Framework Deployment Guide

Internet Protocol version 6 (IPv6)

4/4/2026

Internet Protocol version 6 (IPv6)

Contents

IPv6 is a network layer for packet-switched inter-networks. It is designated as the successor of IPv4, the current version of the Internet Protocol, for general use on the Internet.

> **Important**
>
> - This section contains a detailed description of IPv6 and deployment considerations associated with it. See IPv6 vs. IPv4 Overview for information about activating support for IPv6 for a Genesys component. For a list of Framework connections that support IPv6, see IPv6 Support.
> - This section includes material that is freely available on the Internet and from other public sources.

Addressing

The primary change from IPv4 to IPv6 is the length of network addresses. IPv6 addresses are 128 bits long (as defined by RFC 4291), whereas IPv4 addresses are 32 bits. This amounts to an address space for IPv4 of approximately 4 billion addresses, compared to 3.4 x 10^{38} unique addresses for IPv6.

IPv6 addresses are typically composed of two logical parts: a 64-bit network or subnetwork prefix, and a 64-bit host part. This host part is either generated automatically from the MAC address of the interface, or assigned sequentially. Because globally unique MAC addresses offer an opportunity to track user equipment (and therefore users) across IPv6 address changes, RFC 3041 was developed to reduce the chance of user identity being permanently tied to an IPv6 address, thus restoring some of the anonymity existing with IPv4. RFC 3041 specifies a mechanism by which time-varying random bit strings can be used as interface circuit identifiers, replacing unchanging and traceable MAC addresses.

Notation

IPv6 addresses are normally written as eight groups of four hexadecimal digits separated by colons (:). For example:

```
2001:0db8:85a3:08d3:1319:8a2e:0370:7334
```

If one or more four-digit groups is `0000`, the zeros can be omitted and replaced with two colons (::). For example:

```
2001:0db8:0000:0000:0000:0000:1428:57ab
```

can be shortened to

```
2001:0db8::1428:57ab
```

Following this rule, any number of consecutive `0000` groups can be reduced to two colons, as long as there is only one double colon used in an address. Leading zeros in a group can also be omitted (as in `::1` for a `localhost` address). Therefore, the following addresses are all valid and are equivalent:

```
2001:0db8:0000:0000:0000:0000:1428:57ab
2001:0db8:0000:0000:0000::1428:57ab
```

```
2001:0db8:0:0:0:0:1428:57ab
2001:0db8:0:0::1428:57ab
2001:0db8::1428:57ab
2001:db8::1428:57ab
```

Note that having more than one double-colon syntax element in an address is invalid, as it would make the notation ambiguous. For example, the following address:

```
2001:0000:0000:FFD3:0000:0000:0000:57ab
```

abbreviated to

```
2001::FFD3::57ab
```

could imply any of the following:

```
2001:0000:0000:0000:0000:FFD3:0000:57ab
2001:0000:FFD3:0000:0000:0000:0000:57ab
```

or any other similar permutation.

For more information about IPV6 addressing, refer to RFC 4291.

Literal IPv6 Addresses in URLs

In a URL, the IPv6 address is enclosed in brackets. For example:

```
http://[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]/
```

This notation enables the parsing of a URL without confusing the IPv6 address and port number, such as in:

```
https://[2001:0db8:85a3:08d3:1319:8a2e:0370:7344]:443/
```

Additional information can be found in RFC 2732 and RFC 3986.

Network Notation

IPv6 networks are written using Classless Inter-Domain Routing (CIDR) notation.

An IPv6 network (or subnet) is a contiguous group of IPv6 addresses, the size of which must be a power of two. The initial bits of any address in the network are called the prefix, and are identical for all hosts in the network.

A network is denoted by the first address in the network, and the size (in bits) of the prefix (in decimal), separated with a forward-slash (/). For example:

```
2001:0db8:1234::/48
```

stands for the network with addresses

```
2001:0db8:1234:0000:0000:0000:0000:0000
```

through

```
2001:0db8:1234:ffff:ffff:ffff:ffff:ffff
```

Because a single host can be seen as a network with a 128-bit prefix, host addresses are often followed with /128.

Kinds of IPv6 addresses

IPv6 addresses are divided into the following categories (see RFC 4291 - IP Version 6 Addressing Architecture):

- unicast addresses
- multicast addresses
- anycast addresses

Unicast Addresses

A unicast address identifies a single network interface. A packet sent to a unicast address is delivered to that specific computer. The following types of addresses are unicast IPv6 addresses:

- Global unicast addresses
- Link-local addresses (prefix `fe80::/10`): Valid only on a single link; analogous to `169.254.0.0/16` in IPv4
- Unique local IPv6 unicast addresses
- Special addresses (see examples in the following table)

`::/128`	The address with all zeros is an unspecified address, and is to be used only in software.
`::1/128`	The loopback address is a localhost address. It corresponds to `127.0.0.1` in IPv4.
`::ffff:0:0/96`	This prefix is used for IPv4-mapped addresses (see Transition Mechanisms).
`2002::/16`	This prefix is used for 6to4 addressing.
`2001:db8::/32`	This prefix is used in documentation (RFC 3849). Anywhere where an example of an IPv6 address is given, addresses from this prefix should be used.

Multicast Addresses

Multicast addresses are used to define a set of interfaces that typically belong to different nodes instead of just one. When a packet is sent to a multicast address, the protocol delivers the packet to all interfaces identified by that address. Multicast addresses begin with the prefix `FF00::/8`. The second octet identifies the scope of the addresses, that is, the range over which the multicast address is propagated. Commonly used scopes include link-local (`0x2`), site-local (`0x5`) and global (`0xE`).

Anycast Addresses

Anycast addresses are also assigned to more than one interface belonging to different nodes. However, a packet sent to an anycast address is delivered to just one of the member interfaces, typically the closest as defined by the routing protocol. Anycast addresses cannot be easily identified. They have the structure of normal unicast addresses, and differ only by being injected into the routing protocol at multiple points in the network.

Broadcast Addresses

There are no address ranges reserved for broadcast in IPv6. Applications use multicast to the all-hosts group instead. The Internet Assigned Numbers Authority (IANA) maintains the official list of the IPv6 address space. Global unicast assignments can be found on the various Regional Internet

Registries (RIR) or on the Ghost Route Hunter Default Free Prefixes (GRH DFP) pages.

Transition Mechanisms

Until IPv6 completely supplants IPv4, which is not expected to occur in the foreseeable future, a number of transition mechanisms are needed to enable IPv6-only hosts to reach IPv4 services and to allow isolated IPv6 hosts and networks to reach the IPv6 Internet over the IPv4 infrastructure. An overview of some of the various IPv6 transitions that currently exist is provided at: https://www.sixxs.net/faq/connectivity/?faq=comparison.

Dual Stack

Because IPv6 is a conservative extension of IPv4, it is relatively easy to write a network stack that supports both IPv4 and IPv6 while sharing most of the source code. Such an implementation is called a *dual stack*, and a host implementing a dual stack is called a *dual-stack host*. This approach is described in RFC 4213}}.

Most current implementations of IPv6 use a dual stack. Some early experimental implementations used independent IPv4 and IPv6 stacks. There are no known implementations that implement IPv6 only.

Tunneling

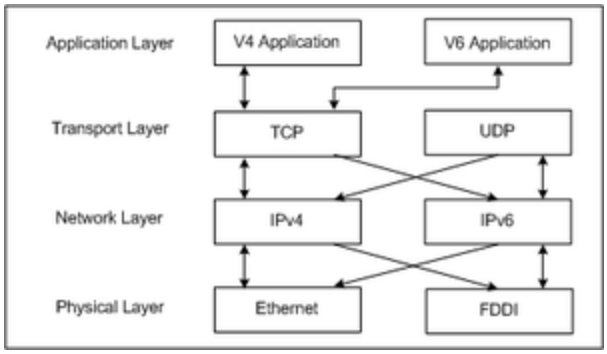
To reach the IPv6 Internet, an isolated host or network must be able to use the existing IPv4 infrastructure to carry IPv6 packets. This is done using a technique referred to as *tunneling*. Tunneling consists of encapsulating IPv6 packets within IPv4, in effect using IPv4 as a link layer for IPv6.

IPv6 packets can be directly encapsulated within IPv4 packets using Protocol 41. They can also be encapsulated within User Datagram Protocol (UDP) packets, for example, to cross a router or Network Address Translation (NAT) device that blocks Protocol 41 traffic. They can also use generic encapsulation schemes, such as Anything In Anything (AYIYA) or Generic Routing Encapsulation (GRE).

Architecture

Dual-Stack IPv6 Implementation

Genesys support for IPv6 relies on true dual-stack IPv6 implementation of the operating system as specified in RFC 3493. Conceptually, the configuration of a dual-stack machine with a v4 TCP and a v6 TCP application is shown in the following figure.



Dual-Stack Architecture

Using this approach, you can write an application that can operate with both IPv4 and IPv6 peers using just one socket. In addition, an application that uses a properly designed Transport Layer library and does not have to operate directly with IP addresses (and other Network Layer elements) may not be aware of the IP version used at all.

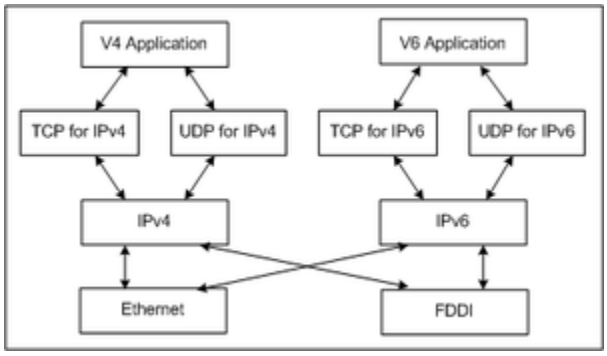
Microsoft Windows Implementation

Microsoft uses slightly different terminology when describing IPv6 architecture. For Microsoft, dual-layer refers to dual-network layers sharing a single transport layer. Dual-stack refers to dual-network layers and dual transport layers, that is, two separate stacks. Only a dual-layer architecture is compliant with RFC 3493.

> **Important**
>
> In this document dual-stack always refers to RFC 3493-compliant implementations, not the Microsoft definition.

Windows Server 2000/2003 and Windows XP

The following figure illustrates Microsoft Windows IPv6 implementation prior to Windows Vista. Microsoft calls this a dual-stack architecture, but it is actually implemented as two separate stacks with separate TCP and UDP paths. This implementation forces an application to open separate sockets to talk to IPv4 and IPv6 peers.



Microsoft IPv6 Stack Prior to Windows Vista

Windows Vista

In Windows Vista, Microsoft calls its next generation IP stack dual-layer architecture, but it is actually a correct dual-stack implementation as described above, where there is only a single transport layer component for TCP and UDP.

Operating Systems Supporting Dual-Stack Architecture for IPv6

Operating system support of dual-stack IPv6 implementation (RFC 3493-compliant) by different operating system platforms is provided in the following table. Refer to platform-specific documentation (including web sites) for additional information about supporting and implementing IPv6.

Operating System	Supporting Releases
AIX	AIX 4.3.3 and later
Linux	kernel 2.6 (Red Hat Enterprise Linux release 4) and later
Mac OS X	Mac OS 10.3 Panther and later
Solaris	Solaris 8 and later
Windows	Windows Vista and later, Windows Server 2008 and later

DNS

Genesys products are using Domain Name System (DNS) resolution of hostnames specified in configuration, and require that the DNS is operating according to the AAAA schema. IPv6 addresses are represented in the Domain Name System by AAAA records (so-called `quad-A records`) for forward lookups; reverse lookups take place under `ip6.arpa` (previously `ip6.int`), where the address space is delegated on nibble boundaries. This scheme, which is a straightforward adaptation of the familiar A record and in-addr.arpa schemes, is defined in RFC 3596. The following table describes the fields in an AAAA record.

Field Name	Description
NAME	Domain name
TYPE	AAAA (28)
CLASS	Internet (1)
TTL	Time to live (seconds)
RDLENGTH	Length of RDATA field
RDATA	String form of the IPv6 address as described in RFC 3513

RFC 3484 specifies how applications should select an IPv6 or IPv4 address for use, including addresses retrieved from DNS. For mixed networks, the DNS must provide both A and AAAA records.

On a historical note, the AAAA schema was one of two proposals at the time the IPv6 architecture

was being designed. The other proposal, designed to facilitate network renumbering, would have had A6 records for the forward lookup and a number of other innovations such as bit-string labels and DNAME records. It is defined in the experimental RFC 2874 and its references (with further discussion of the advantages and disadvantages of both schemes in RFC 3364).

Virtualization

There are no known restrictions from the Genesys side for deploying IPv6 in a virtual operating environment. Check with the documentation specific to the virtual environment you are using for more information and any limitations.

License Control

> **Important**
>
> The information in this section is based on information provided in Flexera documentation, and may be specific to their products. For information about IPv6 support and implementation for other licensing products, consult documentation specific to the product.

Genesys uses FlexLM 9.5 and FlexNet Publisher 11.9-based license control, but only the FlexNet Publisher Licensing toolkit 11.9 supports IPv6. Genesys License Server 8.1 uses FlexNet Publisher 11.9 for all platforms.

The following table summarizes the addressing compatibility of a FlexNet License Server Machine and a Flex-enabled Application Server, as described in this section.

<table>
<tr><th colspan="2"></th><th colspan="4">FlexNet License Server Machine</th></tr>
<tr><th colspan="2"></th><th>IPv4-only</th><th>Dual IPv4/ IPv6 Stack</th><th>IPv6-only</th><th>No Server</th></tr>
<tr><td rowspan="5">Flex-enabled Application Server</td><td>IPv4-only</td><td rowspan="3">Use IPv4 only</td><td>Use IPv4 only</td><td rowspan="2">Not supported</td><td rowspan="2">Use IPv4 only</td></tr>
<tr><td>Dual Stack using IPv4 only</td><td rowspan="2">Use IPv4, IPv6, or both</td></tr>
<tr><td>Dual Stack using IPv4 and IPv6</td><td rowspan="3">Use IPv6 only</td><td>Use IPv4, IPv6, or both</td></tr>
<tr><td>Dual Stack using IPv6 only</td><td rowspan="2">Not supported</td><td rowspan="2">Use IPv6 only</td><td rowspan="2">Use IPv6 only</td></tr>
<tr><td>IPv6 only[a]</td><td>Use IPv6 only</td></tr>
</table>

[a] Genesys does not recommend or support IPv6 environments.

In the license file, an IPv6 address should be defined as the `host` value in the SERVER line. Entries in the license search path that use the `port@host` convention to identify the license server can also specify an IPv6 address as the `host` value.

Deploying License Servers in Mixed Protocol Environments

For FlexNet Publisher components to work properly using IPv6 addresses, all systems in an enterprise (including the network hardware and software) must be configured properly to support communication using IPv6 addresses.

Before testing or deploying a FlexEnabled application that supports IPv6 or IPv4/IPv6 dual communication, make sure that all systems on the network can communicate successfully. If the license server can run under any of the following operating systems:

- Any supported edition of Windows Vista
- Any supported Linux platform
- Any supported Unix platform

it can communicate with FlexEnabled clients using either IPv4 or IPv6, so long as the network is configured properly.

Because these operating systems support dual-layer communication, both IPv4 and IPv6 FlexEnabled clients can communicate with an IPv6 license server. In addition, IPv6 clients can communicate with an IPv4 license server using the IPv4 address.

The FlexNet Publisher license server lmadmin supports both IPv4 and IPv6 clients. If you are using it, you must rename one of your vendor daemon executable files, because separate IPv4 and IPv6 vendor daemons are required.

If the license server runs on Windows XP or Windows Server 2003, there are certain limitations because of the limited dual-layer support on these operating systems (see Windows Server 2000/2003 and Windows XP). IPv4 FlexEnabled clients cannot communicate with an IPv6 license server running on these operating systems. However, IPv6 FlexEnabled clients can communicate with an IPv4 license server running on these operating systems.

If an enterprise runs license servers on Windows 2003 or Windows XP, the license administrators should create and maintain two separate networks - one for IPv6 FlexEnabled clients that will use the IPv6 license server, and one for IPv4 FlexEnabled clients that will use the IPv4 license server.

Using Wildcards in an IPv6 Address

An asterisk (*) can be used as a wildcard character in place of an entire field or on a byte-by-byte basis to specify a range of addresses without having to list them all.

For example, the following feature definition line is locked to four specific addresses:

```
FEATURE f1 myvendor 1.0 1-jan-2010 uncounted \
HOSTID="INTERNET=127.17.0.1,\
INTERNET=2001:0db8:0000:0000:ff8f:effa:13da:0001,\
INTERNET=127.17.0.4,\
NTERNET=2001:0db8:0000:0000:ff8f:effa:13da:0004" \
SIGN="<...>"
```

The following feature definition line specifies an entire range of addresses, including the four specific ones from the line above:

```
FEATURE f1 myvendor 1.0 1-jan-2010 uncounted \
HOSTID="INTERNET=127.17.0.*,\
INTERNET=2001:0db8:0000:0000:*:*:*:000*"\
SIGN="<...>"
```

Genesys IPv6 Support

Genesys supports IPv6 as described in this section.

Common Principles

The implementation of IPv6 in Genesys is based on the following assumptions:

- Dual-stack requirement and backward compatibility
- Dual IPv4/IPv6 server sockets
- IPv4 preference for DNS

Dual-Stack Requirement and Backward Compatibility

Only dual-stack IPv6 implementations are supported. Support of IPv6 on Windows 2002/2003 and XP is not required, while all recent versions of UNIX have dual-stack support already. However, the connection layer must still operate on all other platforms in IPv4 mode only.

On the platforms where IPv6 support is available, the default mode of operation is IPv4 for backward compatibility. IPv6 support must be turned on explicitly by each application using one of the following methods:

- Set the environment variable `GCTI_CONN_IPV6_ON` to 1.
- In the `common` section of the `Application` object's options, set `enable-ipv6` to 1.

Refer to IPv6 vs. IPv4 Overview for more details about enabling IPv6 in Genesys software.

> **Important**
> IPv6 is, by default, not enabled. But once it is enabled using one of the methods described above, it can only be disabled by turning it off in both places-the environment variable and the option. That is, turning it off in one location only disables it if it is not enabled in the other.

Dual IPv4/IPv6 Server Sockets

By default, a server socket opened by a standard method should accept both IPv4 and IPv6 client connections. That is, unless IPv6 is disabled on a particular node, unbound server sockets are opened

with the AF_INET6 family and use the AI_V4MAPPED flag to interact with IPv4 clients. However, a server socket bound to a particular IP address (either IPv4 or IPv6) only accepts a connection of the same IP family.

IPv4 Preference for DNS

Within an application, a name service should be used whenever possible. An AAAA record may return both a IPv4 and IPv6 address for dual stack nodes. For backward compatibility reasons, client connections in this case should prefer IPv4 over IPv6. That preference can be set using the configuration option ip-version.

However, a client connection bound to a particular IP address (either IPv4 or IPv6) can only interact with the server using a connection of the same IP family.

Implementation Characteristics

Individual Genesys components support the following features related to IPv6:

- Full IPv6 support in DNS lookup: Support both AAAA records and DNS over IPv6.
- Transparent server-side socket handling: The existing server-side interface allows IPv6 connections whenever possible using the AI_V4MAPPED flag.
- Transparent client-side connection: The existing client-side connection interface allows IPv6 connections by host name or explicit IP address in text format.
- DNS Lookup modes: Full DNS support using the synchronous method (name lookup using standard system calls) and asynchronous DNS (enabled by the enable-async-dns option in the common section of an Application object's options). Server and client side IPv6 sockets and connections are supported transparently, including hosts being addressed either by name, or by textual IP address in either IPv4 or IPv6 format.
- IPv6-related changes in the configuration environment: Configuration Server keeps IP addresses for all configured hosts, but it is not a replacement for DNS. However it is expected to be affected very little. In particular, a new field for the IPv6 address is not added to the CfgHost structure; while the new configuration option ip-version set at the connection level determines whether the connection uses IPv4 first (4,6; the default), or IPv6 first (6,4). To achieve compatibility with legacy servers (that is, a server without IPv6 support running on a dual-stack host, while IPv6-enabled clients try to connect), the suggested solution is to create an IPv4-only hostname alias for that host.

For more information about the two configuration options, refer to IPv6 vs. IPv4 Overview.

IPv6 Support by Genesys Products

To determine if a Genesys product supports IPv6, refer to the documentation for that product. Framework connections that support IPv6 are listed in IPv6 Support.

Deployment Considerations

When deploying IPv6 in your Genesys environment, you must take into consideration the factors discussed in this section.

Security

Preparation for IPv6 utilization will require careful planning of security measures, because IPv6 presents new challenges compared to IPv4. Some, but not all, of the challenges are discussed in this section.

TLS

In some deployments, multiple hostnames are assigned to a given computer, and are resolved to different IP versions. In this case, the TLS certificate of the given computer will have to be generated for all assigned hostnames.

Refer to the *Genesys Security Deployment Guide* for information about generating certificates.

Firewall and Client-Side Port

Genesys supports fine-grain firewall configuration at the port-level and applied both to incoming client connections and their target server destinations.

In IPv6 deployments, this might become even more valuable, for example, as a countermeasure against Network Discovery (ND) attacks. ND in IPv6 utilizes five different types of ICMPv6 messages for several purposes. ND attacks in IPv6 will likely replace ARP spoofing in IPv4.

Internet Protocol Security

Internet Protocol Security (IPSec) is an optional feature in IPv4, but is mandatory in IPv6. In certain deployments, it could make the use of TLS unnecessary.

DNS Security Extensions

Genesys recommends the use of DNS Security Extensions (DNSSEC), but it is not mandatory. There are no dependencies from the Genesys side.

IP Tunneling

When connecting sites, you may want to use IP tunneling. For example, two sites could be operating in IPv4 mode while the interconnection requires IPv6. In this case, one could consider embedding the IPv4 protocol into an IPv6 connection between sites.

Licensing

The G8.1 License Server (the Genesys vendor daemon) is based on FlexNet Publisher 11.9, and is IPv6 enabled.

However, within G8.1 the IPv6-enabled licensing client libraries (FlexNet Publisher 11.9) are implemented for only the RHEL 5 64-bit, Windows 2008 64-bit, and HP-UX Integrity (Itanium) operating systems. For all other platforms, the G8.1 applications are still using the older client libraries, which are not IPv6 enabled. This is done to provide backward compatibility; otherwise, the deployment of a G8.1 application in an existing environment would have required a complete upgrade of the licensing system.

Therefore, Genesys recommends that IPv4 be used for licensing.

SIP

The SIP protocol can contain explicit IP address values. This creates additional challenges, for example at the NAT level, but also if the same SIP Server instance has to concurrently support multiple SIP interfaces where one is operating in IPv4 mode and another in IPv6 mode.

It is recommended to address those scenarios by using a dedicated SIP Server for IPv4 only and another one for IPv6 only.

You could also consider using available NAT solutions that perform configurable SIP protocol inspection and conversion. One example is F5 Networks Big-IP LTM.

Thin Clients

Some Genesys client applications offer a web browser interface with an HTTP connection to a web server. These connections are under control of the given web technology, and all modern browsers already support IPv6. However, IPv6 must be enabled at both the client computer and server computer sides, and the DNS involved must also support IPv6.

External Interfaces

IPv4 dependencies at external interfaces must be considered. This includes, for example, interfaces to Session Border Controllers (SBC), media gateways, switches, and databases.

Dynamic Runtime Changes

Changes in the IPv4/IPv6 configuration should be performed during maintenance windows, as they will require a restart of impacted processes. These changes will include setting the following:

- Transport parameter `ip-version`
- DNS entries for hostnames
- Local computing node settings

Third-Party Dependencies

Genesys uses several third-party products as part of the suite. The IP capabilities of those products must be considered.