



# The MAK RTI and IPv6

---

MAK RTI 4.1 or later supports Internet Protocol v4 (IPv4) and IPv6. This paper explains some of the basics of IPv6 and IPv4 and how you can configure the RTI to choose between the two.

## IPv6 Addresses

IPv6 is latest version of the Internet Protocol. It succeeds IPv4. The main issue with IPv4 is the size of the address space. IPv4 addresses are 32 bits. This means that IPv4 supports 4,294,967,296 unique addresses. Unfortunately that is not enough to cover the huge number of addresses needed on the internet. To rectify this, IPv6 addresses are 128 bits. That allows for many more unique addresses.

IPv4 addresses are usually represented in dotted-decimal notation. This was done by breaking the 32 bits comprising an IPv4 address into four 8-bit chunks. Each of these chunks was then displayed in decimal format and separated by a period. This results in the familiar looking IP addresses that most of us are used to: 192 . 168 . 3 . 100

That worked great for IPv4, but to use the same scheme for IPv6 would result in an address four times as long. In attempt to make IPv6 addresses a bit easier to deal with, they used a different notation called colon-hexadecimal notation. The 128 bit address is divided into eight 16-bit chunks. These chunks are displayed as a four digit hexadecimal number and separated by a colon. An IPv6 address looks something like this: `fe80:0000:0000:0000:b472:935a:90c3:6a2e`

That is still cumbersome, and there are a lot of zeros in there. That is true of many IPv6 addresses. So to make it just a bit easier, a series of zeros can be compressed to a double-colon. For example, the address in the previous paragraph would become: `fe80::b472:935a:90c3:6a2e`

You can figure out how many zeros have been compressed by counting the number of displayed bits and subtracting that from the total number. This means that you can only compress one group of zeros in any given address. Otherwise you would not be able to tell how many zeros were compressed in each section. The resulting address is still not as easy to work with as an IPv4 address, but it is just hard to represent 128 bits in an easily readable format.

## Loopback Addresses in IPv6

Most people are familiar with the IPv4 loopback address, 127 . 0 . 0 . 1. In IPv6 the loopback address is `::1`. Remember that the double-colon represents compressed zeros, so this address is actually 127 bits of zeros followed by a single 1.

# The MÄK RTI and IPv6

## Multicast Addresses

Just as in IPv4, IPv6 reserves a range of addresses that can be used for multicast traffic. All addresses that start with “ff” are multicast addresses. So the following are examples of IPv6 multicast addresses:

- ♦ ff02::1
- ♦ ff12::4567:789A:BCDE ff15::8:10

The first 16 bits all have some significance. First are the eight bits of 1’s that tell us it is a multicast address (the “ff” noted above). The next four bits are for flags. The first three flags are generally 0, but the last can change. This bit indicates whether the address is a transient address. Most addresses that you will be choosing yourself for use with the RTI will be transient, so you will probably want to set this to 1, as in the second example above. A 0 indicates a permanent address. These addresses are permanently assigned for specific purposes. For instance, ff02::1 is used to multicast to all link-local nodes.

The next four bits are used to specify the scope of the address. The following are the defined scope values:

- ♦ 1 = Interface-local (a single interface on a node, can be used for loopback) 2 = Link-local (local network segment, not forwarded by routers).
- ♦ 4 = Admin-local.
- ♦ 5 = Site-local.
- ♦ 8 = Organization-local E = Global.

In most cases you will probably use a scope of 2 or 5.

## Broadcast Addresses are not Supported in IPv6

IPv6 did away with the notion of broadcast addresses. A similar result can be achieved by using the ff02::1 multicast address to send to all link-local nodes. In general, though, you are often better off using a normal multicast address.

## Figuring Out Your Computer’s Address

On Windows you can run the ipconfig command. Simply open a command window and type “ipconfig”. You should get a print out that looks something like this:

```
Ethernet adapter Local Area Connection: Connection-specific DNS Suffix . :  
Link-local IPv6 Address . . . . . : fe80::b495:9f25:90e1:6b1d%10 IPv4 Address. . . . .  
 . . . . . : 192.168.3.226  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . :
```

This will print out both the IPv4 and IPv6 addresses of your network device. If it does not support IPv6, only the IPv4 address will be listed.

On Linux the command to use is “ifconfig”. This will again display all of your network interfaces and list their IPv4 and IPv6 addresses.

# The MÄK RTI and IPv6

## Using IPv6 with the MAK RTI

When using IPv6, the MAK RTI supports reliable transport, best effort transport, DDM, DM multi-cast filtering, forwarding, UDP forwarding, and all RTI Assistant features. The only feature that does not support IPv6 is the RTIspy web service.

## Configuring the MAK RTI to use IPv6

There are two primary ways to configure the MÄK RTI: the RTI Assistant GUI and the RID configuration file. If you are using the RTI Assistant (Figure 1), you can simply create a new connection configuration as you would in IPv4, but instead of entering IPv4 addresses, use IPv6 addresses. Choose an IPv6 network interface, and assign an IPv6 best effort (multicast) address. You can then either use the hostname or IPv6 address of the machine where the rtiexec or RTI Forwarder is to be run.

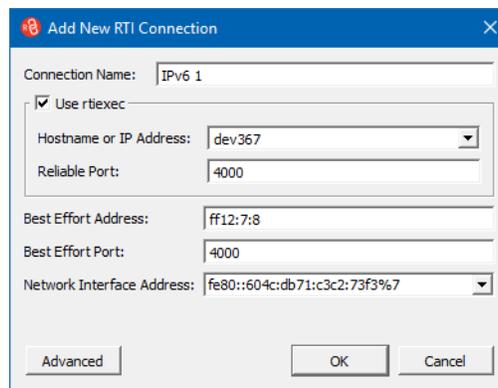


Figure 1. Adding an IPv6 connection

If you are using the RID file, set `RTI_configureConnectionWithRid` to 1. Then set `RTI_destAddrString`, `RTI_tcpForwarderAddr`, and `RTI_networkInterfaceAddr` to use IPv6 addresses.

## The Zone Index

In Figure 1, the network interface address is followed by a %7. That's the zone index, or interface identifier. On Windows it is an integer. All link-local addresses have this special identifier attached to the end to specify which network interface on your machine to use as the outgoing interface. Each interface will have its own zone index.

On Linux, the zone index is the interface name rather than some assigned number, for example %eth0.

# The MÄK RTI and IPv6

---

## Running on a Mixed IPv4 and IPv6 Network

The RTI requires that you choose one protocol when configuring your connection. You cannot use IPv6 for some communications and IPv4 for others. This means that you need to use the same address family (IPv4 or IPv6) for all enabled IP address parameters in your RID file. The only exception is the RTIspy web service which, must always be IPv4 even if all of your other addresses are IPv6.

RTI Assistant will be able to detect active federations on your network using both protocols. The RTI Assistants communicate with each other using IPv4 and IPv6, if available. So they can relay information on any active IPv4 or IPv6 RTI connections in your network. These connections will all be displayed and available for you to choose from. So if you are on a mixed network you can choose what protocol to use when you start up your federate.

## Other MAK Products Support IPv6

VR-Link supports IPv6 for DIS. Since all of the other MÄK products are based on VR-Link, they also support IPv6 for DIS. For HLA it is the RTI that performs all network operations. That means you can upgrade to MÄK RTI 4.1 or later and all other MÄK products you use will immediately have IPv6 support for HLA.

## Where to Learn More about IPv6

Wikipedia has a very good, detailed overview of IPv6, so that is a good starting point:  
<http://en.wikipedia.org/wiki/IPv6>

You can also take a look at RFC 2460, the document developed by the Internet Engineering Task Force (IETF), which describes IPv6:

<http://tools.ietf.org/html/rfc2460>

The IPv6 addressing architecture is discussed in RFC 4291:  
<http://tools.ietf.org/html/rfc4291>

There are also many, many books on the subject. Do a quick search on Amazon and you'll find quite a few. One good reference is The TCP/IP Guide by Charles M. Kozierok. It has a lot of information on both IPv4 and IPv6.