Platform SDK Developer's Guide

IPv6 Resolution

4/9/2025

IPv6 Resolution

Java

Overview

Platform SDK provides two connection configuration options that control IPv4/IPv6 address resolution:

Option Name	Java Constant	Values	Description
enable-ipv6	Connection.ENABLE_IPV6_KEY	`0` (default) `1`	This option enables and disables IPv6 support. When set to `0`, IPv6 support is disabled, even if IPv6 is supported by the platform.
ip-version	Connection.IP_VERSION_KEY	`4,6` (default) `6,4`	Defines the order in which connection attempts will be made to IPv6 and IPv4 addresses. Option values do not contain spaces. This option has no effect if the option `enable-ipv6` is set to `0`. **Note:** This option only applies to clients. **Note:** In Java you can use the predefined value constants: `Connection.IP_VERSION_4_6` or `Connection.IP_VERSION_6_4`

To enable the use of a Netty connection with OIO transport, use one of the following methods:

- start your Java application with the follow JVM option:
  `-Dcom.genesyslab.platform.commons.connection.impl.netty.transport=OIO`
- include the following code at the beginning of your application, before any Platform SDK classes are used:
  ```
  System.setProperty(NettyConnectionFactory.TRANSPORT_TYPE_PARAMETER, "OIO");
  ```

For additional information about working with IPv6, refer to the Networking IPv6 User Guide (https://docs.oracle.com/javase/8/docs/technotes/guides/net/ipv6_guide/index.html).

Code Samples

[+] Genesys Server needs to open the IPv6-only port for listening

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.setIPv6Enabled(true); // cfg.setBoolean(Connection.ENABLE_IPV6_KEY, true) OR
cfg.setOption("enable-ipv6", "1")

Endpoint endpoint = new Endpoint("testServer", "::", 1234, cfg);
ServerChannel server = new ServerChannel(endpoint, new SomeProtocolFactory());
server.open();
```

[+] Genesys Server needs to open IPv4-only port for listening

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.setIPv6Enabled(false); // cfg.setBoolean(Connection.ENABLE_IPV6_KEY, false) OR
cfg.setOption("enable-ipv6", "0")

Endpoint endpoint = new Endpoint("testServer", "0.0.0.0", 1234, cfg);
ServerChannel server = new ServerChannel(endpoint, new SomeProtocolFactory());
server.open();
```

[+] Genesys Server needs to open IPv4/IPv6 dual stack port for listening

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.setIPv6Enabled(true); // cfg.setBoolean(Connection.ENABLE_IPV6_KEY, true) OR
cfg.setOption("enable-ipv6", "1")

Endpoint endpoint = new WildcardEndpoint("testServer", 1234, cfg);
ServerChannel server = new ServerChannel(endpoint, new SomeProtocolFactory());
server.open();
```

[+] Genesys application needs to open connection to IPv6 network interface of backend server

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.setIPv6Enabled(true); // cfg.setBoolean(Connection.ENABLE_IPV6_KEY, true) OR
cfg.setOption("enable-ipv6", "1")
cfg.setIPVersion(Connection.IP_VERSION_6_4); // cfg.setOption(Connection.IP_VERSION_KEY,
Connection.IP_VERSION_6_4) OR cfg.setOption("ip-version", "6,4")

Endpoint endpoint = new Endpoint("testServer", 1234, cfg);
SomeProtocol protocol = new SomeProtocol(endpoint);
protocol.open();
```

[+] Genesys application needs to open connection to IPv4 network interface of backend server

```
PropertyConfiguration cfg = new PropertyConfiguration();
```

```
cfg.setIPv6Enabled(false); // cfg.setBoolean(Connection.ENABLE_IPV6_KEY, false) OR
cfg.setOption("enable-ipv6", "0")

Endpoint endpoint = new Endpoint("testServer", 1234, cfg);
SomeProtocol protocol = new SomeProtocol(endpoint);
protocol.open();
```

[+] Genesys application needs to open connection to IPv6 or IPv4 network interface of backend server, with explicit order of preference (try IPv4 then IPv6)

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.setIPv6Enabled(true); // cfg.setBoolean(Connection.ENABLE_IPV6_KEY, true) OR
cfg.setOption("enable-ipv6", "1")
cfg.setIPVersion(Connection.IP_VERSION_4_6); // cfg.setOption(Connection.IP_VERSION_KEY,
Connection.IP_VERSION_4_6) OR cfg.setOption("ip-version", "4,6")

Endpoint endpoint = new Endpoint("testServer", 1234, cfg);
SomeProtocol protocol = new SomeProtocol(endpoint);
protocol.open();
```

Using the Application Template Application Block

Refer to the Using the Application Template Application Block article for details about defining the IPv6 options in Configuration Manager or loading connection configuration details.

.NET

Overview

Platform SDK provides two connection configuration options that control IPv4/IPv6 address resolution:

Option Name	C# Constant	Values	Description
enable-ipv6	CommonConnection.EnableIPv6Key	`0` (default) `1`	This option enables and disables IPv6 support. When set to `0`, IPv6 support is disabled, even if IPv6 is supported by the platform.
ip-version	CommonConnection.IpVersionKey	`4,6` (default) `6,4`	Defines the order in which connection attempts will be made to IPv6 and IPv4 addresses. Option values do not contain

Option Name	C# Constant	Values	Description
			spaces. This option has no effect if the option `enable-ipv6` is set to `0`. **Note:** This option only applies to clients.

Code Samples

[+] Genesys Server needs to open IPv6-only port for listening

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.IPv6Enabled = true; // cfg.SetBoolean(CommonConnection.EnableIPv6Key, true) OR
cfg.SetOption("enable-ipv6", "1")

Endpoint endpoint = new Endpoint("testServer", "::", 1234, cfg);
ServerChannel server = new ServerChannel(endpoint, new SomeProtocolFactory());
server.Open();
```

[+] Genesys Server needs to open IPv4-only port for listening

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.IPv6Enabled = false; // cfg.SetBoolean(CommonConnection.EnableIPv6Key, false) OR
cfg.SetOption("enable-ipv6", "0")

Endpoint endpoint = new Endpoint("testServer", "::", 1234, cfg);
ServerChannel server = new ServerChannel(endpoint, new SomeProtocolFactory());
server.Open();
```

[+] Genesys Server needs to open IPv4/IPv6 dual stack port for listening

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.IPv6Enabled = true; // cfg.SetBoolean(CommonConnection.EnableIPv6Key, true) OR
cfg.SetOption("enable-ipv6", "1")

Endpoint endpoint = new WildcardEndpoint("testServer", 1234, cfg);
ServerChannel server = new ServerChannel(endpoint, new SomeProtocolFactory());
server.Open();
```

[+] Genesys application needs to open connection to IPv6 network interface of backend server

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.IPv6Enabled = true; // cfg.SetBoolean(CommonConnection.EnableIPv6Key, true) OR
cfg.SetOption("enable-ipv6", "1")
cfg.IPVersion = "6,4"; // cfg.SetOption(CommonConnection.IpVersionKey, "6,4") OR
```

```
cfg.SetOption("ip-version", "6,4")

Endpoint endpoint = new Endpoint("testServer", 1234, cfg);
SomeProtocol protocol = new SomeProtocol(endpoint);
protocol.Open();
```

[+] Genesys application needs to open connection to IPv4 network interface of backend server

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.IPv6Enabled = false; // cfg.SetBoolean(CommonConnection.EnableIPv6Key, false) OR
cfg.SetOption("enable-ipv6", "0")

Endpoint endpoint = new Endpoint("testServer", 1234, cfg);
SomeProtocol protocol = new SomeProtocol(endpoint);
protocol.Open();
```

[+] Genesys application needs to open connection to IPv6 or IPv4 network interface of backend server, with explicit order of preference (try IPv4 then IPv6)

```
PropertyConfiguration cfg = new PropertyConfiguration();
cfg.IPv6Enabled = true; // cfg.SetBoolean(CommonConnection.EnableIPv6Key, true) OR
cfg.SetOption("enable-ipv6", "1")
cfg.IPVersion = "4,6"; // cfg.SetOption(CommonConnection.IpVersionKey, "4,6") OR
cfg.SetOption("ip-version", "4,6")

Endpoint endpoint = new Endpoint("testServer", 1234, cfg);
SomeProtocol protocol = new SomeProtocol(endpoint);
protocol.Open();
```