6 Client automatically renews the tunnel when its lifetime has expired.

Multi-Site Operation

Starting with version 4.2, the Gateway6 Client can operate in multi-site mode. This mode of operation is triggered by a broker presenting the Gateway6 Client with a list of brokers to which it can connect.

Upon reception of the list, the client will go through it in order, attempting to connect to each broker. Through the use of an echo mechanism (ping), the topological distance between the host and each broker is measured. The list is then sorted by distance, starting with the closest broker. Unresponsive brokers are placed at the end of the list, as are brokers with a mismatched address family (for example, a broker resolving to an IPv6 address when an IPv6 tunnel is required). This list is then saved in the `tsp-broker-list.txt` file.

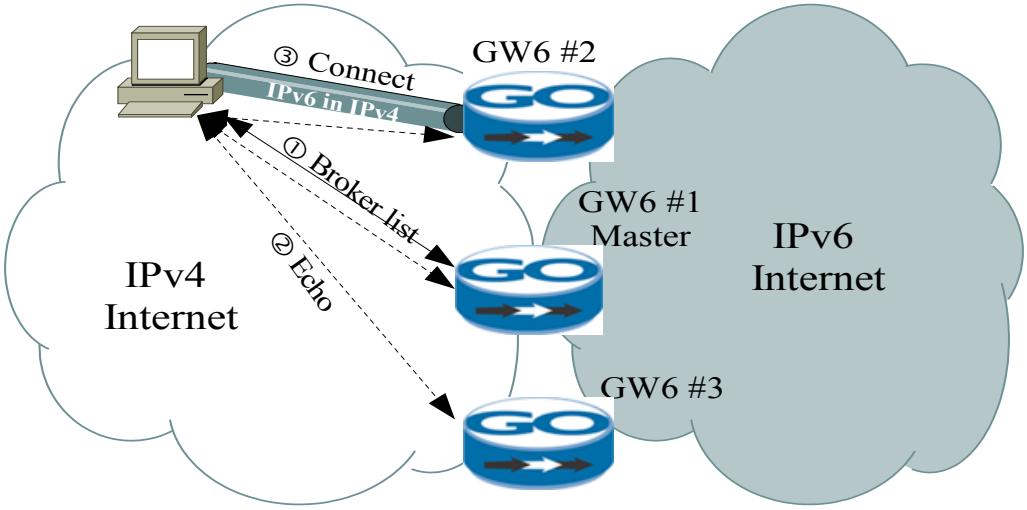


Figure 6 - Multi-site operation of the Gateway6 Client

The client then proceeds to connect to each broker in the list until a successful connection is established. When a successful connection occurs, the name of the broker is saved in the `tsp-last-server.txt` file.

Due to the fact that a given broker will usually return the same address and prefix information to the client, it may be desirable for the host or router to always reconnect to the same broker and thus avoid renumbering. To do so, set the `always_use_same_server` parameter of the `gw6c.conf` configuration file to *yes*. Otherwise, the host will receive a different address and prefix if it connects to a different server than the one to which it usually connects.

Windows: Installing the Gateway6 Client GUI

The Gateway6 Client for Windows is delivered via an installer utility. This section explains the procedures that must be followed in order to properly install the Gateway6 Client on the local node, as well as add the tunnel driver to Windows interfaces.

To install the Gateway6 Client, your system must have the minimum hardware requirements of your installed operating system and a CD-ROM drive (unless you are installing from a network connection).

It is strongly recommended to carefully read over the *Gateway6 Client Release Notes* before proceeding with the installation. When ready, follow the steps listed below and let *Setup* guide you through the installation process. Remember to close all open windows before you begin.

- Launch the *Setup* installer executable by double-clicking the installer icon, as shown below. The introduction screen will appear when you first launch the installation program; click *Next* and Setup will begin.

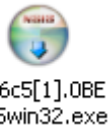


Figure 7 - Gateway6 Client Windows Package File

- When you initially open the executable, the security warning below will be displayed by your operating system. There is no reason for concern. Click the *Run* button to proceed.

Figure 8 - Open File - Security Warning

- Hexago's License Agreement for the Gateway6 Client is then displayed. Click *I Agree* to continue once you have read and understood the terms of the agreement.

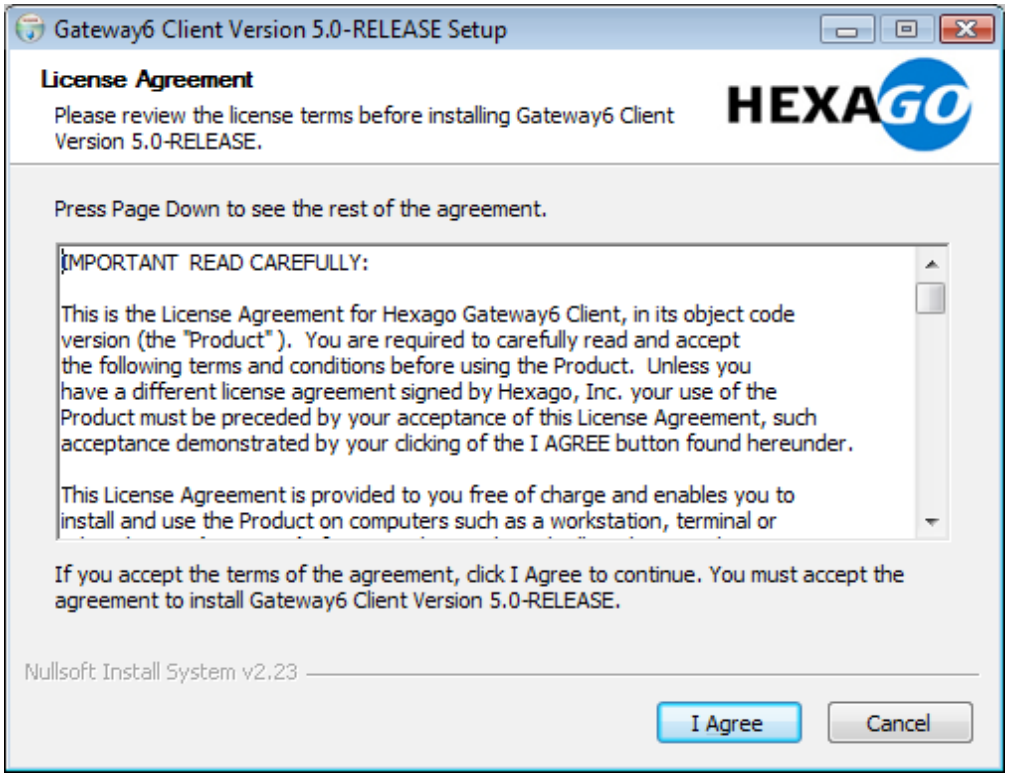


Figure 9 - Gateway6 Client License Agreement

- Select which package component(s) you wish to install by clicking the checkbox beside each one. The total disk space required will be displayed.
  - ☞ The “Gateway6 Client binaries component is mandatory and must be checked for the software to be successfully installed.
  - ☞ Only uncheck the “Tunnel Driver” component if you are certain the Gateway6 Client will never be positioned behind a NAT or if you do not require DSTM (for IPv4-in-IPv6 connectivity).
  - ☞ It is recommended to leave the “Additional Languages” checkbox selected.

  Click the *Next* button to the next screen, where you select the Gateway6 Client software installation directory.

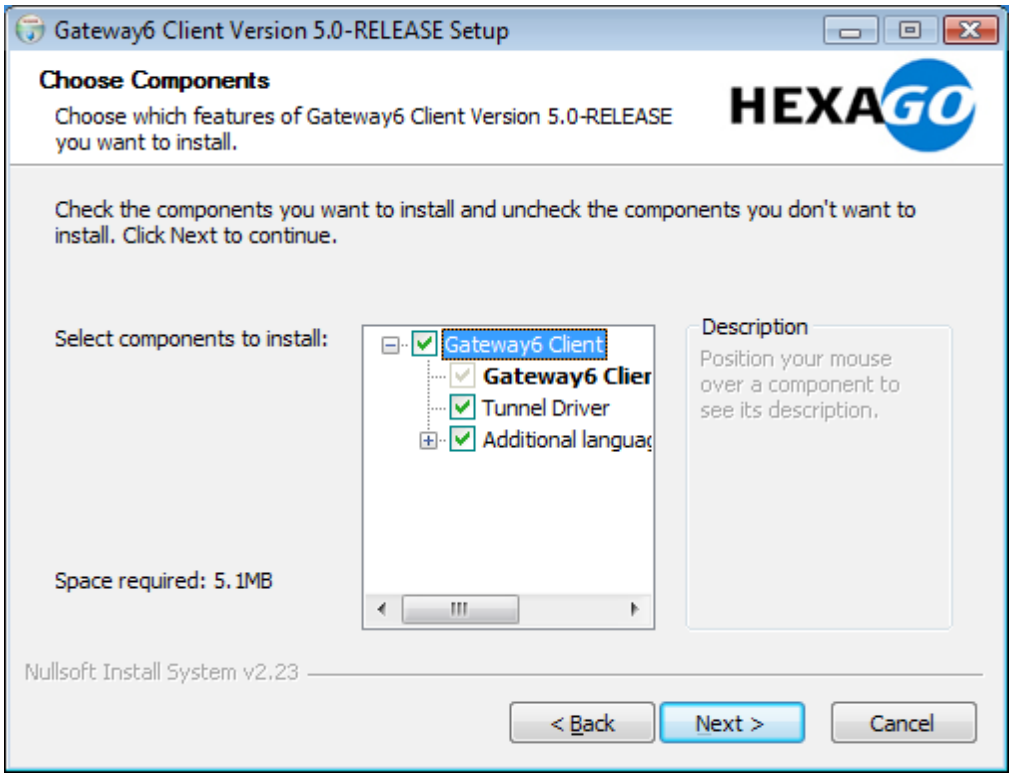


Figure 10 - Choosing Gateway6 Client components

- Define the location where the Gateway6 Client will be installed, if needed, by clicking the *Browse* button.
  - ☞ The default location is **C:\Program Files\Hexago\Gateway6 Client**.

  Click *Install* when you are ready to continue.

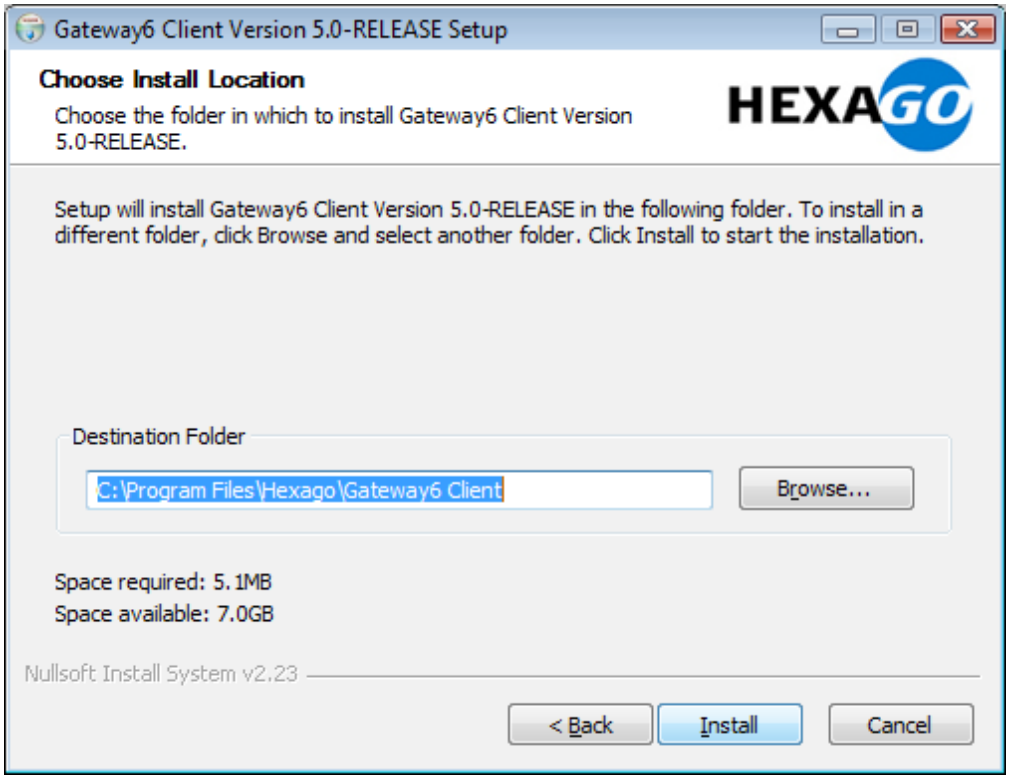


Figure 11 - Choosing installation location

- The Gateway6 Client driver is not yet officially recognized and tested for compatibility by Microsoft. This is why you will probably see this warning message. If you do, simply click *Install this driver software anyway* (Windows Vista) or *Continue Anyway* (Windows XP) to complete the installation.

NOTE: Do not be alarmed by this warning; the Gateway6 Client is already used by many of Hexago's clients and has been proven as an efficient and stable product.

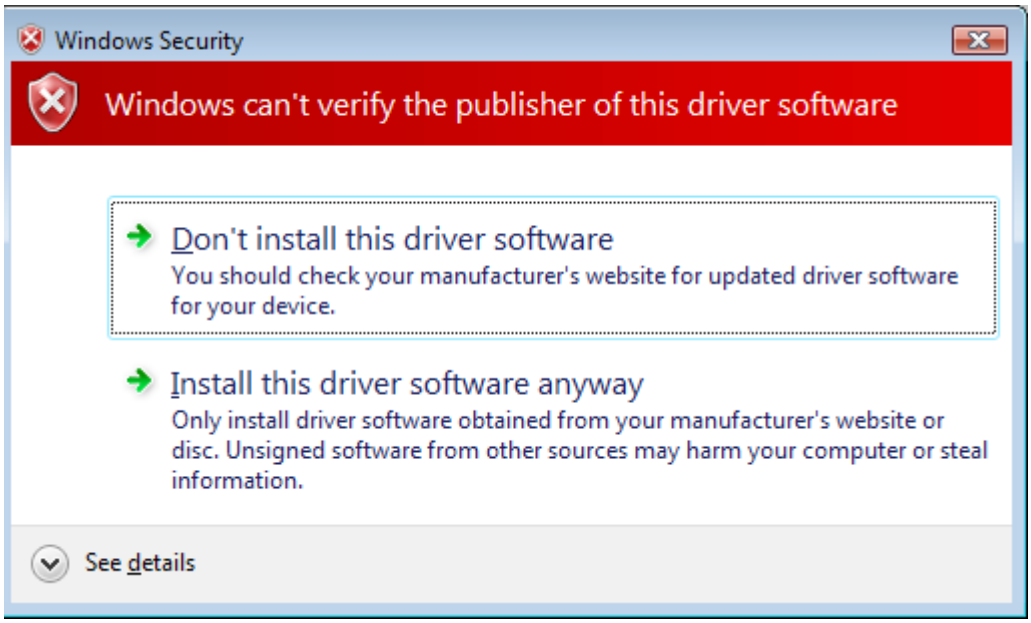


Figure 12 - Unknown Publisher warning

- When the installation is complete, the summary screen is displayed with two options available.
  - ☞ It is always advisable to consult the README file to be aware of the latest changes and any special instructions related to your platform. This is also where you can find important information on the product name and version number, as well as how to reach Hexago Technical Support.
  - ☞ You should also launch the Gateway6 Client Utility to customize the how your Gateway6 Client is configured. Entering a personal userid, password and server details is sufficient for most users.

  Click *Finish* to exit the Gateway6 Client installer.

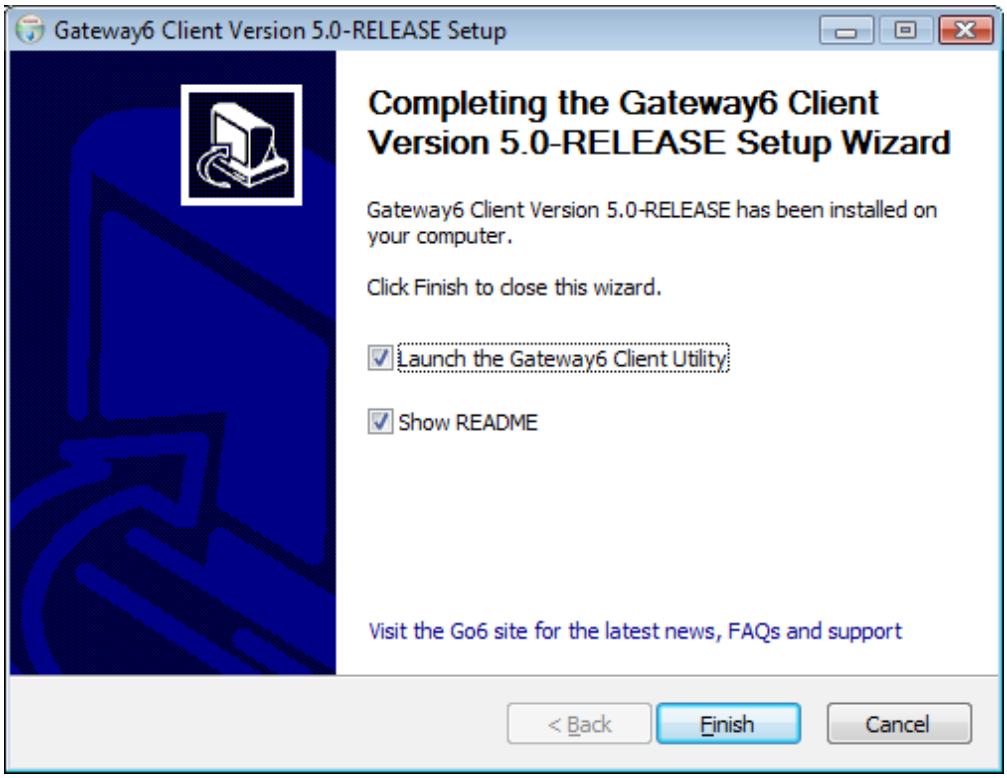


Figure 13 - Installation is complete

The installation utility creates the appropriate files in the destination folder, adds a shortcut to the Windows Start menu and creates a new network connection. If you open *Control Panel* → *Network Connections*, you should see a new "Hexago Virtual Multi-Tunnel Adapter" type connection with an unplugged status.

The Hexago virtual tunnel adapter is used only for v6-in-UDP-in-v4 (*i.e.* for NAT traversal) and v4v6 (DSTM) tunnels. Therefore, the virtual adapter will remain in the 'disconnected' status until one of those tunnel mode is used.

Windows: Overview of the Graphical User Interface

The Gateway6 Client Utility is a Windows application, `gw6c-gui.exe`, that resides on your computer (the local node) in the Gateway6 Client base directory. The Gateway6 Client Utility is a front-end application used to configure the Gateway6 Client and transmit status information to the user. Note that although this Windows interface is not mandatory, it provides a quick and easy way to configure the Gateway6 Client, as well as view important status information.

The utility can be accessed directly via the *Windows Start Menu* (Start → Programs → Hexago → Gateway6 Client → Gateway6 Client Utility). This section describes the user interface.

When you update your settings in the Gateway6 Client Utility, a verification procedure is launched to ensure that the new input is valid and usable by the system. If an error is detected, an exclamation icon will appear next to the invalid data to alert you to the situation. Hovering over the icon will display a tooltip describing why the input value cannot be used. The icon will disappear once the value is corrected. Multiple instances of the icon may appear if several errors are detected.

NOTE: Great care was taken to abide by the Windows Vista User Experience Guidelines developed by Microsoft when creating this utility in order to promote usability and enhance overall quality.

If you configure the Gateway6 Client service to launch automatically upon system startup, you do not need to run the Gateway6 Client Utility interface once the tunnel has been properly configured. However, because it can provide you with valuable status information (ex: whether or not the tunnel has been established, if an error occurred while creating the tunnel, etc.) and real-time feedback whenever needed, it is recommended to let it run anyway. Access the Gateway6 Client Utility in the Windows system tray, located in the lower-right corner of the screen, simply by clicking its icon .

The Gateway6 Client Utility interface is composed of four distinct tabs, each of which is presented below. Press the F1 key at any time while in the application to access the online help for information about specific interface controls.

Basic Tab

The *Basic* tab (shown in Figure 14 on page 34) targets non-technical users who wish to obtain IPv6 connectivity over an existing network with minimal configuration. Conceptually speaking, such connectivity is achieved by means of a *tunnel* between two specific endpoints through which data is transmitted in the desired format. In most instances, an IPv6 tunnel will be created to transmit over an IPv4 network. If you wish to configure the tunnel using more specific or complex options, go instead to the *Advanced* tab. Not all users will need to access these advanced options.

When you first open the application, the icons located at the bottom of the screen will be grayscale images. Upon a successful connection, however, they will become full color to indicate the data tunnel's active status.

If you modify the current configuration of the Gateway6 Client, a message alerting you to the fact that your changes will not be applied until a connection is made will be displayed in bold below the status icons.

NOTE: Because the default tunnel mode is set for IPv6-in-IPv4 connectivity, you must go to the *Advanced* tab and change the tunnel type to "IPv4 in IPv6 (DSTM)" if you are using a native IPv6 network and wish to communicate with an IPv4 network. Most users will not need to make this adjustment.

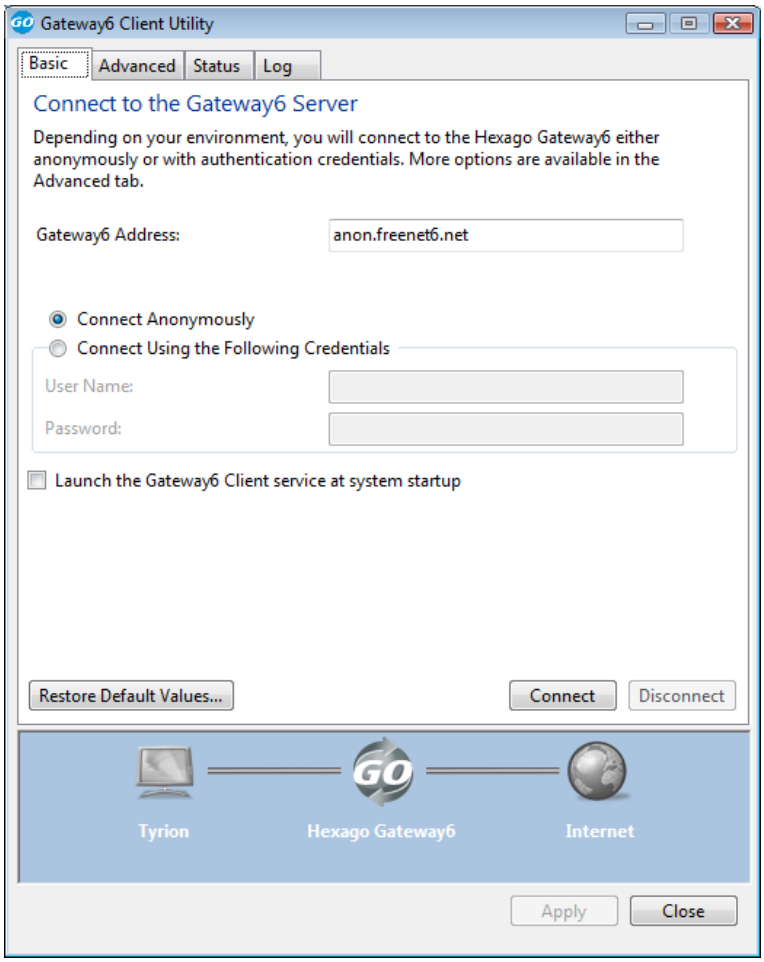


Figure 14 - Gateway6 Client GUI interface — Basic Tab

For your convenience, two connection methods are supported: *anonymous* (no user ID or password) and *authenticated* (user ID and password required). The decision whether to connect anonymously or with authentication depends entirely on your computing environment. Both methods are equally secure concerning data protection. Refer to the *Gateway6 HexOS Configuration Guide* for additional considerations. The main differences between the two connection types are summarized in the table below.

Anonymous Connection	**Authenticated Connection**
IP address obtained from the Gateway6 Server is dynamically renewed (*i.e.* not static)	IP address obtained from the Gateway6 Server is static, meaning it is possible to map it to an explicit domain name (ex: `subdomain.example.org`)

Anonymous Connection	Authenticated Connection
Unable to advertise routing capabilities on the local network	Advertising routing capabilities on the local network is supported
Routing prefixes are not available because they are only assigned to authenticated requesters (see the *Advanced* tab for details)	A routing prefix may be requested. The local computer can then act as an IPv6 router and advertise on the local network (see the *Advanced* tab for details)
No need to enter a valid user name and password in the Gateway6 Client Utility	Valid authentication credentials must be supplied to the Gateway6 Client Utility
No need to register with a forum before gaining access to the server	Registration with a forum is prerequisite to gaining access to the server

NOTE: These differences apply mostly to the free service offered by Hexago (http://www.go6.net). If an ISP requires authenticated sessions, it may not necessarily offer prefix delegation. Other ISPs could potentially offer prefix delegation to anonymous users.

When the Hexago Gateway6 Client service is running, the *Connect* button will be disabled and the *Disconnect* button will become enabled; a message from the notification area will be displayed to inform you of the connection. Furthermore, the status icons located at the bottom of the application dialog will become full color to indicate the tunnel's active status.

When the tunnel has been disconnected, you will be notified by the operating system in a similar fashion. You can verify that the tunnel is inactive by accessing the *Status* tab. The tunnel can be reinstated at any time simply by clicking the *Connect* button; there is no need to exit and restart the application.

Advanced Tab

The *Advanced* tab (shown in Figure 15 on page 37) is used to configure more complex environments and/or tunneling requirements, such as the type of tunnel to create and how it will be authenticated.

The *Tunnel Mode* is the method by which the data tunnel will be negotiated with the Gateway6 server; the default value is “IPv6-in-IPv4 Tunnel.” You can choose between an IPv6-in-IPv4 tunnel (with or without NAT traversal) and an IPv4-in-IPv6 (for DSTM) tunnel.[1] When the default value is selected, the Gateway6 Client will ask the Gateway6 server for an IPv6-in-IPv4 tunnel, without specifying if NAT traversal is required. The Gateway6 server then analyzes the Gateway6 Client's request to determine what kind of tunnel will be established (*i.e.*, native or with NAT traversal).

1. DSTM (*Dual Stack Transfer Mechanism*) is a mechanism that allows completely native IPv6 networks to communicate with IPv4 networks that are yet to be converted to native IPv6 networks. Dual stack means “the ability to provide both IPv4 and IPv6 services.” As such, a node on a native IPv4 network using IPv6 tunneling to acquire IPv6 connectivity (or services) is also a dual-stack node.

The *Tunnel Authentication Method* is linked to the Connection Type from the *Basic* tab. The default tunnel authentication method is "Anonymous" (unauthenticated), but you can change it depending on your system. If you set the tunnel authentication method to "Any," the Gateway6 Client Utility will attempt to connect to the server using the following methods, starting with the most secure ("Pass DSS 3DES-1") and ending with Plain Text.

- Pass DSS 3DES-1
- Digest-MD5
- Plain Text

If none of these methods are successful, you will be notified by the application that the authentication failed; no attempt will be made to connect anonymously. This is also the behavior you can expect if you enter an incorrect password or user name in the *Basic* tab.

A DNS server is used to resolve the domain name to obtain the IPv4 or IPv6 address assigned to the local node by the Gateway6 server. You have the option of specifying multiple FQDN addresses, each separated by a semi-colon (no spaces are required).

The *Connect Using Best Broker* and *Connect Using Preferred Broker* fields are only pertinent when establishing a tunnel via broker redirection. Redirection occurs when the TSP protocol announces multiple sites in order to redirect users in the event of an error or to support multi-site operation. In cases of redirection, the IP address entered in the *Gateway6 Address* field of the *Basic* tab identifies a broker server (or redirector), and not an actual broker that creates tunnels. Communicating with the broker server returns a list of referrals to other brokers with which a tunnel can potentially be created. The Gateway6 Client Utility will proceed to ping each of the brokers on the list to determine which ones are reachable across the Internet, as well as which offer the fastest round-trip time.

- Choosing *Connect Using Best Broker* (the default option) will cause the Gateway6 Client Utility to select the broker that returned the fastest ping value. If the attempt to create a tunnel with the first broker is unsuccessful, the Gateway6 Client will try again using the next entry on the list until either a connection is made or all brokers on the list have failed.
- Choosing the *Connect Using Preferred Broker* option will force the Gateway6 Client Utility to always connect to the specified broker when establishing a tunnel, until configured otherwise. If the attempt to connect to the preferred broker fails for any reason, the application will simply notify you of the unsuccessful connection, and not attempt to connect to another broker.

Not every broker in the referral list will be unconditionally available to create a data tunnel. There are several reasons why a specific broker may refuse a connection request:

- Brokers have a maximum number of permitted connections (ex: in periods of heavy traffic, the broker may not be able to set up a new tunnel because the pool of available resources has been exhausted)
- The broker to which you are attempting to connect may not support the requested tunnel type (ex: you wish to create an IPv4-in-IPv6 tunnel using a broker that only provides IPv6-in-IPv4 tunnels)
- Ambient network conditions may prevent the broker from responding to the connection request

It is recommended to choose the *Connect Using Best Broker* option in most circumstances because the Gateway6 Client Utility is in a better position to determine which broker will be available and offer the best ping. On the other hand, if your topology is based on a static network or you happen to know which broker will consistently deliver the best performance, you may consider choosing the *Connect Using Preferred Broker* option in order to minimize network latency and avoid renumbering.

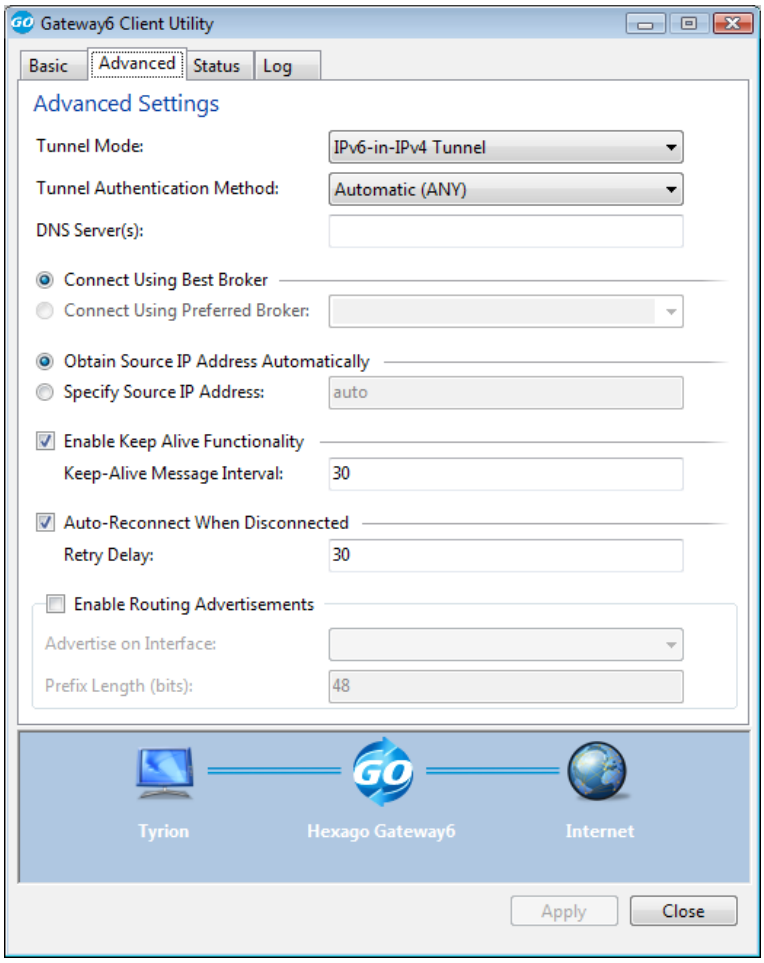


Figure 15 - Gateway6 Client GUI interface — Advanced Tab

The *Obtain Source IP Address Automatically* and *Specify Source IP Address* fields are only pertinent if the local node contains several network adapters. For this reason, the vast majority of users should leave this setting with the default value of *Obtain Source IP Address Automatically*. To specify which network adapter to use when establishing a tunnel with the Gateway6 Client Utility, click the *Specify Source IP Address* option and enter the IP address of the preferred adapter. Both standard IPv4 (ex: `192.168.147.242`) and IPv6 (ex: `2001:5c0::201:6cff:fe84:cc96`) addresses are supported formats for this field.

The *Enable Keepalive Functionality* prompts the Gateway6 Client Utility to send an ICMP packet at the specified interval (expressed in seconds) to the Gateway6 server to ensure the tunnel session remains persistent. The default behavior is to send a keepalive packet every 30 seconds, which is sufficient to keep the session from being dropped in cases of NAT traversal. Not all tunnel servers offer keepalive support; in such a case, you can elect to disable this feature.

If the local node is intended to act as an IPv6 router serving the nodes connected to the same physical IPv4 network, the *Enable Routing Advertisements* option is used to publish the availability of IPv6 connectivity to interested parties on the designated network interface and supply the prefix used to generate their IPv6 addresses. When this option is enabled, the Gateway6 Client Utility expects the server to provide it with a prefix of the specified length (expressed in bits) as part of negotiating the tunnel. Based on the prefix advertised by the

Gateway6 Client Utility, the requesters on the local network requiring IPv6 connectivity can thus calculate their individual IPv6 addresses for access to the tunnel.

Given an address prefix of 48 bits, the 128-bit IPv6 address generated by IPv4 nodes for tunnel access can be broken down as shown below:

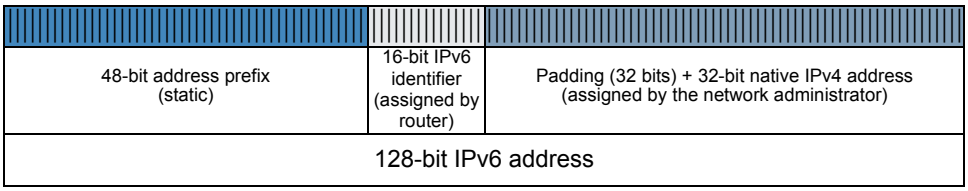


Figure 16 - Bit composition of a prefix-based IPv6 address

Status Tab

The *Status* tab (shown in Figure 17 below) provides a current snapshot of the Gateway6 connection status, as well as detailed usage statistics. None of the presented information can be modified directly in this tab. To update the tunnel configuration, go instead to the *Basic* or *Advanced* tab.

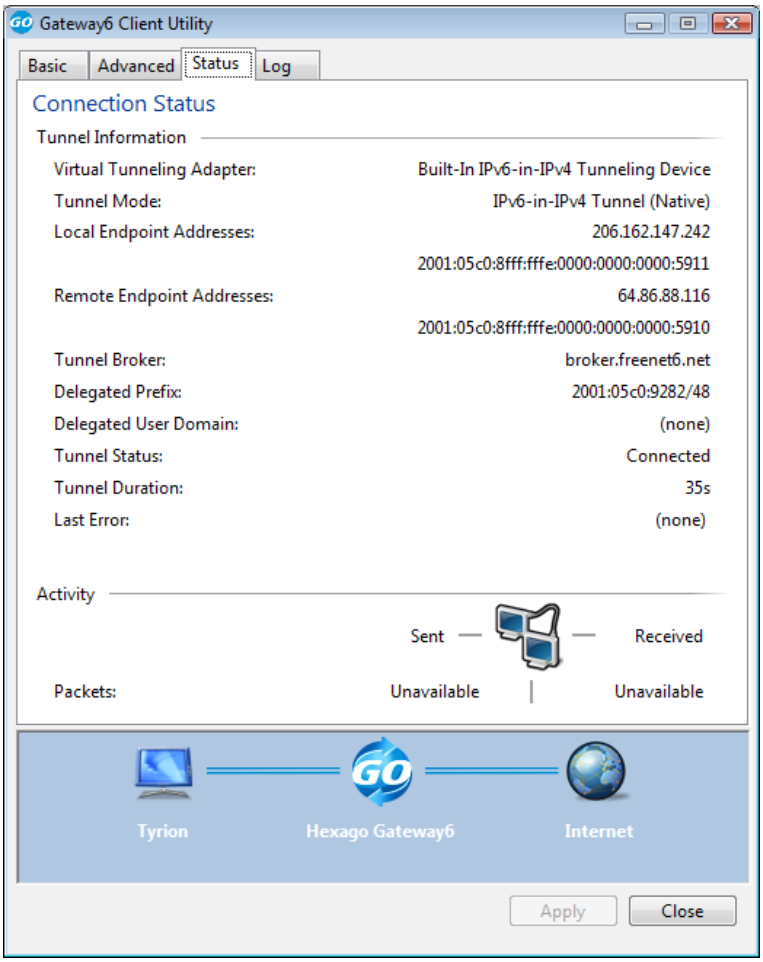


Figure 17 - Gateway6 Client GUI interface — Status Tab

The *Status* tab of the Gateway6 Client Utility can provide the following types of real-time feedback whenever needed.

Virtual Tunneling Adapter

The text that appears here is dependent on how the tunnel was created (*i.e.*, the *Tunnel Mode* field of the *Advanced* tab).

- With native IPv6-in-IPv4 connectivity, you are most likely using Window's built-in virtual adapter, whose default name is “Built-In IPv6-in-IPv4 Tunneling Device”. Bear in mind, however, that the name of this adapter is not static and can be modified in the *Network Connections* tool of the *Windows Control Panel*.
- If you are using either an IPv6-in-IPv4 tunnel with NAT Traversal or DSTM for an IPv4-in-IPv6 tunnel, the name of the Hexago Virtual tunneling device will appear here. You can customize it via the *Network Connections* tool of the *Windows Control Panel*.

Tunnel Mode

Native will always be given as the default tunnel mode, unless your connectivity is comprised of an IPv6-in-IPv4 tunnel with NAT Traversal. NAT traversal is chosen only if the Gateway6 server detects that the Gateway6 Client is located behind a NAT device.

Local Endpoint Addresses

The IPv4 and IPv6 addresses of the local node (this computer).

- In the case of an IPv6-in-IPv4 tunnel, the local endpoint address is comprised of the native IPv4 address and the IPv6 address supplied by the Gateway6 server.
- In the case of an IPv4-in-IPv6 tunnel, the local endpoint address is comprised of the native IPv6 address and the IPv4 address supplied by the Gateway6 server.

Remote Endpoint Addresses

The IPv4 and IPv6 addresses of the tunnel broker. If the Gateway6 server identified in the *Gateway6 Address* field of the *Basic* tab is only a redirection broker, the address of the server that actually negotiated the tunnel will appear here (as opposed to the address of the redirector).

Tunnel Broker

The address of the tunnel broker which negotiated the current tunnel session.

Delegated Prefix

If the Gateway6 Client requested a prefix from the tunnel server in the *Advanced* tab, the exact contents of the prefix are provided here. This field will be set to *(None)* if the tunnel was established anonymously and/or if the *Enable Routing Advertisements* option is disabled.

Delegated User Domain

This information is supplied by the Gateway6 server for use by the local node, based on the user name and the server's domain. No user domain is delegated for anonymous tunnels.

If the local node is a router for the attached subnetwork, devices on the same network are able to obtain individual IPv6 addresses from the local node via the prefix provided by the tunnel server. The delegated user domain provided by the broker is available for these devices to map a userid-based hostname to their IP address to simplify connections (ex: `john.users.mygateway.com`).

Tunnel Status	The message *Connected*, *Connecting*, *Disconnected (idle)* or *Disconnected (error)* will appear here, depending on whether a tunnel is currently instantiated. You can verify this information by examining the color of the status icons located at the bottom of the tab.
Tunnel Duration	The time that has elapsed since the current tunnel was successfully created appears here in the format *AdBhCmDs (*A days, B hours, C minutes, D seconds).
Last Error	A concise textual description of the last error that was detected by the Gateway6 Client Utility. A failed login attempt due to a mistyped password, for example, will cause *Authentication Error* to appear here. If no error has been detected since the application was started, this field will remain blank. For greater detail regarding any errors that may have occurred, either go to the *Log* tab and open the log window, or consult the log file directly.

The lower portion of the *Status* tab deals with data transmission activity that is taking place along the tunnel. Here you can view in real time the amount of data, expressed in packets, that has been sent and received since the current tunnel was established.

NOTE: This information cannot be provided for native IPv6-in-IPv4 tunnels; instead, the message *Not available* will be displayed on-screen.

Log Tab

The *Log* tab (shown in Figure 18 below) is used to configure how the Gateway6 Client Utility manages its record of tunnel negotiation and activity. The log itself is an ongoing textual record of messages exchanged between the Gateway6 Client Utility and the tunnel server as tunnels are created, configured and destroyed. Creating a log file is not mandatory, although you may wish to enable logging for troubleshooting purposes.

The created log files are read by the Gateway6 Client Utility and formatted to help you quickly locate error situations.

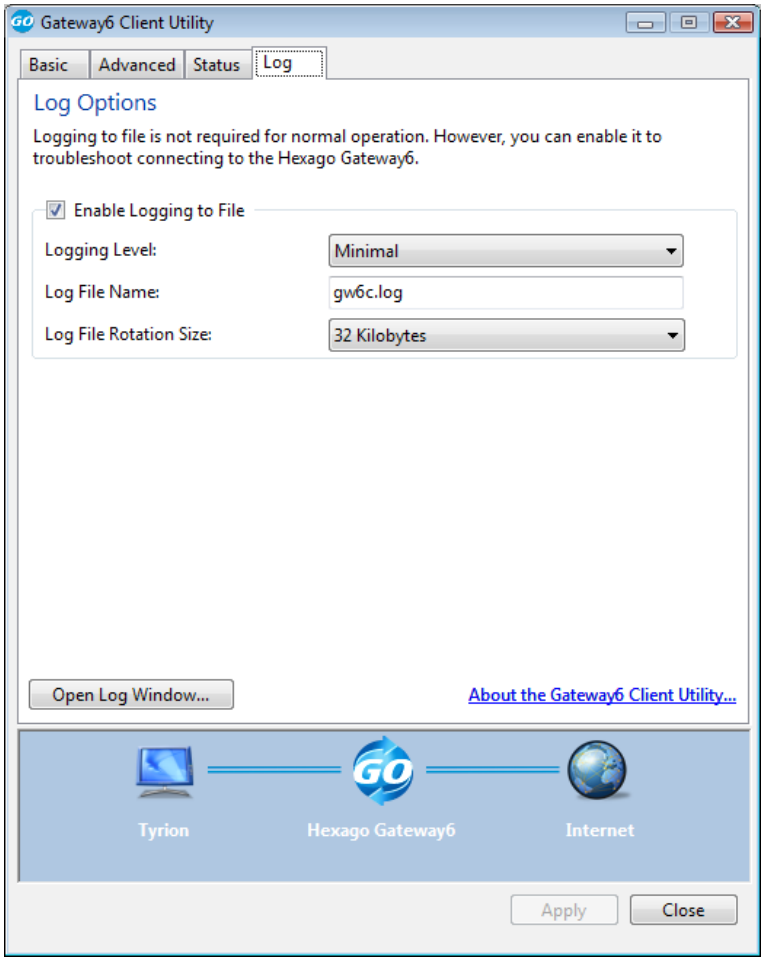


Figure 18 - Gateway6 Client GUI interface — Log Tab

Tick the *Enable Logging to File* check box to make the Gateway6 Client Utility maintain a log file of tunnel activity. This feature is enabled by default, and the *Logging Level* is set to "Minimal". Other logging levels are "Verbose" (handy for troubleshooting purposes) and "Debug" (gives the most debugging information, such as the TSP session XML content).

Enter a specific name for the log file to be created, if desired, in the *Log File Name* field. The default filename is `gw6c.log`. Note that you cannot insert spaces in the filename you choose, and the maximum length of the name is 255 characters.

When the log file reaches the size specified in the *Log File Rotation Size* field, the Gateway6 Client Utility will rename the current log file with the date, then create a new empty file to

resume logging. The default rotation size is 32KB, but you can set it to any of the following sizes, limited only by your computing environment: 16KB, 32KB, 128KB, 1024KB.

Click the **About the Gateway6 Client Utility** link in the lower-right corner of the tab to display important application-related information such as the copyright details, as well as the software version number. This information will be required if you need to contact Hexago Technical Support.

Click the *Open Log Window...* button to view the log files stored on the local machine. A dialog box will open with which you can navigate to the file you wish to consult. Once selected, the log file will be displayed in an HTML browser window, as shown in Figure 19.

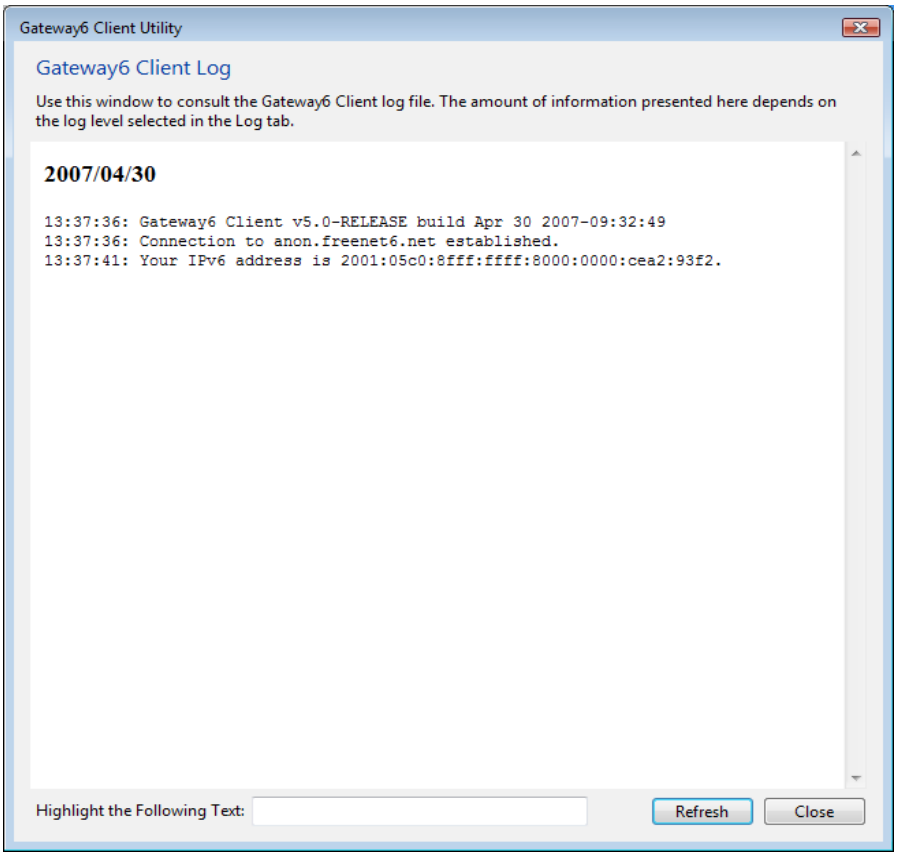


Figure 19 - Gateway6 Client GUI interface — Log Viewer

The log window will present information regarding the Gateway6 connection status, as well as list any errors or inconsistencies that might have occurred. The level of detail recorded in the log files is configured in the *Logging Level* field of the *Log* tab. Bear in mind, however, that once a log file has been recorded, it becomes static and cannot be changed to include more or less detail simply by modifying the *Logging Level* value.

To look for a specific text string in the displayed log file, enter the search text in the *Highlight the Following Text* textbox located at the bottom of the window, then press ENTER (or the *Refresh* button). You can also use the scrollbar on the right side of the text display to navigate through longer log files instead of searching. When you have finished consulting the log file, click the *Close* button to return to the *Log* tab of the Gateway6 Client Utility.

NOTE: You must also close the log window and return to the Gateway6 Client Utility if you wish to view a different log file. Click the *Open Log Window...* button again to view another file.

Click the *Refresh* button to reload the current file in the viewer window. By doing so, any entries added to the log file since it was last refreshed will become visible. The log file will not automatically refresh itself.

Windows: Configuring the Gateway6 Client Service

If the Gateway6 Client has been installed as a Windows service, you may configure its startup behavior in the *Services* tool of the *Windows Control Panel*. Double-click the *Administrative Tools* → *Services* icon (shown in Figure 20 below) to open it and begin editing.

NOTE: You may prefer to complete this task in the *Basic* tab of the Gateway6 Client Utility. To do so within the GUI interface, open the Basic tab and tick the *Launch the Gateway6 Client service at system startup* check box at the bottom of the tab.

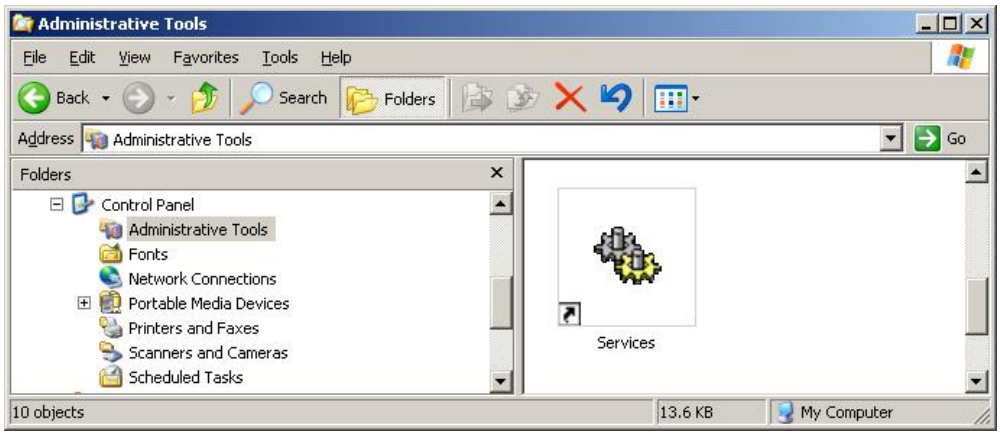


Figure 20 - Windows Control Panel

You can configure the Gateway6 Client to run at startup or wait for manual execution by changing the *Startup type* parameter.

To modify how the Gateway6 Client service is launched, follow the steps listed below:

- Double-click the Hexago Gateway6 Client item
- Choose a new value (Automatic, Manual, Disabled) from the *Startup type* combo box as shown in Figure 21
- Click the *Apply* button to commit your changes
- Press *OK* to dismiss the dialog box and return to the main *Services* window

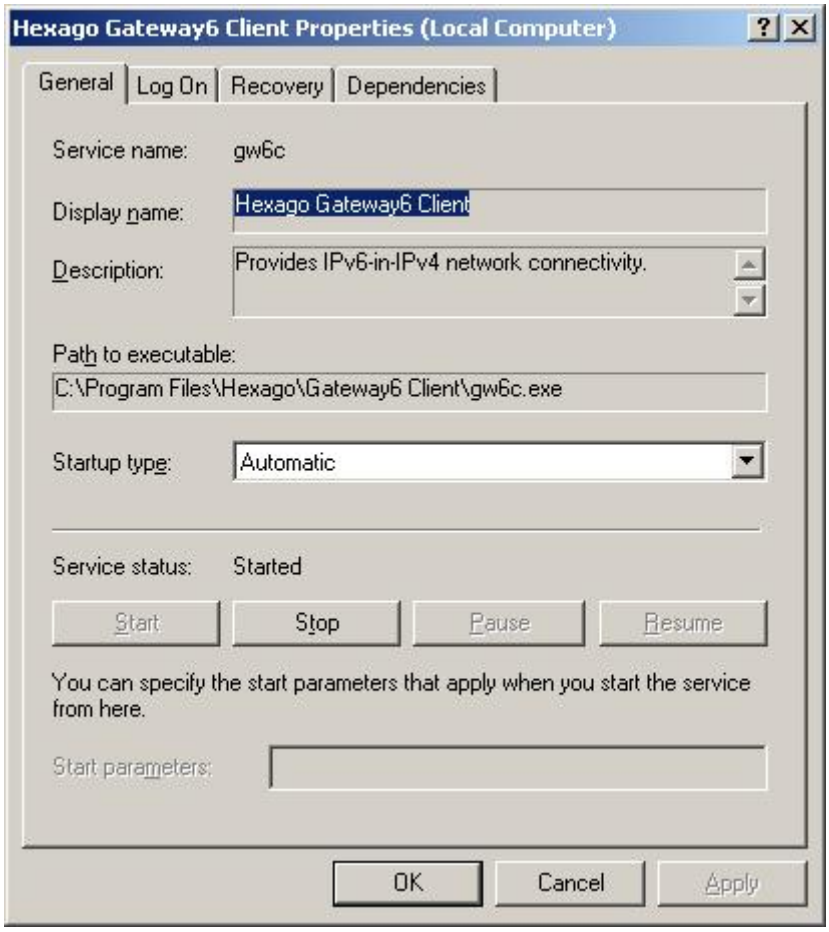


Figure 21 - Modifying the Gateway6 Client service

Windows: Running the Gateway6 Client Manually

If the Gateway6 Client has been installed as a service, it is possible to start and stop the service manually by executing the commands `net {start|stop} gw6c` at the command line.

To run the client manually for debugging purposes, open a *Command Interpreter* window, navigate to the installation directory and type `gw6c`. You can increase the logging level in the `gw6c.conf` configuration file if you require more verbose output. For example, using `log_stderr=3` sets the debugging to its highest level.

Below is an example of running the Gateway6 Client in verbose mode, which shows the TSP session XML exchange between the client and the Gateway6, in addition to the script file execution.

```
C:\Program Files\Hexago\Gateway6 Client>gw6c
Gateway6 Client v5.0-RELEASE build Apr 30 2007-09:32:49
Using TSP protocol version 2.0.1.
Establishing connection to tunnel broker gateway6.hexago.com using
reliable UDP.
```

```
Getting capabilities from server.
RUDP packet 0, RTO 2.000000, sequence 0xf00000f0 timestamp 771.
Reply: RUDP packet 0, RTO 2.000000, sequence 0xf00000f0 timestamp 771.
Connection to broker.freenet6.net established.
Authenticating johnsmith.
Using DIGEST-MD5 authentication mechanism.
RUDP packet 0, RTO 2.000000, sequence 0xf00000f1 timestamp 771.
Reply: RUDP packet 0, RTO 2.000000, sequence 0xf00000f1 timestamp 771.
RUDP packet 0, RTO 2.000000, sequence 0xf00000f2 timestamp 771.
Reply: RUDP packet 0, RTO 2.000000, sequence 0xf00000f2 timestamp 771.
RUDP packet 0, RTO 2.000000, sequence 0xf00000f3 timestamp 771.
Reply: RUDP packet 0, RTO 2.000000, sequence 0xf00000f3 timestamp 771.
Authentication success.
Authentication success.
Using [192.0.2.1] as source IPv4 address.
Sent:
Content-length: 221
<tunnel action="create" type="v6anyv4" proxy="no">
 <client>
  <address type="ipv4">192.0.2.1 </address>
  <keepalive interval="30">
    <address type="ipv6">::</address>
  </keepalive>
 </client>
</tunnel>

RUDP packet 0, RTO 2.000000, sequence 0xf00000f4 timestamp 771.
Reply: RUDP packet 0, RTO 2.000000, sequence 0xf00000f4 timestamp 771.
Received:
200 Success
<tunnel action="info" type="v6v4" lifetime="604800">
  <server>
    <address type="ipv4">192.0.2.6</address>
    <address type="ipv6">2001:0db8:4000:0000:0000:0000:0000:0004</address>
  </server>
  <client>
    <address type="ipv4">192.0.2.1</address>
    <address type="ipv6">2001:0db8:4000:0000:0000:0000:0000:0005</address>
    <address type="ns">johnsmith.broker.freenet6.net</address>
    <router>
      <prefix length="60">2001:0db8:4000:0020:0000:0000:0000:0000</prefix>
    </router>
    <keepalive interval="30">
      <address type="ipv6">2001:0db8:4000:0000:0000:0000:0000:0004</address>
```

```
    </keepalive>
  </client>
</tunnel>

Processing response from server.
Sent:
Content-length: 35
<tunnel action="accept"></tunnel>

RUDP packet 0, RTO 2.000000, sequence 0xf00000f5 timestamp 831.
Reply: RUDP packet 0, RTO 2.000000, sequence 0xf00000f5 timestamp 831.
Obtained tunnel parameters from server. Setting up local tunnel.
Obtained tunnel parameters from server. Setting up local tunnel.
Keepalive interval: 110.
TSP_TUNNEL_MODE=v6v4
TSP_HOST_TYPE=host
TSP_TUNNEL_INTERFACE=2
TSP_HOME_INTERFACE=101
TSP_CLIENT_ADDRESS_IPV4=192.0.2.1
TSP_CLIENT_ADDRESS_IPV6=2001:0db8:4000:0000:0000:0000:0000:0005
TSP_SERVER_ADDRESS_IPV4=192.0.2.6
TSP_SERVER_ADDRESS_IPV6=2001:0db8:4000:0000:0000:0000:0000:0004
TSP_TUNNEL_PREFIXLEN=128
TSP_PREFIX=2001:0db8:4000:0020
TSP_PREFIXLEN=60
TSP_VERBOSE=3
TSP_HOME_DIR=C:\Program Files\Hexago\Gateway6 Client
Executing configuration script:  "C:\Program Files\Hexago\Gateway6
Client\template\windows.bat".
Executing configuration script:  "C:\Program Files\Hexago\Gateway6
Client\template\windows.bat".
Tue 08/05/2007
08:37 AM
Testing IPv6 presence.
Testing Windows NT version.
Cycling the interface.
Configuring V6V4 for XP Service Pack 1 and newer
Overriding TSP_TUNNEL_INTERFACE from 2 to hexago_tunv6
Setting MTU to 1280 on tunnel interface "hexago_tunv6"
Success! Now you're ready to use IPv6 connectivity to Internet IPv6
End of script.
Script completed successfully.
Script completed successfully.
```

```
Your IPv6 address is 2001:0db8:4000:0000:0000:0000:0000:0005.
Your IPv6 prefix is 2001:0db8:4000:0020:0000:0000:0000:0000/60.
The tunnel type is v6v4.
Client proxying is disabled.
The host type is 'host'.
Keepalive initialized with 2001:0db8:4000:0000:0000:0000:0000:0004 as a
peer. Max KA value of 110.
Next KA scheduled in 8.0 seconds.
Next KA scheduled in 13.3 seconds.
```

You can configure the installation utility to run unattended, *i.e.*, without any user intervention, by adding the `/S` flag to the installation command at the command line.

Windows: Uninstalling the Gateway6 Client GUI

If you attempt to install the Gateway6 Client GUI, but the utility is already present on your system, the error message shown below will be displayed.

- Press *OK* to launch the uninstaller to remove the previous version of the software.

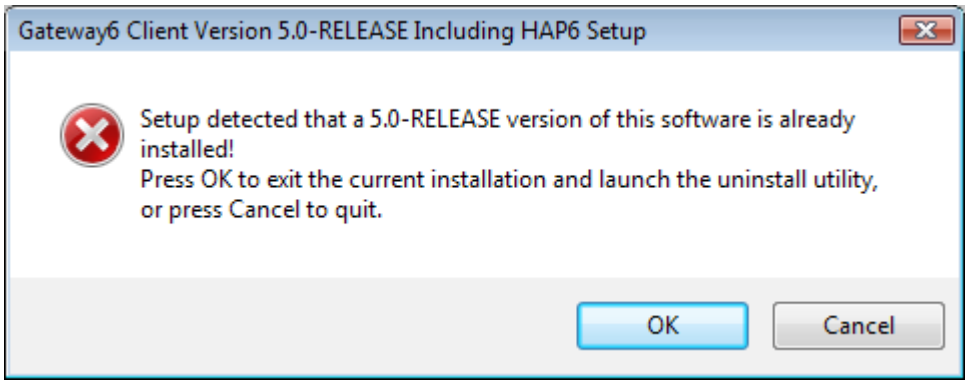


Figure 22 - Gateway6 Client is already installed

The main uninstaller screen will subsequently be displayed, as shown in Figure 23.

- Click the *Uninstall* button to proceed with the software removal.

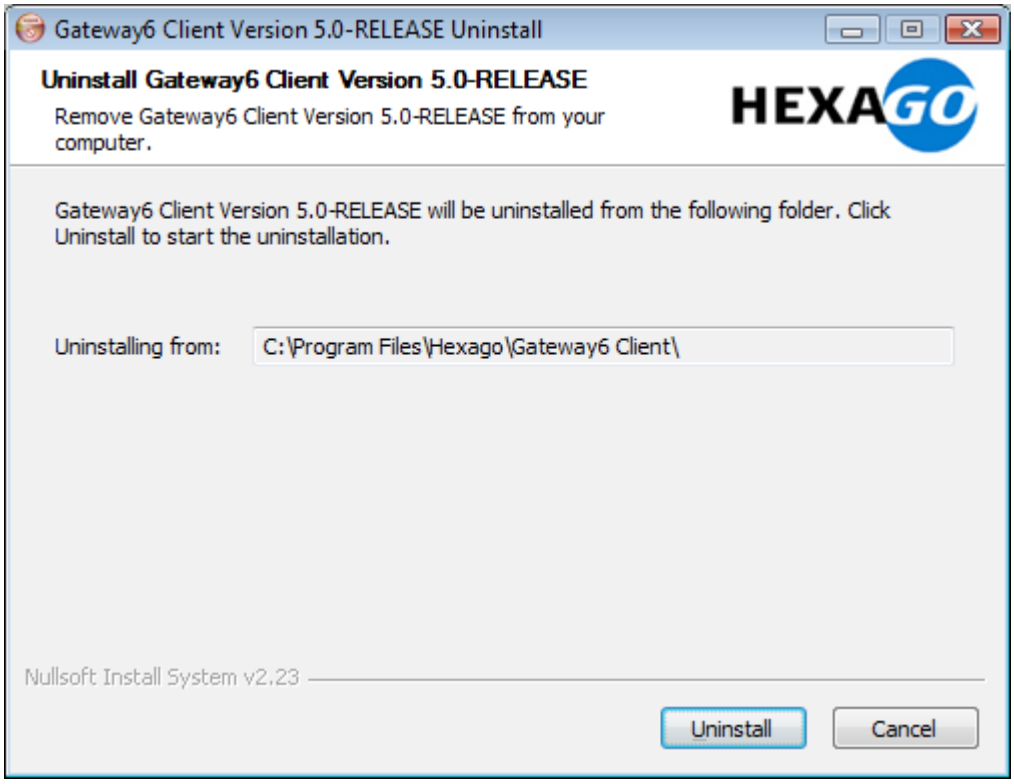


Figure 23 - Uninstalling the Gateway6 Client

- A dialog box inquiring if you wish to retain your Gateway6 Client configuration files will subsequently be displayed. It is recommended to click *Yes* if you plan to reinstall the Gateway6 Client Utility at a later date.

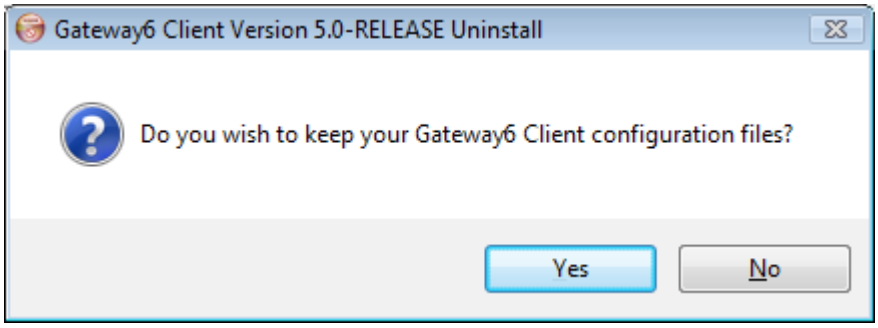


Figure 24 - Modifying the Gateway6 Client service

Regardless of your response, the uninstall will begin at this point. The window shown in Figure 25 on page 49 will be displayed to indicate the progress of the uninstall.

- Once the files have been removed, the progress bar will extend across the full length of the screen and the message “Completed” will appear in the upper portion of the window.

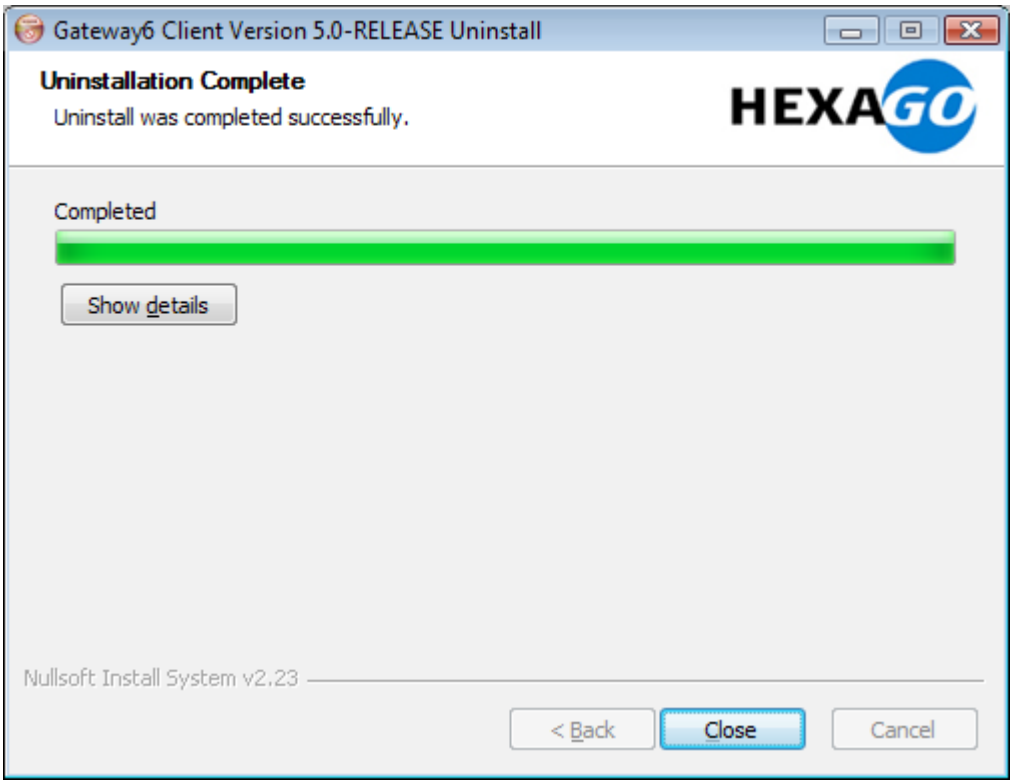


Figure 25 - Uninstall is complete

- Click the *Close* button to dismiss the uninstall utility and return to the Windows Desktop.

Linux

The Gateway6 Client was tested on Redhat 7.2, 8.0 and Fedora Core 1. It is reported to work on most Linux distributions with kernel versions 2.4 or 2.6. Among the Linux distributions, Debian has created a package for the Gateway6 Client called *freenet6* that is available at http://packages.debian.org/stable/net/freenet6.

The ipv6 kernel module should be pre-loaded when running Linux. When using the v6udpv4 encapsulation mode, the `tun` kernel module should also be pre-loaded. Load the modules by entering the commands below:

```
# modprobe ipv6
# modprobe tun
```

The `template` variable in `gw6c.conf` should be set to `linux` to ensure the tunnel is created properly.

The tunnel interface used by the Gateway6 Client under Linux depends on how the operating system is configured. While the default settings should work in most cases, you may need to change the *if_tunnel_v6udpv4* variable in the `gw6c.conf` file if the tunnel is not being established. Examine which interfaces are already in use by the node, then modify the value accordingly.

When the Linux host is a router and advertises prefixes, the `radvd` package must also be present and installed.

Using the Gateway6 Client requires superuser (root) privileges.

FreeBSD

The Gateway6 Client supports FreeBSD 4.X and later.

IPv6 must be enabled before the Gateway6 Client executes. To do so, add the following line to the `/etc/rc.conf` file:

```
ipv6_enable="YES"
```

The `template` variable in `gw6c.conf` should be set to `freebsd` to ensure the tunnel is created properly.

The Gateway6 Client is also included in the FreeBSD ports as `freenet6` under the `net` and `ipv6` subdirectories (`/usr/ports/net/freenet6`, `/usr/ports/ipv6/freenet6`). The port installs the files in the following locations:

- `/usr/local/bin/tspc`
- `/usr/local/etc/tspc.conf`
- `/usr/local/bin/tspc-freebsd.sh`
- `/usr/local/bin/checktunnel.sh`
- `/usr/local/etc/rc.d/freenet6.sh`

If the Gateway6 Client is installed from the TSP distribution instead of the FreeBSD port, then all the configuration files will be located in the installation directory.

Using the Gateway6 Client requires superuser (root) privileges.

Source Code Installation of the Gateway6 Client

Complete the following steps to install the Gateway6 Client from the source code:

1. Retrieve the source code (`*.tgz` or `*.zip` archive) and decompress it to a temporary directory.
2. Go to the `gw6c` directory and type: **`make target=`***`osname`* **`all`**

   where *osname* = `windows`, `freebsd`, `linux`, `darwin`, `netbsd` or `openbsd`

   The system will then proceed to compile the Gateway6 Client.

Executing the Gateway6 Client requires the files listed below:

- The gw6c binary file (`gw6c`) located in the `bin` directory
- A sample `gw6c.conf` file
- The `template` subdirectory containing the operating system scripts

To install the Gateway6 Client in the `/usr/local/gw6c` directory with the necessary files, run the following command:

`make target=`*`osname`* **`installdir=`***`/usr/local/gw6c`* **`install`**

Customizing the Gateway6 Client

When the Gateway6 Client completes its transaction with the broker, the Gateway6 Client calls the shell script (or batch file) specified by the `template` variable of the `gw6c.conf` configuration file, located in the `gw6_dir/template` directory. All the information needed to configure the tunnel is pushed as environment variables from the gw6c program to the shell script (or batch file). The table below lists these environment variables.

Environment Variable Name	Description	Values
`TSP_TUNNEL_MODE`	The tunnel encapsulation mode	`V6V4`, `V6UDPV4`, `V4V6`
`TSP_HOST_TYPE`	The type of node	`HOST`, `ROUTER`
`TSP_TUNNEL_INTERFACE`	Tunnel interface name on the host operating system	*Values are O/S specific*
`TSP_CLIENT_ADDRESS_IPV4`	IPv4 address of the Gateway6 Client	N/A
`TSP_SERVER_ADDRESS_IPV4`	IPv4 address of the tunnel server	N/A
`TSP_CLIENT_ADDRESS_IPV6`	IPv6 address of the Gateway6 Client	N/A
`TSP_SERVER_ADDRESS_IPV6`	IPv6 address of the tunnel server	N/A
`TSP_TUNNEL_PREFIXLEN`	Prefix length used on the tunnel link	N/A
`TSP_HOME_INTERFACE`	In IPv6 router mode, the interface name used to advertise the prefix (`TSP_PREFIX`). The interface should be attached to a link where other IPv6 nodes are auto-configured	N/A
`TSP_HOME_DIR`	Points to the TSP installation directory, which is specified as the `gw6_dir` in the configuration file	N/A
`TSP_PREFIX`	In router mode, the IPv6 prefix allocated by the tunnel broker when using IPv6-in-IPv4 tunnels, or the IPv4 prefix when using IPv4-in-IPv6 tunnels	N/A
`TSP_PREFIXLEN`	In IPv6 router mode, the length of the IPv6 prefix allocated by the tunnel broker when using IPv6-in-IPv4 tunnels, or the IPv4 prefix when using IPv4-in-IPv6 tunnels	N/A
`TSP_VERBOSE`	Level of debug messages. Set to 0 for no messages	`0,1,2,3`

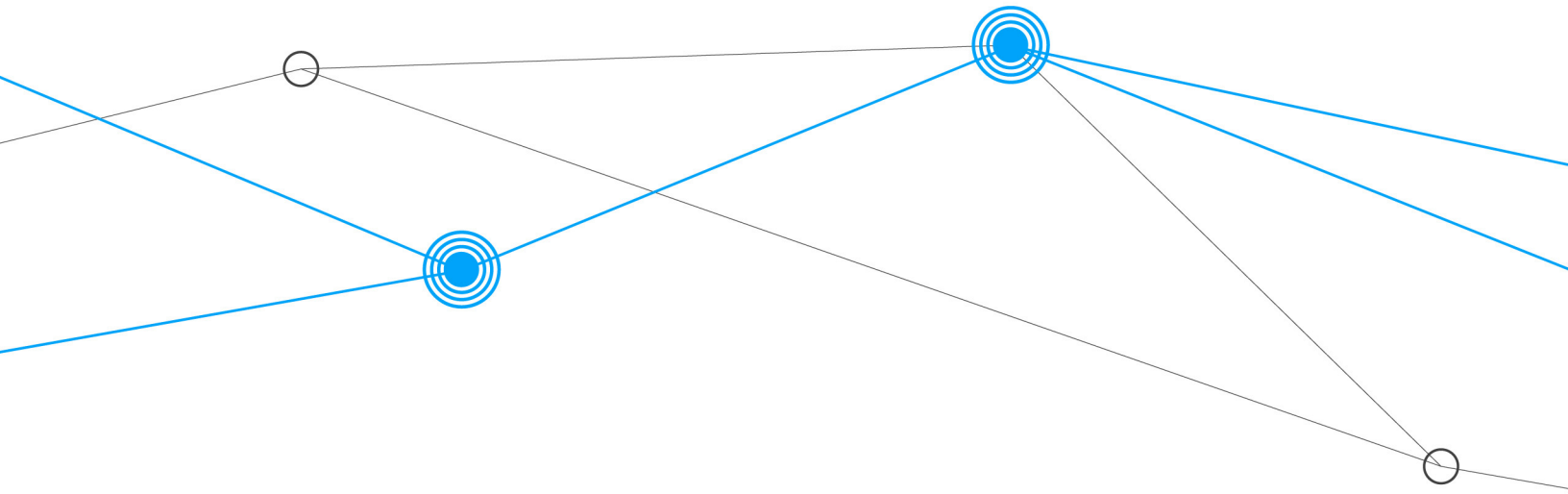
NOTE: These variables are only useful when modifying the script.

Client License

The Gateway6 Client, with the exception of the graphical user interface, is provided under two licenses: open-source and commercial. The open-source license is provided with the source code in the `CLIENT-LICENSE.TXT`. If you wish to use the source code of the Gateway6 Client for commercial purposes that are prohibited in the open-source license, please contact Hexago (info@hexago.com) directly for information on the commercial license.

The graphical user interface is only provided under a commercial license

Copyright Notice



1470 PEEL ST., SUITE 910, MONTREAL, QC CANADA H3A 1T1 T +1.514.906.4786 x1010 F +1.514.904.0468
www.hexago.com support@hexago.com