



▶ TCI/IP Support

▶ Handling Data

Sending and Receiving Data

Once your TCPSocket object is connected, you can send and receive data. It makes no difference whether it was originally a server or a client; the mechanisms for data transfer are the same.

The *type* of data that you can send and receive is defined by the Style property which was established when you created the TCPSocket object. The default Style is `'Char'` which allows you to send and receive character vectors. Conversion to and from your `AV` is performed automatically.

If you choose to set Style to `'Raw'`, you can send and receive data as integer vectors whose values are in the range 0 to 255, or -127 to 128. This allows you to avoid any character translation.

If you set Style to `'APL'`, you may transmit and receive arbitrary arrays, including arrays that contain `OR`'s of namespaces. This is normally only appropriate for communicating between two Dyalog sessions. Note however, that there is no mechanism to ensure that both ends of the connection use the same Style.

To send data, you execute the TCPSend method. For example, the following expression will transmit the string "Hello World" to the remote task connected to the TCPSocket object `S0`:

```
2 [NQ'S0' 'TCPSend' 'Hello World'
```

For information on how your data is transmitted, see Output Buffering – Below.

To receive data, you must attach a callback function to the TCPRecv event. Note that for a Stream connection you are not guaranteed to receive a complete message as transmitted by the sender. Instead, the original message may be received as separate packets or several messages may be received as a single packet. This means that you must perform your own buffering and you must implement a specific protocol to recognise message boundaries.

Output Buffering

When you use TCPSend to transmit a block of data, APL copies the data into a buffer that it allocates *outside* your workspace from Windows memory. APL then asks TCP/IP to send it.

Dyalog Ltd

South Barn
Minchens Court
Minchens Lane
Bramley
Hampshire
RG26 5BH
United Kingdom

Phone:

+ 44 (0) 1256 830 030

Fax:

+ 44 (0) 1256 830 031

e-mail:

sales@dyalog.com



▶ **TCP/IP Support**

▶ **Handling Data**

However, the amount of data that can be transmitted in one go is limited by the size of various TCP/IP buffers and the speed of the network. Unless the block is very small, the data must be split up and transmitted bit by bit in pieces. This process, which is handled by APL in the background, continues until the entire data block has been transmitted. It could be several seconds or even minutes after you execute TCPSend before the entire block of data has been sent from your PC.

If in the meantime you call TCPSend again, APL will allocate a second buffer in Windows memory and will only try to send the second block of data when all of the first block has been transmitted.

If you call TCPSend repeatedly, APL will allocate as many buffers as are required. However, if you attempt to send too much data too quickly, this mechanism will fail if there is insufficient Windows memory or disk space to hold them.

If you need to transmit a very large amount of data, you should break it up into chunks and send them one by one. Having sent the first chunk, you can tell when the system is ready for the next one using the TCPReady event. This event is reported when the TCP/IP buffers are free *and* when there is no data waiting to be sent in the internal APL buffers. You should therefore attach a callback to this event whose job is to send the next chunk of data.

However, the amount of data that can be transmitted in one go is limited by the size of various TCP/IP buffers and the speed of the network. Unless the block is very small, the data must be split up and transmitted bit by bit in pieces. This process, which is handled by APL in the background, continues until the entire data block has been transmitted. It could be several seconds or even minutes after you execute TCPSend before the entire block of data has been sent from your PC.

If in the meantime you call TCPSend again, APL will allocate a second buffer in Windows memory and will only try to send the second block of data when all of the first block has been transmitted.

If you call TCPSend repeatedly, APL will allocate as many buffers as are required. However, if you attempt to send too much data too quickly, this mechanism will fail if there is insufficient Windows memory or disk space to hold them.

If you need to transmit a very large amount of data, you should break it up into chunks and send them one by one. Having sent the first chunk, you can tell when the system is ready for the next one using the TCPReady event. This event is reported when the TCP/IP buffers are free *and* when there is no data waiting to be sent in the internal APL buffers. You should therefore attach a callback to this event whose job is to send the next chunk of data.

Dyalog Ltd

South Barn
Minchens Court
Minchens Lane
Bramley
Hampshire
RG26 5BH
United Kingdom

Phone:

+ 44 (0) 1256 830 030

Fax:

+ 44 (0) 1256 830 031

e-mail:

sales@dyalog.com