

# Supported Communication Methods

Entire Connection supports the following methods:

- TN3270(E)
  - Telnet VTxxx
  - BS2000 TCP/IP
  - HLLAPI
  - Serial, VTxxx
  - VT100 Protocol Converter
- 

## TN3270(E)

Entire Connection supports TCP/IP TN3270 and TN3270E communication for display sessions. It also supports TCP/IP TN3270E communication for host printer sessions.

You can use any network adapter that is supported by any TCP/IP stack software which provides the WinSock 2 interface. This mode supports extended attribute bytes (EABs).

The TCP/IP stack software must be installed and active in order to activate terminal emulation.

For IBM host printer emulation, it is necessary to define generic, specific or associated printers on the Telnet server. See your Telnet server documentation for details.

See also: communication parameters for TN3270(E) in the *Overview of Object Properties*.

## Telnet VTxxx

Entire Connection supports VT100, VT220 and VT320 communication with any network adapter that is supported by any TCP/IP stack software which provides the WinSock 2 interface.

The TCP/IP stack software must be installed and active in order to activate terminal emulation.

See also: communication parameters for Telnet VTxxx in the *Overview of Object Properties*.

## BS2000 TCP/IP

This communication method emulates the standard 9750 terminal which is a 24 by 80 characters display without colors. Local printing is not supported. In addition to the standard 9750 terminal features, the following features of the 975x family are supported:

- 80 FTZ per line

- 20 P-keys
- 24 F-keys
- reverse video
- full 9756-type memory support for P-Registers

In Natural environments, the color terminal type 9763 (7 bit) is also supported. As a prerequisite, Natural Version 3 or above must be installed. By default, Natural uses the terminal type 9750 (monochrome). To activate the terminal type 9763, use the following Natural terminal command (either in a screen or in a program):

```
%T=9763
```

When activating the terminal type 9763, it is recommended that you also load the Siemens function keys F1 through F20 using the following Natural terminal command:

```
%KN
```

Entire Connection supports TCP/IP communication with BS2000 hosts with any network adapter that is supported by any TCP/IP stack software which provides the WinSock 2 interface.

The prerequisite on the host side is the communication subsystem BCAM version V.11, which establishes the connection with the host (available within the Siemens product DCAM).

No third-party software is needed for Entire Connection to activate terminal emulation.

To make the terminal emulation key settings similar to those on a BS2000 keyboard, use the predefined key scheme BS2000KEYS.

See also: communication parameters for BS2000 TCP/IP in the *Overview of Object Properties*.

## HLLAPI

Entire Connection supports any communication environment for which HLLAPI software for Windows (32 bit) is available. Support for extended attribute bytes (EABs) depends on the third-party HLLAPI software.

### Notes:

1. Many programs will support extended attribute bytes in DFT mode, but not in CUT mode.
2. Some vendors' APIs must be started before Entire Connection.

To activate terminal emulation, Entire Connection requires the vendor-supplied emulator package and HLLAPI. Install and test the vendor's emulator in your specific communication environment before you start Entire Connection.

Once your vendor-supplied programs are successfully communicating with the host, invoke Entire Connection. If any of the vendor-supplied software required by Entire Connection is removed from memory when Entire Connection is terminated, the vendor-supplied software must be reinvoked each time you wish to invoke Entire Connection.

When using HLLAPI mode to communicate with the mainframe, the `SESSION` command allows you to switch to different logical unit (LU) sessions.

Windows Terminal Services are not supported.

See also: communication parameters for HLLAPI in the *Overview of Object Properties*.

## Serial, VTxxx

Entire Connection supports any serial port (COM1 through COM4). If you are not using a direct connection, an internal or external asynchronous modem is required.

VT100/VT220/VT320 escape sequences are supported (private DEC codes as well as ANSI standard codes for VT100/VT220/VT320). ANSI colors (VT340+) are also supported.

When using Entire Connection to communicate with a VMS or UNIX machine, the line from the PC must be connected to a port on the VMS host or on a terminal server that is either identified as VT100/VT220/VT320 or set to request terminal identification.

To set up Entire Connection for serial communication with a VTxxx host, enable XON/XOFF flow control if it is supported by the host machine to which you are connected. If the host machine supports bidirectional flow control (i.e. an XOFF can be sent from the host to an application and an XOFF can be sent from the application or user to the host), enable both directions.

Windows Terminal Services are not supported.

See also: communication parameters for the VTxxx serial port in the *Overview of Object Properties*.

## VT100 Protocol Converter

A protocol converter converts the 3270 data stream into another communication protocol. There are a number of different communication protocols. Entire Connection, however, supports only the ANSI VT100 protocol. Non-standard extensions to the ANSI VT100 protocol are not supported.

Extended attribute bytes (EABs) are not supported.

Most protocol converters convert normal 3270 field types and then assign VT100 attributes to each field type. You can define the colors you want to use for displaying the attributes.

See also: communication parameters for VT100 Protocol Converter in the *Overview of Object Properties*.

### To set up your protocol converter for use with Entire Connection

1. Configure the protocol converter for VT100 mode.
2. Set the protocol converter to enable XON/XOFF flow control, if available. If the protocol converter supports bidirectional flow control (i.e. an XOFF can be sent from the protocol converter to the application and an XOFF can be sent from the application or user to the protocol converter), both directions should be enabled.

3. Disable any status line display generated by the protocol converter.
4. Import the terminal function code table for your protocol converter.
5. Use one of the supplied *.key* files or create a *.key* file that contains all valid escape code sequences required by your protocol converter.

If none of the supplied *.key* files is compatible with your protocol converter, you must either create a new *.key* file or modify one of the supplied *.key* files.

6. Check each escape sequence in the *.key* file to ensure that it corresponds to the escape sequence required by your particular protocol converter.

Because most protocol converters may be customized when installed, this applies to both supplied and customized *.key* files. It is important to verify that the escape sequences needed by the protocol converter have not been modified.

If you are using multiple protocol converters and different sets of escape sequences are required among them, you must create a unique *.key* file for each protocol converter.

7. Import each *.key* file to internally store this information for terminal emulation purposes.
8. Define all required communication parameters.