TCP/IP Option Administration Guide

Entire Net-Work TCP/IP Option Administration

Version 6.5.1

October 2020

ADABAS & NATURAL

This document applies to Entire Net-Work TCP/IP Option Version 6.5.1 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: WTC-DOC-651-20191115

Table of Contents

Preface

This document provides information for administrators responsible for configuring and running the Entire Net-Work TCP/IP and Simple Connection Line Drivers once Entire Net-Work is installed.

Note: The TCP/IP and Simple Connection Line Drivers are combined under the Software AG product option called the Entire Net-Work TCP/IP Option (product code WTC), which is an add-on to the Entire Net-Work product and must be ordered separately.

This documentation is organized as follows:

Release Information	Provides a high-level explanation of this release of Entire Net-Work TCP/IP Option.
TCP/IP Overview	Provides general information about TCP/IP and its use by Entire Net-Work. This information is applicable to all Entire Net-Work line drivers that use TCP/IP.
Installing Entire Net-Work TCP/IP Option	Describes the installation of Entire Net-Work TCP/IP Option.
TCP/IP Line Driver Administration	Provides information for administrators responsible for configuring and running the Entire Net-Work TCP/IP line driver once Entire Net-Work is installed.
Simple Connection Line Driver Administration	Provides information for administrators responsible for configuring and running the Entire Net-Work Simple Connection Line Driver once Entire Net-Work is installed.

1 Conventions

Notation "*vr* SP *s*", *vrs*, or *vr*: When used in this documentation, the notation "*vr* SP *s*", *vrs*, or *vr* stands for the relevant version, release, and system maintenance level numbers. For further information on product versions, see *version* in the *Glossary*.

This document covers the following topics:

Syntax Conventions

The following table describes the conventions used in syntax diagrams of Entire Net-Work statements.

Convention	Description	Example
uppercase, bold	Syntax elements appearing in uppercase and bold font are keywords. When specified, these keywords must be entered exactly as shown.	**DRIVER TCPI [DRVCHAR =** *driver-char* **| #]** The syntax elements DRIVER, TCPI, and DRVCHAR are Entire Net-Work keywords.
lowercase, italic, normal font	Syntax elements appearing in lowercase and normal, italic font identify items that you must supply.	**DRIVER TCPI [DRVCHAR =** *driver-char* **| #]** The syntax element *driver-char* identifies and describes the kind of value you must supply. In this instance, you must supply the special character used to designate that an operator command is directed to the TCP/IP line driver, rather than to a specific link.
underlining	Underlining is used for two purposes: 1. To identify default values, wherever appropriate. Otherwise, the defaults are explained in the accompanying parameter descriptions. 2. To identify the short form of a keyword.	**DRIVER TCPI [DRVCHAR =** *driver-char* **| #]** In the example above, # is the default that will be used for the DRVCHAR parameter if no other record buffer length is specified. Also in the example above, the short version of the DRVCHAR parameter is D.

Convention	Description	Example
vertical bars (|)	Vertical bars are used to separate mutually exclusive choices. **Note:** In more complex syntax involving the use of large brackets or braces, mutually exclusive choices are stacked instead.	DRIVER TCPI API = { BS2 | CNS | EZA | HPS | OES } In the example above, you must select BS2, CNS, EZA, HPS, or OES for the API parameter. There are no defaults.
brackets ([ ])	Brackets are used to identify optional elements. When multiple elements are stacked or separated by vertical bars within brackets, only one of the elements may be supplied.	DRIVER TCPI [DRVCHAR = *driver-char* | #] In this example, the DRVCHAR parameter is optional.
braces ({ })	Braces are used to identify required elements. When multiple elements are stacked or separated by vertical bars within braces, one and only one of the elements must be supplied.	DRIVER TCPI API = { BS2 | CNS | EZA | HPS | OES } In this example, one of the following values is required for the API parameter: BS2, CNS, EZA, HPS, or OES.
other punctuation and symbols	All other punctuation and symbols must be entered exactly as shown.	LINK *linkname* TCPI [NETADDR = *n1.n2.n3.n4*] [,] [-] In this example, the periods must be specified in the IP address. In addition, options must be separated by commas and dashes should be used as needed to indicate that parameter settings continue on the next line.

Syntax Rules

The following rules apply when specifying Entire Net-Work parameter statements:

- Each Entire Net-Work parameter statement occupies positions 1 - 72 of at least one line.
- The statement type (NODE, LINK, TRANSDEF, or DRIVER) must be specified as the first non-blank item on the statement.
- The node name, driver name, translation definition function, or link name follows the statement type, separated by at least one blank (space).

- Keyword parameters may be specified following either the node name on NODE statements or the driver name on DRIVER and LINK statements. Keyword parameters are separated from their arguments by an equal (=) sign, and from other keyword parameters by at least one blank (space) or a comma (,).
- When the acceptable values for a parameter are Y and N (yes and no), any other value is treated as an N, unless there is a documented default, and processing continues without any warning.
- When the acceptable values for a parameter fall within a range (e.g., 1 - 2147483647) and a value outside the range is specified, the value is automatically reset to the maximum value within the range, unless documented otherwise for the parameter. Processing continues without any warning.
- A statement can be continued beginning in any column of the next line by specifying a dash (-) as the last nonblank character in any column of the current line, before column 73.
- Comment lines begin with an asterisk (*) in position 1 and can be inserted anywhere in the statement sequence.
- Some keywords may require a list of subparameters separated by commas; the list must be enclosed in parentheses ( ) unless only the first subparameter is to be entered. Omitted ("defaulted") subparameters must be represented by placeholder commas if subsequent parameters are to be entered. The following are examples of correct subparameter strings:

```
KEYWORD=(value1,value2,value3)
KEYWORD=(value1,,value3)
KEYWORD=(,,value3)
KEYWORD=(,value2)
KEYWORD=value1
```

- Hexadecimal keyword values can be entered by prefixing the value with an "X". For example:

```
LINK . . . ADJID=X0064, . . .
```

2 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at **http://documentation.softwareag.com**. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at **https://empower.softwareag.com/**.

You can find product information on the Software AG Empower Product Support website at **https://empower.softwareag.com**.

To submit feature/enhancement requests, get information about product availability, and download products, go to **Products**.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the **Knowledge Center**.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at **https://empower.softwareag.com/public_directory.asp** and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at **http://techcommunity.softwareag.com**. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

3 Entire Net-Work TCP/IP Option Release Information

Read this chapter carefully before installing and using Entire Net-Work TCP/IP Option version 6.5.

For complete information about the installation media for Entire Net-Work 6.5 and about new and changed features for Entire Net-Work 6.5, read *Entire Net-Work 6.5 Release Information* in the *Entire Net-Work Release Notes*.

Enhancements for Entire Net-Work TCP/IP Option 6.5

The following enhancements have been made to this release of Entire Net-Work TCP/IP Option.

New and Changed Simple Connection Line Driver (TCPX) DRIVER and LINK Parameters

The following table summarizes the new Simple Connection Line Driver (TCPX) DRIVER and LINK parameters introduced in Entire Net-Work 6.5:

Parameter	DRIVER or LINK Statement?	New or Changed?	Description	Introduced in Release
ADI	DRIVER	New	This parameter was added to the TCPX line driver in this release. It specifies whether Adabas Directory Server (ADI) support is enabled or disabled.	6.5 SP1
ADIHOST	DRIVER	New	This new parameter specifies the hostname of the Adabas Directory Server (ADI). The hostname is used to attempt to acquire the TCP/IP address of the system where the ADI resides.	6.5 SP1
ADIPART	DRIVER	New	This new optional parameter specifies the partition name to be used with the Adabas Directory Server (ADI). If specified, the partition name will be included in all target entries added to the ADI by this session. Partitions are used to restrict database access; when an application queries the ADI for a target and specifies a partition, only entries with the same partition name are returned. Likewise, if the query does not specify a partition, only entries that do not have a partition are returned.	6.5 SP1
ADIPORT	DRIVER	New	This new parameter specifies the port number used to communicate with the Adabas Directory Server (ADI).	6.5 SP1

For more information about TCPX DRIVER statement parameters, read ***TCPX DRIVER Statement***, elsewhere in this guide. For more information about TCPX LINK statement parameters, read ***TCPX LINK Statement***, elsewhere in this guide.

Requirements and Restrictions

Compatibility with Earlier Versions of the Line Drivers

The TCP/IP line driver supports connections with TCP/IP line drivers from all prior versions of Entire Net-Work.

The Simple Connection Line Driver supports the same connections that were supported in all prior versions of Entire Net-Work.

TCP/IP Transport Providers Supported

Entire Net-Work 6.5 supports the following levels of TCP/IP transport providers:

Platform	TCP/IP Protocol Supported
z/OS	IBM TCP/IP for z/OS Version 1.10, 1.11, and 1.12.
z/VSE	IBM and Connectivity Systems TCP/IP z/VSE Version 4.2 and 4.3.
BS2000/OSD Sockets Subsystem	Sockets Version 2.0 and above. For Sockets 2.2 and above, the SOC6 subsystem will be used, otherwise the SOCKETS subsystem is used. When accessing via IPV6 addressing, a SOC6 subsystem with Sockets 2.5 is the minimum requirement.

4 History of TCP/IP

In the late 1960s, the Defense Advanced Research Projects Agency (DARPA) of the United States Department of Defense initiated a project to interconnect computers. A small network was created that interconnected four computers, resulting in the creation of ARPANET. Colleges, universities, businesses, government offices, and military installations slowly began gaining access to ARPANET, and two large-scale communication backbone networks developed:

- the INTERNET for use by the government, colleges, and businesses; and
- the MILNET for exclusive military use.

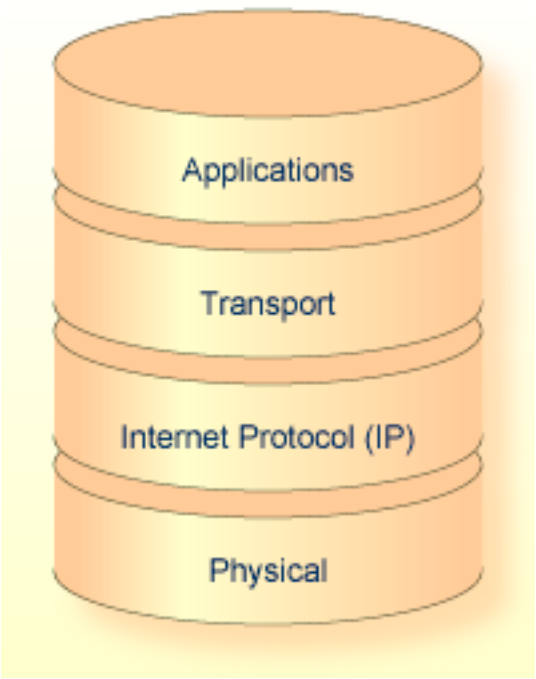
DARPA created a layered protocol stack in which each layer operates independently according to a set of prescribed rules known as Requests for Comments (RFCs) and Military Standards (MILSTDs). These are maintained and updated by the Department of Defense National Information Center (NIC) and the Internet Advisory Board (IAB).

Since its creation, the Transmission Control Protocol/Internet Protocol (TCP/IP) has been widely adopted as a standard in commercial applications, office automation, and personal computing networks. A copy of TCP/IP is integrated into the Berkeley UNIX Operating System, making TCP/IP the common protocol among UNIX systems.

5 TCP/IP Protocol Stack

As shown in the following diagram, the TCP/IP protocol stack contains four layers:

- Physical layer
- Internet Protocol (IP) layer
- Transport layer, comprising
  - Transmission Control Protocol (TCP); and
  - User Datagram Protocol (UDP)
- Applications layer



Physical Layer

At the bottom of the stack is the physical layer, which deals with the actual transmission of data over physical media such as serial lines, Ethernet, token rings, FDDI rings, and hyperchannels. Messages can also be sent and received over other, non-physical access methods such as VTAM/SNA.

Internet Protocol (IP) Layer

Above the physical layer is the Internet Protocol (IP) layer, which deals with the routing of packets from one computer to another. The IP layer

- determines which lower level protocol to use when multiple interfaces exist.
- determines whether to send a packet directly to the host or indirectly to a relay host known as a router.

  When a packet is larger than the size supported by the physical medium, the IP layer breaks the packet into smaller packets, a process referred to as "fragmentation and reassembly".
- provides some control services for packets, and ensures that they are not sent from router to router indefinitely.

However, the IP layer does not keep track of a packet after it is sent, nor does it guarantee that the packet will be delivered.

Transport Layer

Above the IP layer is the transport layer, which contains the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) guarantees that data sent by higher levels is delivered in order and without corruption. To accomplish this level of service, the TCP implementation on one computer establishes a session or connection with the TCP implementation on another computer. This process is referred to as Connection Oriented Transport Service (COTS).

After a session is established, data is sent and received as a stream of contiguous bytes; each byte can be referenced by an exact sequence number. When data is received by the remote TCP, it sends an acknowledgment back to the local TCP advising it of the sequence number of the last byte of data received. If an acknowledgment is not received, or if an acknowledgment for previously sent data is received twice, the local TCP retransmits the data until it is all acknowledged. The remote TCP discards any bytes that are received more than once.

All data sent and received by TCP is validated for corruption using checksums. Whenever a checksum is incorrect, the bad data is discarded by TCP, and the correct data is retransmitted until it is accurately received.

User Datagram Protocol (UDP)

Unlike the TCP, the User Datagram Protocol (UDP) transmits and receives data in packets (datagrams), and delivery is not guaranteed. The contents of the data can be sent with or without a checksum. The use of checksums varies widely from one implementation to another.

Applications Layer

Above the transport layer is the applications layer, which contains both general applications and function libraries for use by applications.

Some general applications that run over TCP include

- File Transfer Protocol (FTP);
- remote terminal emulation (TELNET in line mode, TN3270 in full screen);
- Electronic Mail (SMTP); and
- Entire Net-Work.

Some general applications that run over UDP are

- Network File Server (NFS); and
- Domain Name Server (DNS).

Interface with TCP and UDP

Function libraries provide routines to simplify the interface between applications and TCP/UDP of the Transport layer:

- The most common function library is known as Sockets, which allows an application written in C to access TCP as if it were just another stream input/output device.
- Another function library that is less commonly used is Remote Procedure Call (RPC), which allows applications to make calls to functions that are located in another application on a different computer.

The environment in which an application runs often dictates the interface used between it and TCP or UDP:

- Most UNIX, OS/2, and Windows applications are written in C and utilize a direct socket interface.
- On IBM mainframes and other systems based on the same architecture such as Fujitsu Technology Solutions, applications are often written in S/390 assembler, and use either a pseudo-socket interface or an application program interface (API) to gain access to the TCP/IP protocol stack.

Ports

The interface that exists between an application and TCP is referred to as a port. Ports are classified as server ports and client ports:

- Server ports are generally ports on which the application "listens" for incoming connections to be made.
- Client ports are generally ports on which the application "connects" outwardly to a server port.

  An application may control multiple client ports and server ports simultaneously.

  Each port is identified by a port number, which ranges from 1 to 65535.
- The port number used by client ports usually has no significance and is often assigned by TCP.
- Server port numbers, however, are usually required to be "well known"; that is, the client must know which port the server is listening on when it attempts to connect. Server port numbers usually are specified by the server application.

6 Supported TCP/IP Transport Providers

This section describes the TCP/IP transport providers supported:

- Under z/OS, IBM's eNetwork Communications Server is supported.
- Under z/VSE, Connectivity Systems TCP/IP stack and API or IBM TCP/IP for z/VSE (software provided by Connectivity Systems) only are supported.
- Under BS2000/OSD, Sockets, a component of Open Net Server, is supported.

IBM TCP/IP Communication Subsystem and API

In z/OS environments, OpenEdition sockets (API=OES) or the high performance native sockets interface (API=HPS) are used to communicate with the IBM TCP/IP address space. A valid OpenEdition security context, referred to as an OMVS segment, must be defined for Entire Net-Work. For more information, see the IBM document z/OS OpenEdition Planning Guide.

If the OMVS segment is not properly defined, TCP/IP driver initialization will fail with error 156. The Entire Net-Work error message received is NETP600I if API=HPS or NETP700I if API=OES.

Connectivity Systems TCP/IP Stack and API

In z/VSE environments, the native Sockets interface is used to communicate with the TCP/IP partition. To use the Connectivity Systems TCP/IP stack and API, users must ensure that the Adabas program ADARUN, which is used to invoke Entire Net-Work, has been linked with ADMODE (31). This is the distributed default upon release for both the Adabas library and the Adabas limited library.

Sockets Subsystem

In BS2000/OSD environments, the Sockets subsystem is used to communicate with BCAM:

- BS2000/OSD Versions 5 and above are supported.
- Sockets subsystem versions 2.0 and above are supported. This version number can be determined using the `SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=SOCKETS` command.

7 Connection Handling in Mixed IPv4/IPv6 Environments

This chapter describes the type of connection used when the TCPI or TCPX ALLOWIP6 driver parameter is set to "Y" and the TCP/IP stack is enabled for IPv6.

Outgoing Connections

The type of outgoing connection used depends on the settings of other parameters.

- If a value is specified for the V6IPADDR link parameter, the connection is attempted with IPv6.
- If a value is specified for the INETADDR link parameter, the connection is attempted with IPv4.
- If a value is specified for the ADJHOST link parameter, the connection attempt is based on whether the DNS name resolves to an IPv4 or and IPv6 address.
- If values are specified for both the V6IPADDR and ADJHOST parameters, the V6IPADDR setting takes precedence. Likewise, if values are specified for both the INETADDR and ADJHOST parameters, the INETADDR setting takes precedence. The V6IPADDR and INETADDR parameters are mutually exclusive; they cannot both be specified in the same link.

Note: If an IPv6 connection is attempted in a BS2000 environment, the target driver must have ALLOWIP6=Y to listen on the IPv6 address.

Incoming Connections

Links are searched for a match to the incoming connection request based on the following:

- If V6IPADDR is specified, the link is selected if its connecting address matches the V6IPADDR setting.
- If INETADDR is specified, the IPv4 address is typically delivered as an IPv4-mapped address. The link is selected if the IPv4 address in the mapped address matches the INETADDR setting.

  IPv4-mapped addresses are IPv4 addresses that are embedded within an IPv6-format address using the following form: 10 bytes of x'00', two bytes of x'FF', and the four-byte IPV4 address. For example:

  ```
  ::FFFF:129.144.52.38
  ```

  Note: This form has the IPv4 address in dotted-decimal notation. This convention can be used in the V6IPADDR parameter.

  For more information about IPv4-mapped addresses, refer to your IBM documentation.
- If ADJHOST is specified, the connecting IP address is resolved to a host name. The link is selected if the host name matches the ADJHOST setting.

- If no matching link is found and the ACCEPTUI parameter is set to "Y", the connection is treated as unsolicited (a new link is created). Otherwise, the connection attempt is rejected.

Note: If ALLOWIP6=Y is set in a BS2000 environment, it is listening on the IPv6 address. Connection using an IPv4 address cannot be made.

8 TCP/IP API Modules

The following table contains a list of available TCP/IP API modules. All support the Domain Names Services (DNS) GetHostByName and GetHostByAddr.

API	Interface Loaded	GetHostByName	GetHostByAddr
BS2	BS2000/OSD interface NWTCPBS2	Yes	Yes
CNS	z/VSE interface NWTCPCNS	Yes	Yes
EZA	z/VSE interface NWTCPEZA	Yes	Yes
HPS	IBM interface NWTCPHPS (High Performance Native Sockets)	Yes	Yes
OES	IBM OpenEdition sockets interface NWTCPOES	Yes	Yes

9 Problem Determination

If you have questions or difficulties concerning the installation or operation of this product, contact your Software AG technical support representative. Before doing so, however, Software AG recommends that you have the following information available:

- The type and release level of the operating system being used.
- A brief description of the system configuration; for example, the types of and number of partners in the network, the software being used by the partners.
- A brief description of the problem you are experiencing.
- A hard copy of the Entire Net-Work DDPRINT DD card output file and the Entire Net-Work console log showing all Entire Net-Work write-to-operators (WTOs).

If Software AG support personnel request tracing information, they will indicate the necessary parameters and provide you with the appropriate settings for these parameters.

10 Installing Entire Net-Work TCP/IP Option

Entire Net-Work TCP/IP Option includes two line drivers:

- The Entire Net-Work TCP/IP line driver (TCPI) allows a z/OS-compatible system to participate as a partner in a TCP/IP network with other Entire Net-Work nodes running on a variety of platforms, including OpenVMS, UNIX, Windows, and Macintosh, in addition to other mainframes.
- The Entire Net-Work Simple Connection Line Driver (TCPX) provides communication between ADI-enabled client applications and Adabas Version 8 databases running on z/OS. ADI-enabled clients are Software AG products that use Software AG's *Directory Server* (ADI) component. ADI facilitates the central management of directory services, eliminating the need for directory configuration files on every machine.

This chapter provides information for administrators responsible for installing Entire Net-Work TCP/IP Option.

Installing Entire Net-Work TCP/IP Option Under z/OS	Describes the installation of the Entire Net-Work TCP/IP Option on z/OS operating systems. Note that the Simple Connection Line Driver is not supported on MSP systems.

11 Contents of the Release Media

The following table describes most of the libraries included on the release (installation) media. Once you have unloaded the libraries from the media, you can change these names as required by your site, but the following lists the names that are delivered when you purchase Entire Net-Work TCP/IP Option.

Note: The complete list of libraries provided with Entire Net-Work can be found in the full Entire Net-Work documentation.

Library Name	Description
WTC*vrs*.LOAD	The z/OS load library for Entire Net-Work TCP/IP Option. The *vrs* in the library name represents the *version* of Entire Net-Work TCP/IP Option.

12 Installing Entire Net-Work TCP/IP Option Under z/OS

This section describes the installation steps for installing Entire Net-Work TCP/IP Option on a z/OS host systems.

Note: The Simple Connection Line Driver is not supported on MSP systems.

Installation Overview

To gain an understanding of the entire installation process, Software AG recommends that you read all of the installation steps before you perform the individual steps:

Step	Description
1	Verify the connectivity between cooperating nodes. This step is necessary only to support the TCP/IP line driver. It is not necessary for the Simple Connection Line Driver.
2	Unload the Entire Net-Work mainline and Entire Net-Work TCP/IP line driver libraries.
3	Outline your Entire Net-Work configuration and obtain all of the data necessary to configure the Entire Net-Work startup parameters.
4	Alter the Entire Net-Work Startup Job
5	Add the TCPI or TCPX-specific DRIVER and LINK statements.
6	Configure client information. This step is necessary only to support the Simple Connection Line Driver. It is not necessary for the TCP/IP line driver.
7	Ensure that ADARUN has been linked with attribute AMODE(31). This step is necessary only to support the TCP/IP line driver. It is not necessary for the Simple Connection Line Driver.
8	Start Entire Net-Work and perform installation verifications.

Installation Steps

Step 1. Verify the connectivity between cooperating nodes .

Note: This step is necessary only to support the TCP/IP line driver. It is not necessary for the Simple Connection Line Driver.

Before installing the Entire Net-Work TCP/IP line driver, verify the connectivity between cooperating nodes. Use the PING utility from the remote system to perform a loopback test with the local IBM TCP/IP node as follows:

```
PING node-name
```

where node-name is the name that identifies the IBM node.

Step 2. Unload the Entire Net-Work mainline and Entire Net-Work TCP/IP Line Driver libraries.

If not already performed, install the Entire Net-Work mainline libraries, using the procedure for your operating system environment (z/OS). For more information, see the section *Installation Overview* in the *Entire Net-Work Installation Guide*. Then unload the Entire Net-Work TCP/IP line driver components from the installation media as follows:

Under z/OS

Under z/OS, use IEBCOPY to restore the required data sets. Refer to the *Software AG Product Delivery Report* for the correct data set sequence numbers and names.

Step 3. Outline your Entire Net-Work configuration and obtain all data necessary to configure the Entire Net-Work startup parameters.

Note: If you are only interested in using the Simple Connection Line Driver, this step is not necessary.

Before you set up your network parameters, obtain the following information:

For the Local IBM Node

- The name of the node;
- The number of the Adabas SVC to be used;
- The subsystem name of the TCP/IP transport provider;
- The job name or started task name of the TCP/IP transport provider;
- The well known port number (referred to on the IBM mainframe as the SERVERID parameter of the TCP/IP Driver statement) that Entire Net-Work will use to listen for incoming connections.

For Each Remote Node

Note: This is not necessary or recommended if you plan to use the Simple Connection Line Driver.

- The name of each node;
- The node's Internet Protocol (IP) address or, if a Domain Name Resolver is being used, the Internet host name;
- The well known port number (referred to on the IBM mainframe as the SERVERID parameter of the DRIVER/LINK statement) that Entire Net-Work will use when establishing a connection to that node.

Note: On UNIX machines, refer to the /etc/services file for an entry "NETPTCP NNN/TCP". If this entry is not provided, the default well known port number is 7869.

Step 4. Alter the Entire Net-Work startup job.

Make the following changes to the Entire Net-Work startup JCL.

Under z/OS

A sample startup JCL member called JCLNET is provided in the source libraries. Then add the Entire Net-Work TCP/IP Option load library to the STEPLIB concatenation.

Step 5. Add the TCPI or TCPX-specific DRIVER and LINK statements.

Use the Entire Net-Work configuration built in Step 3, along with the parameters of the DRIVER and (optionally) LINK statements, to create the necessary DRIVER and LINK statements for your environment in the Entire Net-Work DDKARTE input file.

Note: If you are installing the Simple Connection Line Driver on z/VSE systems, the value of the TCPX DRIVER's API parameter must be "CNS".

Step 6. Configure client information.

Note: This step is necessary only to support the Simple Connection Line Driver. It is not necessary for the TCP/IP line driver.

In order for a client to correctly send the database request to the Entire Net-Work node where the database is located, Adabas Directory Server entries must be added for each database. These entries tell the client application where the server (Entire Net-Work) is located and which databases it serves. A Directory Server access entry must be added for each database that the client will call via the Simple Connection Line Driver.

A Directory Server access entry looks like this:

XTSaccess.*targetid***[0]=***protocol***://***host***:***port***[?***parm1***=***value***[&***parm2***=***value***]...]**

where *targetid* is the target database ID, *protocol* is the communication protocol to use, *host* is the name of the host computer where the server (Entire Net-Work with the Simple Connection Line Driver) runs, *port* is the port on which the server (Simple Connection Line Driver) will listen, *parm1* and *parm2* are optional parameter names and *value* represents the values of those parameters. The port number must match the Simple Connection Line Driver's SERVERID parameter. If one Simple Connection Line Driver will serve multiple databases, an Access entry for each database is required, but these entries would all specify the same port number.

For example:

```
XTSaccess.68[0]=TCPIP://ahost:3001
```

In this example, the target database is database 68. It should be accessed using the TCP/IP protocol on the "ahost" computer at port 3001.

For more information about the Directory Server, read *Adabas Directory Server Documentation* in the *Software AG Directory Server Installation and Administration Guide*.

Step 7. Ensure that ADARUN has been linked with AMODE(31).

Note: This step is necessary only to support the TCP/IP line driver. It is not necessary for the Simple Connection Line Driver.

The TCP/IP driver attempts to allocate all private virtual storage above the 16-megabyte line. To take advantage of this feature, ensure that ADARUN has been linked with attribute AMODE(31).

Step 8. Start Entire Net-Work and verify the installation.

Because of the many possible variations of the Entire Net-Work, Adabas, and applications topology, Software AG does not provide standard installation verification procedures. However, the following procedure is suggested for verifying the TCP/IP line driver installation:

1. Start the Entire Net-Work system and make connections to each link defined to the system.
2. Test the connections and verify that the links can be established from either side by connecting and disconnecting the links several times from each node. While the links are connected, issue the Entire Net-Work operator command DISPLAY TARGET to display the targets and the nodes on which they are located.
3. Test your applications running across Entire Net-Work. At first, run one application at a time, then verify the results.
4. For the final verification test, run a load test through the network (that is, multiple users on each node accessing data on the partner node).

The following procedure is suggested for verifying the Simple Connection Line Driver installation:

1. Start the Entire Net-Work system.
2. Test your applications running across Entire Net-Work. At first, run one application at a time, then verify the results.
3. For the final verification test, run a load test through the network (that is, multiple users on each node accessing data on the partner node).

13 TCP/IP Line Driver Overview

The Entire Net-Work TCP/IP line driver allows a z/OS-compatible system to participate as a partner in a TCP/IP network with other Entire Net-Work nodes running on a variety of platforms, including OpenVMS, UNIX, Windows and Macintosh, in addition to other mainframes.

This chapter provides information for administrators responsible for configuring and running the Entire Net-Work TCP/IP line driver once Entire Net-Work is installed.

The TCP/IP line driver documentation is organized as follows:

TCP/IP DRIVER Statement	Describes the syntax and parameters of the TCPI DRIVER statement.
TCP/IP LINK Statement	Describes the syntax and parameters of the TCPI LINK statement.
Operator Commands	Describes the operator commands you can use with the TCP/IP line driver.
Model Links	Describes the model link support offered by the TCP/IP line driver.
Statistics	Describes the statistics provided by the TCP/IP line driver you can use to help tune each link and the driver itself.
Configuration Example	Provides an example of a TCP/IP network with its related Entire Net-Work TCP/IP line driver statements and parameters.
User Exits	Describes the user exit support provided with the TCP/IP line driver.

14 TCP/IP DRIVER Statement

The TCP/IP DRIVER statement and its parameters are used to activate and define the characteristics of the local IBM mainframe node. The access method name TCPI instructs Entire Net-Work to load the line driver module NETTCPI.

In most cases, only one DRIVER statement needs to be coded in your Entire Net-Work startup parameters. However, multiple DRIVER statements can be defined to allow Entire Net-Work to listen on multiple ports.

DRIVER Statement Format

The TCP/IP DRIVER statement has the following format:

```
DRIVER  TCPI   [ACCEPTUI = { N | Y } ]
               [ ALLOWIP6 = { Y | N } ]
               API = { BS2 | CNS | EZA | HPS | OES }
               [ CONNQUE = { n | 10 } ]
               [ DRVCHAR = { character | # } ]
               [ DRVNAME = { driver-name | TCPI } ]
               [EXIT = name ]
               [ KEEPALIV = { N | Y } ]
               [MULTSESS = { Y | N } ]
               [ NODELAY = { N | Y } ]
               [ OPTIONS1 = ( n , n , n , n , n , n , n , n , n , n ) ]
               [ OPTIONS2 = ( x , x , x , x , x ) ]
               [PSTATS = { N | Y } ]
               [RESTART = ( interval , retries } ]
               [RSTATS = { Y | N } ]
               [ SERVERID = { n | 1995 } ]
               [STATINT = { stat-interval | 3600 } ]
               [ SUBSYS = { subsys-name | VMCF } ]
               [ TRACE = { Y | N } ]
               [ TRACELEV = ( N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y ) ]
               [ TRACESIZ = { size | 4096 } ]
               [ USERID = { { userid | TCPIP } } ]
                          { { nn | 00 } }
```

For more information about syntax conventions and rules used in this section, read *Conventions*.

Modifying the Driver Statement Parameters

The DRIVER statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. Some parameters can be modified when the line driver is open or closed. Others can be modified only when the line driver is closed. See ALTER and CLOSE in the section *TCP/IP Operator Commands*. The open/closed requirement for each parameter is included in the following descriptions.

DRIVER Statement Parameters

This section describes all of the parameters that can be used for the TCPI DRIVER statement.

For more information about syntax conventions and rules used in this section, read *Conventions*.

ACCEPTUI Parameter

```
ACCEPTUI = { Y | N }
```

This optional parameter determines whether the line driver will accept connections from systems that have not been previously defined with LINK statements. The ACCEPTUI parameter can be modified when the line driver is open or closed.

Valid values are "Y" (Yes) or "N" (No).

- If "Y" is specified, Entire Net-Work will accept connection requests from an undefined system and the required control blocks are built dynamically. Normal "handshaking" procedures with the new connections are performed. This is the default.

  User exits can be defined to provide security for incoming connections. For more information, see the section *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.
- If "N" is specified, Entire Net-Work will reject incoming requests from unknown source nodes.

ALLOWIP6 Parameter

```
ALLOWIP6 = { Y | N }
```

This optional parameter determines whether the line driver will accept connections using IPv6 communication. When the driver is opened, initialization for IPv6 communication is attempted. If the stack is not IPv6-enabled, IPv4 communication is used.

Valid values are "Y" (Yes) or "N" (No).

- If "Y" is specified, Entire Net-Work will attempt IPv6 communications when the driver is opened. ALLOWIP6=Y is only a valid specification if the API parameter has been set to one of the following values: BS2, HPS, OES, or EZA.

  Note: If ALLOWIP6=Y is set in a BS2000 environment, it is listening on the IPv6 address. Connection using an IPv4 address cannot be made.

- If "N" is specified, Entire Net-Work will not attempt IPv6 communications when the driver is opened.

API Parameter

```
API = { BS2 | CNS | EZA | HPS | OES }
```

This *required* parameter specifies the name of the TCP/IP application program interface being used. The API parameter can be modified only when the line driver is closed. Supported values are shown in the table below. There is no default.

Value	Description	Valid for Platforms
BS2	Loads the BS2000/OSD interface NWTCPBS2	BS2000
CNS	Loads the z/VSE interface NWTCPCNS	z/VSE
EZA	Loads the z/VSE interface NWTCPEZA. This option can be used only with the TCP/IP stack from Barnard Software, Inc.	z/VSE
HPS	Loads the IBM interface NWTCPHPS (High Performance Native Sockets)	z/OS and OS/390
OES	Loads the IBM OpenEdition sockets interface NWTCPOES	z/OS and OS/390

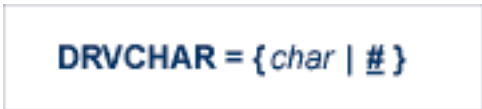
CONNQUE Parameter

```
CONNQUE = {n | 10}
```

This optional parameter specifies the number of connect queue entries. The value specified must accommodate the maximum number of simultaneous connection requests from remote nodes.

After the connection is accepted or rejected, connect queue entries are reused. If the value of this parameter is not high enough, the API routine is not able to process the incoming connection and the partner eventually will time out. Depending on the API being used, a message may be displayed indicating that an error has occurred. Values can range from 1 to 64; a value greater than 64 is reset to 64. The default value is 10. The CONNQUE parameter can be modified only when the line driver is closed.

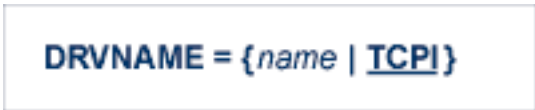
DRVCHAR Parameter



This optional parameter specifies the special character used to designate that an operator command should be directed to this line driver rather than to a specific link. The DRVCHAR parameter can be modified only when the line driver is closed.

The default is "#".

DRVNAME Parameter

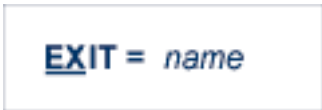


This optional parameter specifies the 4-byte driver name. The DRVNAME parameter can be modified only when the line driver is closed.

The default is "TCPI".

The DRVNAME parameter enables sites to make multiple TCP/IP API routines available at the same time. For example, the IBM APIs can be made available within the same Entire Net-Work address space. This parameter also allows two or more drivers to be defined so that Entire Net-Work can listen on multiple ports simultaneously.

EXIT Parameter



This optional parameter specifies the name of a user exit. The default is to run no user exit. For more information, read *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.

If EXIT is coded but the load module cannot be loaded, execution continues as if no exit were specified. The EXIT parameter can be modified only when the line driver is closed.

KEEPALIV Parameter

This optional parameter allows you to maintain connections when there is no other traffic with the remote links. Valid values are "Y" or "N."

- When this value is set to "Y", it causes internal TCP messages to be sent periodically to all remote links, thus maintaining the connections when there is no other traffic with the remote links. The amount of time between messages is determined by an initialization parameter in the TCP stack.
- When this value is set to "N", internal TCP messages are no longer sent periodically and the connections are not maintained.

The default for this parameter is Y.

KEEPALIV can also be set for individual remote links. For more information, read about the KEEPALIV parameter associated with the **TCP/IP LINK statement**.

MULTSESS Parameter

This optional parameter determines whether a connect request from a host that has an active connection is treated as a new link. This parameter can be modified when the line driver is open or closed.

A value of "Y" indicates that the connect request is treated as a new link; a value of "N" indicates that the connect request is treated as a reconnection of an existing link.

The default for this parameter is "N".

NODELAY Parameter

This optional parameter allows you to indicate whether the IBM socket option TCP_NODELAY is enabled or disabled for a link. TCP_NODELAY indicates whether data sent over the socket is subject to the Nagle algorithm (RFC 896). For more information, refer to your IBM documentation.

Valid values for this parameter are "Y" (the TCP_NODELAY option is enabled) or "N" (the TCP_NODELAY option is disabled). The default is "Y". When the NODELAY parameter on the DRIVER statement is specified, you do not need to specify the NODELAY parameter on the LINK statement. The value from the DRIVER statement is used. If the NODELAY parameter is not specified on either the DRIVER or LINK statements, a value of "Y" is assumed.

Note: The setting of this parameter is only effective if the API parameter is also set to "OES" or "HPS."

OPTIONS1 Parameter

```
OPTIONS1 = (n,n,n,n,n,n,n,n,n,n)
```

This optional parameter allows up to ten numeric API-specific options to be set. The values can be modified when the line driver is open or closed. There are no default values.

Not all APIs use the OPTIONS1 parameter.

The BS2000/OSD (BS2) API uses only two of the OPTIONS1 fields (prior to Entire Net-Work version 5.8, nine fields were used):

- The first, second, third, fourth, and fifth values are no longer used and must be set to 0.
- The sixth value is the Sockets task tracing level:
  - A value of 0 inhibits any tracing.
  - Values 1 and 2 give the corresponding level of high-level logic flow.
  - Values 3 through 9 log the transferred data and invoke the FSC sockets tracing.

  Tracing should be used only under the direction of Software AG.
- The seventh value is the maximum number of connections between the BS2 API and the Entire Net-Work partners. It is used for storage allocation by the Sockets task. The valid range is 2-2048 and the default value is 2048.
- The eighth and ninth value are no longer used and must be set to 0 or omitted.

OPTIONS2 Parameter

```
OPTIONS2 = (x,x,x,x,x)
```

This optional parameter allows up to five alphanumeric API-specific options to be set. The values can be modified when the line driver is open or closed. There are no default values.

Not all APIs use the OPTIONS2 parameter.

Beginning with Entire Net-Work version 5.8, the BS2000/OSD API (BS2) does not use the OPTIONS2 parameter at all.

PSTATS Parameter

```
PSTATS = { N | Y }
```

This optional parameter determines whether or not statistics are printed.

A value of "Y" indicates that statistics should be printed to DDPRINT when the statistics interval expires; a value of "N" indicates that the statistics should not be printed.

This parameter does not affect the STATS command and can be modified when the driver is open or closed.

The default for this parameter is "Y".

RESTART Parameter

```
RESTART = (interval,retries)
```

This optional parameter specifies the retry interval in seconds (*interval*) and the number of retries (*retries*) that Entire Net-Work will attempt to reopen the access method with the API after a shutdown due to a failure. The RESTART parameter can be modified when the line driver is open or closed.

If RESTART is not specified, or *interval* is specified as zero, no retry is attempted. By specifying (*retries*) as zero, an infinite number of retries can be requested.

If RESTART is specified on the TCP/IP DRIVER statement, a corresponding RESTART parameter value on each related LINK statement should be specified to control restart attempts on the individual link.

The RESTART parameter is particularly useful with the TCP/IP line driver when Entire Net-Work is started at IPL and communication with the API or VTAM ACB is unsuccessful because TCP/IP is not yet fully initialized. Using this parameter, you can instruct Entire Net-Work to reopen the TCP/IP session, thereby giving TCP/IP sufficient time to become active.

The TIMER parameter on the NODE statement affects the RESTART parameter (see the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*.) The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead.

RSTATS Parameter

```
RSTATS = { Y | N }
```

This optional parameter determines whether or not statistics are reset. A value of "Y" indicates that statistics should be reset when the statistics interval expires; a value of "N" indicates that the statistics should not be reset. The default is "N".

The RSTATS parameter can be modified when the line driver is open or closed.

The statistics interval value is set by link parameter `STATINT`. If LINK parameter `STATINT` is not define, the interval is set by the DRIVER parameter `STATINT`.

SERVERID Parameter

```
SERVERID = { n | 1995 }
```

This optional parameter specifies a well known port number used by Entire Net-Work while awaiting connection requests from participating Entire Net-Work partners. Values may range from 1 to 65535. The SERVERID parameter can be modified only when the line driver is closed.

When specified in a DRIVER statement, the SERVERID parameter specifies the port number of the Entire Net-Work being initialized. If SERVERID is not specified for a link, the SERVERID specified for the driver is used as the default port for the link.

- On UNIX platforms, the SERVERID parameter used by Entire Net-Work for UNIX can be found in the /etc/services file. The following are examples of the information in this file:

SAGBET21	1405/TCP	(VMS)
NETPTCP	1405/TCP	(UNIX)

  If no entry exists in the /etc/services file, Entire Net-Work for UNIX uses a default of "7869" for the SERVERID parameter.
- Under OpenVMS, this information may be located in different places, depending on the API being used. Refer to the *Entire Net-Work for OpenVMS Reference Guide* for more information.

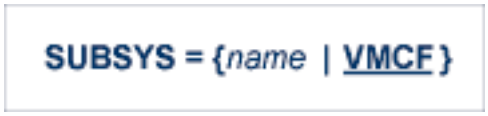
The default for this parameter is 1995. Both the DRIVER statement and the LINK statement have a SERVERID parameter.

STATINT Parameter

This optional parameter specifies the amount of time, in seconds, before statistics are automatically printed or reset. The default is 3600. The STATINT parameter can be modified when the line driver is open or closed.

Acceptable values range from 1 to 2147483647. Any value outside this range is in error.

SUBSYS Parameter



This parameter specifies the name of the subsystem to be accessed by the API routines that use subsystem control blocks in interaddress space communications. The default value is VMCF. The SUBSYS parameter can be modified only when the line driver is closed.

In a z/OS environment, the IBM API routines communicate to the system address space by locating the subsystem control table and retrieving the information required to perform cross-memory communication. If the subsystem is specified incorrectly, the driver is not able to perform its open processing and no connections are possible.

This parameter is not used in a z/VSE or BS2000 environment.

TRACE Parameter

```
TRACE = { Y | N }
```

This parameter indicates whether tracing for this line driver should be active (Y) or not (N). When tracing is activated, trace information is placed in the trace table. The default is N (no). The TRACE parameter can be modified when the line driver is open or closed.

This is equivalent to specifying `TRACE=linedriver-code` or `TRON=linedriver-code` in the NODE statement (for example, `TRACE=CTCA`).

TRACELEV Parameter

```
TRACELEV = ( Y | N, Y | N, Y | N, Y | N, Y | N, Y | N, Y | N, Y | N, Y | N, Y | N )
```

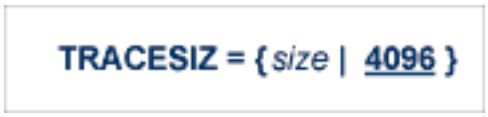
This optional parameter specifies the levels of tracing that the line driver will perform. It is a series of flags that determine which events are traced. The TRACELEV specification must be enclosed in parentheses. For example:

```
TRACELEV=(N,N,N,N,N,N,N,N,N,N)
```

Trace levels are positional within the parameter syntax and are set using Y (Yes) or N (No). It is recommended that all settings within the TRACELEV parameter be N. If your system experiences problems, contact your Software AG technical support representative for the settings that produce the appropriate trace information. The TRACELEV parameter can be modified when the line driver is open or closed.

Note: The tracing information provided is sent to the DDPRINT data set. In addition to setting the TRACELEV flags, the trace must also be turned on using either the DRIVER statement parameter `TRACE=Y` or the operator command `TRACE=linedriver-name`. Tracing dramatically affects the overall performance and throughput of Entire Net-Work.

TRACESIZ Parameter



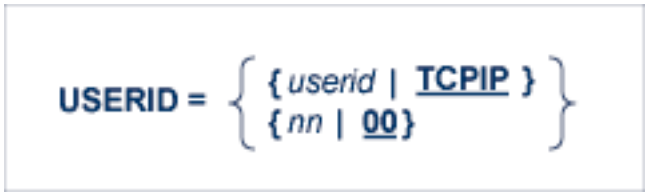
This optional parameter specifies the size, in bytes, of the driver-specific trace table.

This parameter is also used as the default size of the link specific trace table when the LINK statement does not include a TRACESIZ specification.

Valid values can range from 4096 to 4194304. A value less than 4096 is reset to 4096; a value greater than 4194304 is reset to 4194304. The default is "4096".

The TRACESIZ parameter can be modified only when the line driver is closed.

USERID Parameter



This parameter applies only to the IBM and CNS (z/VSE) APIs. Its value can be modified only when the line driver is closed.

- For IBM APIs (i.e., the NWTCPIBM interface, the NWTCPHPS interface, and the NWTCPOES interface), the USERID parameter specifies the name of the started task, job, or virtual machine in which the IBM TCP/IP protocol stack is running. The value is 1-8 characters. The default value is TCPIP.
- For the CNS API (i.e., the NWTCPCNS interface), the USERID parameter is used to direct traffic to a particular TCP/IP stack. The value is a two-digit number that must match the ID= value in the PARM field of the TCP/IP stack. The default value is 00, which is also the default for the TCP/IP stack. The value of USERID is not validated; if its value does not match the ID= value of an active TCP/IP stack, the TCP/IP driver will fail to open and it will return messages similar to the following:

```
NETP571W TCP API ERROR ON OPEN - RC = 0008
NET0101I  TCPI DRIVER OPEN FAILED - RC = 0004
```

To use multiple TCP/IP stacks, one DRIVER statement must be provided for each stack, and each link must specify the driver associated with the stack it will use. When defining multiple drivers, copy the module NETTCPI.phase and change the last four characters of the name to match the DRIVER name. For example, to define the following, NETTCPI would be copied (not renamed) to NETTCP2:

```
*Links using Driver TCPI use TCP/IP stack with ID=00
DRIVER TCPI  API=CNS
LINK PC01 TCPI,INETADDR=(x,x,x,x)
```

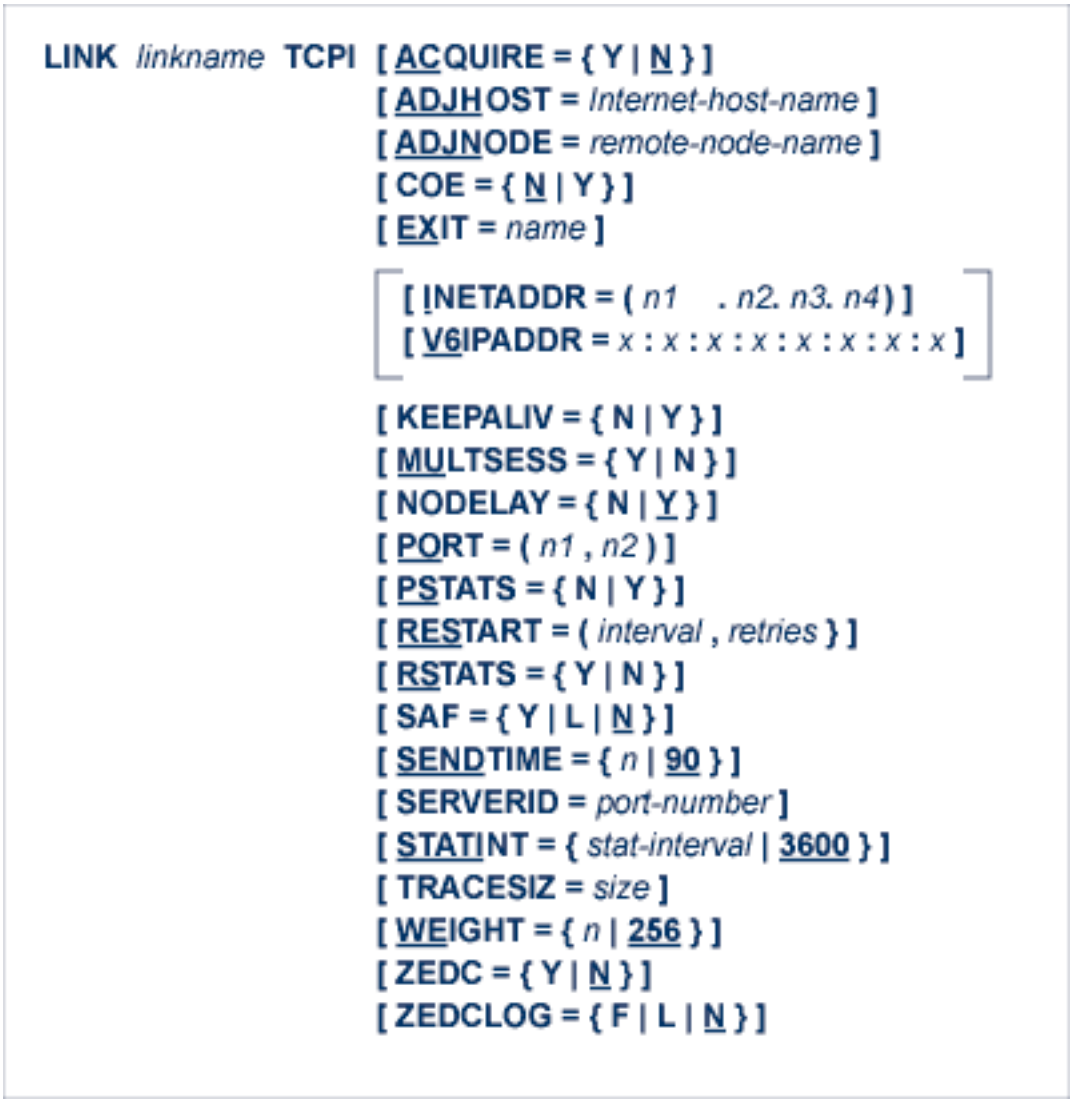
```
*Links using Driver TCP2 use TCP/IP stack with ID=02
DRIVER TCP2 API=CNS,USERID=02,DRVNAME=TCP2
LINK PC02 TCP2,INETADDR=(x,x,x,x)
```

15 TCP/IP LINK Statement

The LINK statement and its parameters are used to define the characteristics of the remote node. Normally, there is more than one LINK statement defined in the network startup parameters, that is, one link per node.

LINK Statement Format

The TCP/IP LINK statement has the following format:



For more information about syntax conventions and rules used in this section, read *Conventions*.

Modifying the LINK Statement Parameters

The LINK statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. Some parameters can be modified when the link is open or closed. Others can be modified only when the link is closed. See ALTER and CLOSE in the section *TCP/IP Operator Commands*. The open/closed requirement for each parameter is included in the following descriptions.

LINK Statement Parameters

This section describes all of the parameters that can be used for the TCPI LINK statement.

For more information about syntax conventions and rules used in this section, read *Conventions*.

linkname Parameter

```
linkname
```

The required *linkname* parameter specifies the name by which this link is to be known. It is positional and must be specified immediately after the LINK keyword and immediately before the driver name (TCPI). The link name must be unique on the node. All operator commands affecting the link must specify this name.

If the link name begins with the characters "MODEL", the link is defined as a model link. See the section ***Model Links***, elsewhere in this section.

TCPI Parameter

```
TCPI
```

TCPI is required and specifies the protocol name that defines the driver associated with this link. It must be the same as the value specified for the **DRVNAME parameter** in the **TCPI DRIVER statement**. In most cases, this value is "TCPI". If DRVNAME is changed to a value other than "TCPI", this parameter must also be changed.

ACQUIRE Parameter

```
ACQUIRE={N | Y}
```

This optional parameter specifies whether or not a connection with the remote node should be attempted when the driver is opened for the first time (during system initialization). If N is specified, the link is connected manually using operator commands from the client or server node. The ACQUIRE parameter can be modified only when the link is closed. The default value is N.

ADJHOST Parameter

```
ADJHOST=Internet-host-name
```

This optional parameter specifies the Internet host name of a node with which a connection is to be established. The name can be resolved to either an IPv4 or IPv6 address. Its value can be 1 - 255 characters. The ADJHOST parameter can be modified only when the link is closed.

The ADJHOST parameter uses Domain Name Services (DNS), as follows:

- The GetHostByName function is used to determine the IP address of a host name specified with ADJHOST. IP address is used both for connecting to another node and for locating the link for an incoming connection.
- The GetHostByAddr function is used to determine the host name of a node that is trying to connect to this node. This is necessary when the IP address of a host name specified with ADJHOST changes after the link has been opened.

Software AG recommends the use of the ADJHOST parameter for sites that assign IP addresses via the DHCP protocol. Entire Net-Work will use the GetHostByName function for every outgoing connection on nodes that have ADJHOST specified as long as INETADDR is not specified.

The following table lists the APIs that support Domain Name Services:

API	GetHostByName	GetHostByAddr
BS2	Yes	Yes
CNS	Yes	No
EZA	Yes	Yes
HPS	Yes	Yes
OES	Yes	Yes

For performance reasons, Software AG recommends that all LINK statements containing an ADJHOST value be defined after the LINK statements containing an INETADDR or V6IPADDR specification. If none of these parameters is specified, the link is not usable. If both the INETADDR and ADJHOST parameters are specified, the INETADDR parameter takes precedence. Likewise, if both the V6IPADDR and ADJHOST parameters are specified, the V6IPADDR parameter takes precedence. The INETADDR and V6IPADDR parameters are mutually exclusive; only one of them can be specified in the same link.

ADJNODE Parameter

```
ADJNODE=remote-node-name
```

This optional parameter specifies a node name to be assigned to this link. It is required only in cases where there is more than one link to the same IP address on systems that do not pass their node name. Such systems include older versions of UNIX and Entire Net-Work for the workstation. The name specified can be 1 to 8 alphanumeric characters and must be unique. Special characters should not be used; some systems do not accept node names with special characters. The ADJNODE parameter can be modified only when the link is closed.

COE Parameter

```
COE={N | Y}
```

This parameter can be used to force a node ID of 0 (zero) for all unsolicited connections via this link. It is similar to the NIDO parameter on the NODE statement, but allows you to set the COE value at the link level. The COE link parameter applies to the TCP/IP driver only.

A node ID of 0 means that the node is to be treated as a Client Only Element. A Client Only Element is a node that has no databases active and has no other Entire Net-Work nodes connected to it. Only non-mainframe nodes can be Client Only Elements; a COE setting for a mainframe link is ignored.

If COE=N (the default value), Entire Net-Work will respect the connecting node's client mode setting and will determine the COE status for the link based on that setting.

If COE=Y, Entire Net-Work will force the node ID to zero and set the link's COE flag to on.

The connection to a COE node is not broadcast to other nodes; only this node will include it in a response to a DISPLAY TARGETS command. When displayed by a DISPLAY TARGETS command, it is denoted as a communicator node, but with a lower-case 'c' (see message NET0124 in the section *Entire Net-Work Control Module Messages* of the *Entire Net-Work Messages and Codes Manual*).

When specified on a model link within this driver, any unsolicited connections handled by this driver will be treated as COE nodes.

EXIT Parameter

```
EXIT={name | none}
```

This parameter specifies the name of a user exit. For more information, read *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.

If a value for EXIT is specified but the load module cannot be loaded, execution continues as if no exit were specified. The EXIT parameter can be modified only when the link is closed.

INETADDR Parameter

```
INETADDR=(n1.n2.n3.n4)
```

This optional parameter specifies the IPv4 address of the remote host associated with this link. The INETADDR parameter can be modified only when the link is closed.

IP address is used both for connecting to another node and for locating the link for an incoming connection. It is provided to Entire Net-Work in the form of INETADDR=(n1,n2,n3,n4) or INET-ADDR=(n1.n2.n3.n4) where each value represents 8 bits of the 32-bit IP address. Acceptable values are between 0 and 255 and may be separated by commas or periods. For example:

```
INETADDR=(157,182,17,20) ↩
```

or

```
INETADDR=(157.182.17.20) ↩
```

On most UNIX-based machines, this address can be found in the /etc/hosts file. The following are examples of the information in the /etc/hosts file:

157.182.17.20 DALLAS dallas (VMS)

157.182.17.18 DENVER denver (UNIX)

Each host on the INTERNET is assigned a unique IP address which is used by the IP and higher level protocols to route packets through the network. The IP address is logically made up of two parts: the network number and the local address. This IP address is 32 bits in length and can take on different formats or classes. The class defines the length (number of bits) of each part. There are four classes (A, B, C, and D); the class is identified by the allocation of the initial bit.

For performance reasons, Software AG recommends that all LINK statements containing an INET-ADDR value be defined before the LINK statements containing an ADJHOST specification. If neither of these parameters or the V6IPADDR parameter are specified, the link is not usable. If both INETADDR and ADJHOST are specified, the INETADDR parameter takes precedence.

Note: Do not specify both V6IPADDR and INETADDR in the same link statement; they are mutually exclusive parameters.

KEEPALIV Parameter

```
KEEPALIV={Y | N}
```

KEEPALIV=Y (Yes) or T (True) causes internal TCP messages to be sent periodically to the remote node, thus maintaining the connection when there is no other traffic with the node. The amount of time between messages is determined by an initialization parameter in the TCP stack. If no KEEPALIV value is specified for the link, it defaults to the KEEPALIV value on the DRIVER statement.

MULTSESS Parameter

```
MULTSESS={N | Y}
```

This optional parameter determines whether a connect request from a host that has an active connection will be treated as a new link (Y), or a reconnection of an existing link (N). The default value is the value specified for the MULTSESS parameter in the DRIVER statement (N or Y, with N as the default). The MULTSESS parameter can be modified when the link is open or closed.

NODELAY Parameter

```
NODELAY = { N | Y }
```

This optional parameter allows you to indicate whether the IBM socket option TCP_NODELAY is enabled or disabled for a link. TCP_NODELAY indicates whether data sent over the socket is subject to the Nagle algorithm (RFC 896). For more information, refer to your IBM documentation.

Valid values for this parameter are "Y" (the TCP_NODELAY option is enabled) or "N" (the TCP_NODELAY option is disabled). The default is "Y". When the NODELAY parameter on the DRIVER statement is specified, you do not need to specify the NODELAY parameter on the LINK

statement. The value from the DRIVER statement is used. If the NODELAY parameter is not specified on either the DRIVER or LINK statements, a value of "Y" is assumed.

Note: The setting of this parameter is only effective if the API parameter is also set to "OES" or "HPS."

PORT Parameter

```
PORT=(n1,n2)
```

This optional parameter specifies the port numbers used for this link. It is required only in cases where there is more than one link to the same IP address. The local port number (n1) is the port number used to bind to on the local stack. In most cases, no value for n1 should be specified so the default value (the value specified by SERVERID=) will be used, i.e., PORT=(,n2). The remote port number (n2) specifies the port number of the remote partner. It is used to uniquely identify this link on an incoming connect request when more than one remote Entire Net-Work node has the same IP address. The value of n2 must be set to the port number of the remote Entire Net-Work partner that this link represents. The PORT parameter can be modified only when the link is closed.

PSTATS Parameter

```
PSTATS={N | Y}
```

This optional parameter determines whether or not (Y or N) statistics are printed to DDPRINT when the statistics interval expires. The default value is PSTATS=Y. This parameter does not affect the STATS operator command. The PSTATS parameter can be modified when the link is open or closed.

RESTART Parameter

```
RESTART= (i,n)
```

This optional parameter specifies the retry interval in seconds (*i*) and the number of retries (*n*) that are made to start the connection to the remote node. If RESTART is not specified, or (*i*) is specified as zero, no retry is attempted. By specifying (*n*) as zero, an infinite number of retries can be requested.

The TIMER parameter on the NODE statement affects the RESTART parameter (see the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*. The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead. The RESTART parameter can be modified when the link is open or closed.

RSTATS Parameter

```
RSTATS={N | Y}
```

This optional parameter determines whether or not (Y or N) statistics are automatically reset when the statistics interval expires. The default value is the value specified for the RSTATS parameter in the **DRIVER statement**. The RSTATS parameter can be modified when the link is open or closed.

SAF Parameter

```
SAF={Y | L | N}
```

If SAF=Y or SAF=L is specified, Entire Net-Work will call the SAF Interface for all incoming requests on this link; failure to load the Interface is considered a security violation and Entire Net-Work will shut down. If SAF=L, the calls are traced and the output directed to DDPRINT. An error code is transmitted to the user if access to SAF is denied. The SAF parameter can be modified when the link is open or closed. The default value is N (No).

SENDTIME Parameter

```
SENDTIME={n | 90}
```

This optional parameter specifies the time (in seconds) that the TCP/IP line driver allows for a send to complete. When this time is exceeded, the line driver writes a message to the operator console indicating a possible error condition on the remote node. The connection is considered severed and link disconnect processing is initiated. The default value is 90 seconds. The SENDTIME parameter can be modified when the link is open or closed.

SERVERID Parameter

```
SERVERID=port-number
```

This parameter specifies a well known port number associated with a remote partner. Entire Net-Work will attempt to use this port number when connecting to the remote partner. If this parameter is not specified, or is set to zero (0), the value specified for the SERVERID parameter on the **DRIVER statement** is used. The SERVERID parameter can be modified only when the link is closed.

STATINT Parameter

```
STATINT={statistics interval | 3600}
```

This optional parameter specifies the amount of time, in seconds, before statistics are automatically printed or reset. Acceptable values are 1 - 2147483647. Any value outside this range is in error. The default value is 3600. The STATINT parameter can be modified when the link is open or closed.

TRACESIZ Parameter

```
TRACESIZ=trace table size
```

This optional parameter specifies the size of the TCP/IP link specific trace table. Value can be 4096 - 4194304. A value less than 4096 is reset to 4096. A value greater than 4194304 is reset to 4194304. The default value is the value specified for the TRACESIZ parameter in the **DRIVER statement**. The TRACESIZ parameter can be modified only when the link is closed.

V6IPADDR Parameter

```
V6IPADDR=x:x:x:x:x:x:x:x
```

This optional parm specifies the IPv6 address of the remote host associated with this link. The V6IPADDR parameter can be modified only when the link is closed. Specify the address as *x:x:x:x:x:x:x:x*, where each *x* represents a hexadecimal value of the eight 16-bit pieces of the address.

Standard abbreviated forms are allowed:

- Leading zeros in an individual value may be omitted;
- Groups of zeros may be eliminated, specifying them as a double colon '::'. A double colon may be used only once in an address.
- IPv4-compatible and IPv4-mapped IPv6 addresses can be specified.

The following are examples of valid IPv6 addresses:

```
2001:DB8:7654:3210:FEDC:BA98:7654:3210
2001:DB8:0:0:8:800:200C:417A
2001:DB8::8:800:200C:417A
::13.1.68.3
::FFFF:129.144.52.38
```

For performance reasons, Software AG recommends that all LINK statements containing a V6IPADDR value be defined before the LINK statements containing an ADJHOST specification. If neither of these parameters or the INETADDR parameter are specified, the link is not usable. If both V6IPADDR and ADJHOST are specified, the V6IPADDR parameter takes precedence.

Note: Do not specify both V6IPADDR and INETADDR in the same link statement; they are mutually exclusive parameters.

WEIGHT Parameter

```
WEIGHT={n | 256}
```

This parameter specifies the weight of this link with respect to other links going to the same node. If a given target can be reached by more than one path (chain of connected links), the path with the lowest weight is used. Slow or expensive links should be given a higher value than fast or inexpensive links. Values range from 1 to 999999. The default value is 256. The WEIGHT parameter can be modified only when the link is closed.

ZEDC Parameter

```
ZEDC={ Y | N }
```

This parameter indicates whether zEnterprise Data Compression (zEDC) compression can occur for the link. Valid values are "Y" or "N"; "N" is the default. Determination of whether or not zEDC data compression occurs is based on a combination of the settings of this parameter and the ZEDCINIT parameter on the NODE statement, as described in the following table:

LINK ZEDC Parameter Setting	NODE ZEDCINIT Parameter Setting	Result
Y	Y	Outbound buffers for the link are compressed.
Y	N	Outbound buffers are not compressed.
N	Y	Outbound buffers for TCPI links are not compressed, but other outbound buffers might be (depending on the setting of their LINK statement ZEDC parameters).
N	N	Outbound buffers are not compressed.

Note: If the node-to-node handshake indicates that the destination node does not support zEDC data compression, the outbound payload will not be compressed, regardless of any zEDC parameter settings on the NODE statement or any LINK statement.

zEnterprise Data Compression (zEDC) can occur only on z/OS operating systems. Consequently, ZEDC=Y can be specified only on z/OS systems that support zEDC. For complete information on z/OS requirements for zEDC support, refer to IBM documentation regarding *zEnterprise Data Compression (zEDC)*.

When compression occurs it occurs on buffers with sizes greater than the value defined by the NODE statement's ZEDCSZ parameter.

ZEDCLOG Parameter

```
ZEDCLOG={ F | L | N }
```

This optional parameter indicates what level of trace data will be logged for zEDC compression processing. This trace data logging occurs independently of Entire Net-Work's global tracing parameter setting (LOG=YES or LOG=FULL parameter settings on the NODE statement). Valid values are described in the following table:

ZEDCLOG Setting	Result
F	Trace data is logged prior to and after compression and decompression processing. The amount of data logged is equivalent to the length of the data.
L	Trace data is logged prior to and after compression and decompression processing. The amount of data logged is 100 (x'64') bytes.
N	This is the default. No trace data is logged.

Note: The F and L settings of ZEDCLOG should be used sparingly; these settings greatly increase the DDPRINT output size.

The ZEDCLOG parameter, can be modified when a link is open or closed.

Note: If the node-to-node handshake indicates that the destination node does not support zEDC data compression, the outbound payload will not be compressed, regardless of any zEDC parameter settings on the NODE statement or any LINK statement.

zEnterprise Data Compression (zEDC) can occur only on z/OS operating systems. Consequently, the ZEDCSLOG parameter specification should be made only on z/OS systems that support zEDC. For complete information on z/OS requirements for zEDC support, refer to IBM documentation regarding *zEnterprise Data Compression (zEDC)*.

16 TCP/IP Line Driver Operator Commands

Entire Net-Work's TCP/IP line driver has the ability to process operator commands that are directed to a specific link or directly to the driver.

Operator Command Syntax

Under z/OS, the TCP/IP line driver operator commands have the following format:

```
F NETWORK, TCPI target cmd
```

The following table describes this syntax.

Syntax Representation	Description
TCPI	Informs Entire Net-Work that the command is destined for the TCP/IP driver. If more than one TCPI DRIVER statement exists, use the name specified on the DRVNAME parameter of the **DRIVER statement** instead of TCPI.
target	A value that informs TCPI what the target of the command is, as follows: ■ Specify an asterisk (*) if the target is all links. ■ Specify the DRVCHAR value ("#" is the default) if the target is the driver itself (see the DRVCHAR parameter on the ***TCP/IP DRIVER Statement***). ■ Specify the link name if the target is a specific link.
cmd	The operator commands to be carried out. Multiple commands can be specified in a single command statement. When the `ALTER` command is specified, it must be the last command in the statement, because everything following the `ALTER` command is treated as a DRIVER or LINK statement parameter. One or more DRIVER or LINK statement parameters must be specified.

Examples

The following are examples of TCP/IP line driver operator commands:

```
F NETWORK,TCPI * CLOSE
```

```
NETWORK TCPI # STATS
```

```
F NETWORK,TCPI link3 CONNECT
```

Driver Commands

The Entire Net-Work TCP/IP line driver supports the commands listed in the following table when the target is the driver. The underlined portion of the command is the minimum abbreviation.

Command	Action

ALTER *driver-parms*	Dynamically changes the driver configuration. The ALTER command is followed by the driver configuration parameters to be altered. The driver configuration parameters are the same as those specified in the DRIVER statement. For example:
	`TCPI # ALTER`
	`ACCEPTUI=Y`
	Refer to the specific parameter description for information on possible restrictions about modifying the parameter using the ALTER command.

CLOSE	Disconnects and closes all links that are connected to other nodes. Releases all resources held by the driver as well as all open links. Closes the driver.
OPEN	Reopens the driver after it is closed with the CLOSE operator command or because of an access method failure. Allocates all the resources needed by the driver to communicate with TCP/IP. Also attempts to resolve any unresolved host names.
RESET	Resets all statistics for the driver. Statistics are printed only if the STATS command precedes the RESET command.
SHOW	Displays the current configuration of the driver. The current configuration is always shown automatically following an ALTER command.
SNAP	Causes all control blocks specific to the link to be snapped (printed in hexadecimal). Driver specific control blocks and Entire Net-Work specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. This command has no effect on the next automatic printing of statistics. To print and reset statistics, specify RESET immediately after the STATS command. For example: `TCPI # STATS RESET`
STATUS	Displays the current status of the driver as well as a count of messages received and sent.
TRACE	Causes the TCP/IP driver to format and print the driver-specific trace table. The trace table is formatted and printed in hexadecimal automatically when the SNAP command is processed.

Note: When the driver is closed, it does not recognize the commands CLOSE, STATS, or RESET.

Link Commands

The Entire Net-Work TCP/IP line driver supports the commands listed in the following table when the target is a link or all links. The underlined portion of the command is the minimum abbreviation.

Command	Action
ALTER *link-parms*	Dynamically changes the link configuration. The ALTER command is followed by the link configuration parameters to be altered. The link configuration parameters are the same as those specified on the LINK statement. For example: `TCPI LINK1 ALTER ADJHOST=DALLAS` Refer to the specific parameter description for information on possible restrictions about modifying the parameter using the ALTER command.
CLOSE	Disconnects the link if it is connected to another node and releases all resources held by the link.
CONNECT ↩	Attempts to establish one or more TCP/IP sessions with the target link(s). If the link is already connected or is in the process of connecting, the command is ignored.
DISCONNECT ↩	Starts the disconnect sequence for the target link(s). If the link is already disconnected or is in the process of disconnecting, the command is ignored.
LOGLON *linkname*	Turns on selective logging for the specified link.
LOGLOFF *linkname*	Turns off selective logging for the specified link.
OPEN	Allocates all the resources needed by the link to communicate with TCP/IP. Does not initiate a connect to the remote node. The status of the link displayed via the SHOW operator command is not affected by the OPEN request.
RESET	Resets all statistics for the link. Statistics are printed only if the STATS command precedes the RESET command.
RESUME	Restarts processing on a link that was temporarily stopped due to a SUSPEND command.
SHOW	Displays the current configuration of the link. The current configuration is always shown automatically following an ALTER command.
SNAP	Causes all link specific control blocks and the link specific trace table to be snapped (printed in hexadecimal). Driver specific control blocks and Entire Net-Work specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. This command has no effect on the next automatic printing of statistics. To print and reset statistics, specify RESET immediately after the STATS command. For example: `TCPI LINK1 STATS RESET`
STATUS	Displays the current status of the link as well as a count of messages received and sent.

Command	Action
SUSPEND	Temporarily stops all processing on a link. Processing can be restarted with the RESUME command.
TRACE	Causes the link specific trace table to be formatted and printed. The trace table is formatted and printed in hexadecimal automatically when the SNAP command is processed.

BS2000/OSD Net-Work Sockets Tracing

The BS2000 sockets task includes code to trace the logic, the data transferred, and the interface with the FSC Sockets subsystem. This tracing is normally requested using the sixth value of the OPTIONS1 parameter. In order to facilitate the diagnosis of problems, there are two operator commands that can be directed to the Sockets task. The first allows the level of tracing to be altered dynamically. The changes the SYSOUT file used by the task, allowing the tracing output to be viewed without shutting down Entire Net-Work. The operator commands can be supplied from an operator console or from a DIALOG session in a USER-ID with operating privileges ($TSOS, for example).

Command	Action
CHANGE-TRACE *n*	The tracing level is set to *n*, where *n* is the tracing level. On z/OS and z/VSE systems, the tracing levels can range from 0-9. On BS2000 systems, the tracing levels can range from 0-19 and behave as described below: ■ A value of "0" disables tracing immediately. ■ A value of "1" traces connect/accept events plus error situations. ■ A value of "2" or "3" traces payload messages as well. ■ Values "4" through "9" also include a tracing of the socket library of BS2000. ■ A value of "10" is ignored. ■ A value of "11" behaves in the same manner as a value of "1", but goes into level "5" automatically when outgoing data transfer is stopped due to system congestion. When the congestion is cleared, level "5" is switched back to level "1" automatically. ■ Values ranging from "12" through *19* are ignored. On BS2000 systems, all values greater than "1" can have a negative impact on your systems performance and should only be activated in problem situations.
CHANGE-SYSOUT	The current SYSOUT file is replaced. If an FGG is in use, the generation will be used; if a SAM file is in use, a new file will be created with a suffix (its name will appear at the end of the original SYSOUT file).

Examples

To switch on tracing:

```
/INF-PROG 'CHANGE-TRACE 2',*TSN(<Sockets task TSN>)
```

To switch off tracing:

```
/INF-PROG 'CHANGE-TRACE 0',*TSN(<Sockets task TSN>)
```

To free the SYSOUT file so the tracing output can be viewed:

```
/INF-PROG 'CHANGE-SYSOUT',*TSN(<Sockets task TSN>)
```

17 Model Links

The TCP/IP line driver supports dynamically added links, thus reducing the time required to set up and maintain the TCP/IP LINK statements. Model links can be coded and used to define new links as they are added.

The TCP/IP line driver is permitted to add links dynamically if ACCEPTUI=Y is coded on the DRIVER statement. A new link block is created and is used to control all further communications on the link. The link block can be initialized with default values that will be applied to each new link. Alternatively, one or more model links can be defined to override the values contained in the link block A user exit can be written to select model link parameters when accepting a connection from a host that is not predefined to Entire Net-Work. For more information, see the section *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.

The model link statement is identical to other LINK statements, except that the link name begins with the characters 'MODEL'. Most of the model link parameters, such as PSTATS and RSTATS, are copied into the dynamically built link block. Some parameters, such as INETADDR, are not copied because they are truly link-specific.

18 Statistics

The Entire Net-Work TCP/IP line driver issues API calls to communicate with TCP. To help tune each link and the driver itself, the TCP/IP driver provides the statistics shown below. The statistics for a link and the driver are identical, with the exception of the title line (a).

```
    +-------------------------------------------------------------------------+
(a) + Statistics For Driver   TCPI  Period     0:01:02 (     62.232 Secs)  +
    + -------------  ---Bytes----  --Messages--  -Api Calls--  ----------- +
(b) + Writes              0.000K             0             0  Total       +
(c) +                     0.000K             0             0  Per Second  +
(d) + Reads               0.000K             0             0  Total       +
(e) +                     0.000K             0             0  Per Second  +
    + -------------  ---Total----  ----Task----  ---Other----  ----------- +
(f) + Write Cmd's               0             0             0  Total       +
(g) +                           0             0             0  Per Second  +
(h) + Read Cmd's                0             0             0  Total       +
(i) +                           0             0             0  Per Second  +
    +-------------------------------------------------------------------------+
```

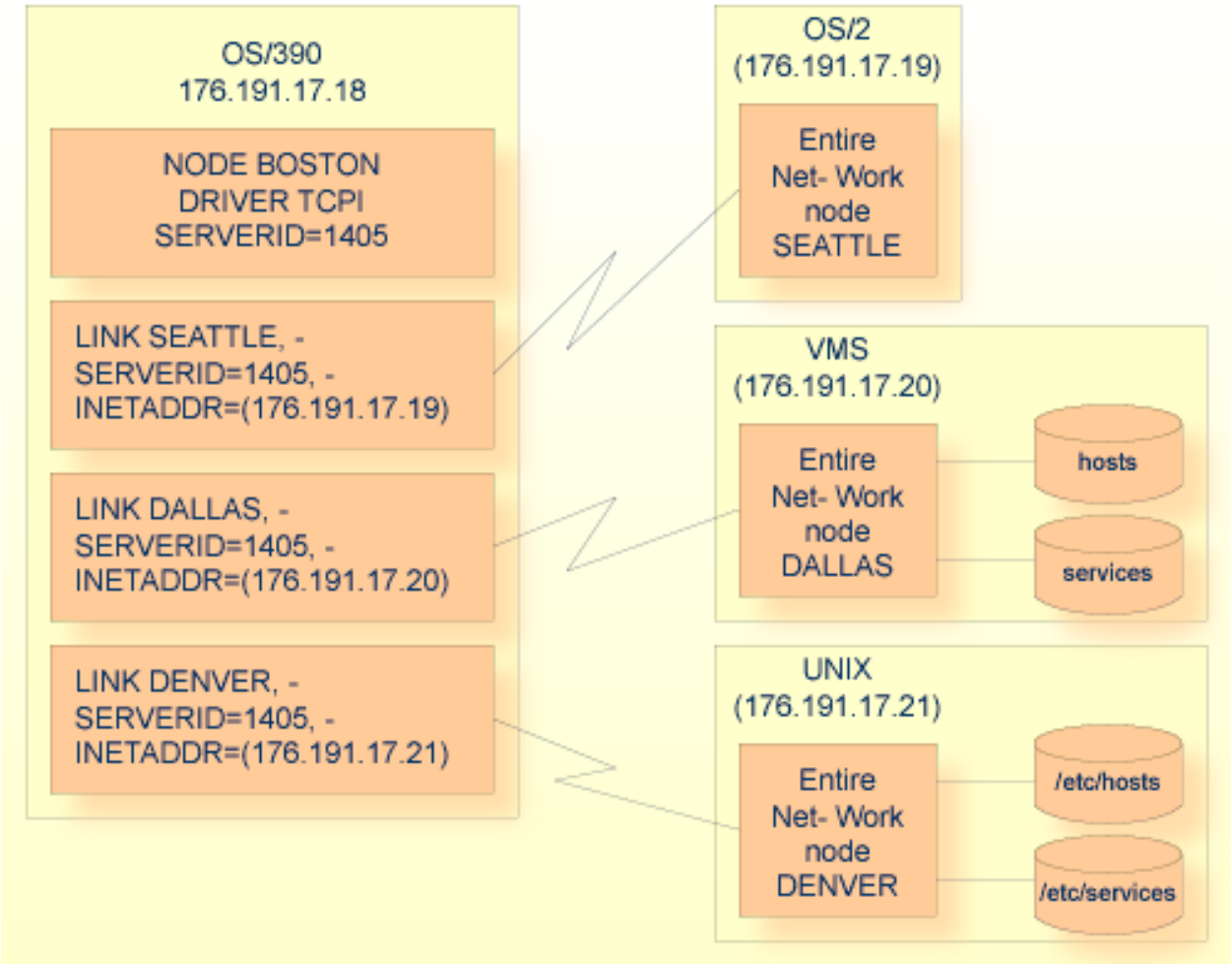
This multiple line display is produced when the STATS operator command is issued either for the TCPI driver or its links. This display is also produced when the automatic statistics interval expires and the PSTATS=Y is specified in the TCP/IP DRIVER or LINK statement. Values are displayed and updated asynchronously; therefore, the totals displayed may not always be accurate. The contents are as follows:

Line	Shows the . . .
a	name of the link or driver and the length of time since statistics were last reset or the link was last connected. Length of time is displayed in *hours:minutes:seconds* and in *seconds:milliseconds*.
b	cumulative number of bytes and messages written, and the cumulative number of API calls.
c	average number of bytes and messages written, and the average number of read API calls per second.
d	cumulative number of bytes and messages read, and the cumulative number of read API calls.
e	average number of bytes and messages read, and the average number of read API calls per second.

Line	Shows the . . .
f	cumulative number of WRITE commands that occurred. The total number of WRITEs is equal to the number of WRITEs from the Entire Net-Work task plus the average number of WRITEs from asynchronous routines.
g	average number of WRITE commands that occurred per second. The total average number of WRITEs is equal to the average number of WRITEs from the Entire Net-Work task plus the average number of WRITEs from asynchronous routines.
h	cumulative number of READ commands that occurred. The total number of READs is equal to the number of READs from the Entire Net-Work task plus the number of READs from asynchronous routines.
i	average number of READ commands that occurred per second. The total average number of READs is equal to the average number of READs from the Entire Net-Work task plus the average number of READs from asynchronous routines.

19 Configuration Example

The following diagram is an example of a TCP/IP network with its related statements and parameters:



20 User Exits

Entire Net-Work supports a user exit interface that allows the TCP/IP line driver to call an optional user exit. For more information, see the section *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.

21 Simple Connection Line Driver Overview

The Entire Net-Work Simple Connection Line Driver provides communication between client applications that use Software AG's new e-business connections and Adabas databases running in mainframe (z/OS,) and open systems environments. The new e-business connections make use of:

- an enhanced communication protocol provided by Software AG that links e-business applications with enterprise servers
- the Adabas Directory Server (ADI).

Notes:

1. Mainframe Entire Net-Work does not use the Directory Server directly. Instead, open systems Entire Net-Work open systems installations (version 7 and later) notify the mainframe node of the databases that Entire Net-Work 7 serves and the mainframe node then broadcasts this information throughout the network.
2. Mainframe-to-mainframe connections are not allowed via the Simple Connection Line Driver (TCPX).

Software AG products that support the e-business communication protocol and the Adabas Directory Server currently include Tamino, Jadabas, Entire Net-Work 7 and any other product that transports client requests using Software AG's ADALNKX module. The underlying transport mechanism is TCP/IP.

Note: The Simple Connection Line Driver cannot connect to a normal TCP/IP line driver.

The Adabas Directory Server is a centralized component that provides all directory information required to communicate between clients and servers and eliminates the need for directory configuration files on every machine. The code for the Adabas Directory Server is included in the Entire Net-Work Client code available on Software AG's Empower. To use the Simple Connection Line Driver, you must have the Adabas Directory Server installed somewhere on your system. If

you have already installed this code with another Software AG product, we recommend that you use the installed code, so that your organization uses only one shared Directory Server. For more information about the Adabas Directory Server, read *Software AG Directory Server Documentation* in the *Software AG Directory Server Installation and Administration Guide*. The documentation for Entire Net-Work Client is included with its code on Empower.

When using the new Simple Connection Line Driver, mainframe Adabas conversion must be enabled for all the databases that will be called by requests coming through the driver. For information on enabling Adabas conversion, read ***Simple Connection Line Driver Prerequisites***, elsewhere in this guide.

This chapter provides information for administrators responsible for configuring and running the Entire Net-Work Simple Connection Line Driver once Entire Net-Work is installed.

The Simple Connection Line Driver documentation is organized as follows:

How the Entire Net-Work Simple Connection Line Driver Operates	Describes how the Simple Connection Line Driver works.
Simple Connection Line Driver Prerequisites	Lists prerequisites for installing and using the Simple Connection Line Driver.
Installing Entire Net-Work TCP/IP Option Under z/OS	Describes the installation of Entire Net-Work TCP/IP Option on z/OS host systems.
Connecting to Entire Net-Work 7	Describes Entire Net-Work 7 support provided by the Simple Connection Line Driver, its limitations, and what you need to do to implement it.
TCPX DRIVER Statement	Describes the syntax and parameters of the TCPX DRIVER statement.
TCPX LINK Statement	Describes the syntax and parameters of the TCPX LINK statement.
Simple Connection Line Driver Operator Commands	Describes the operator commands you can use with the Simple Connection Line Driver.
Model Links	Describes the model link support offered by the Simple Connection Line Driver.
TCPX Statistics	Describes the statistics provided by the Simple Connection Line Driver you can use to help tune each link and the driver itself.

22 How the Entire Net-Work Simple Connection Line Driver Operates

The Entire Net-Work Simple Connection Line Driver uses TCP/IP as its transport mechanism. The Simple Connection Line Driver establishes a TCP/IP listen on the port specified in the driver SERVERID parameter. This port matches the port specified in the access URL in the Directory Server used by the ADI-enabled client. (For complete information about Adabas Directory Server, read *Adabas Directory Server Documentation* in the *Software AG Directory Server Installation and Administration Guide*.)

When a mainframe client makes an Adabas call to an open systems database, mainframe Entire Net-Work determines that the call needs to be delivered to the Entire Net-Work 7 node that serves the database. The Simple Connection Line Driver then sends the Adabas call to the Entire Net-Work 7 node.

Note: Mainframe Entire Net-Work does not use the Directory Server directly. Instead, open systems Entire Net-Work (version 7 and later) notifies the mainframe node of the databases that Entire Net-Work 7 serves and the mainframe node then broadcasts this information throughout the network.

The ADI-enabled client needs to know only the database ID. No client-side configuration or application changes are required. Client applications do not need to be modified; however, the Directory Server must be configured with the URL information for each target database.

The Directory Server does not participate in the data flow. It participates only in client-server session creation, by providing the client the correct network address of the target database server.

URLs contained in the Directory Server have the following format:

XTSaccess.*targetid***[0]=***protocol***://***host***:***port***[?***parm1***=***value***[&***parm2***=***value***]...]**

For example:

```
XTSaccess.68[0]=TCPIP://ahost:3001?retry=3
```

In this example:

Entry	Description
XTSaccess	Identifies this as a Directory Server Access entry, read by the client.
68	The target database ID.
TCPIP	The communications protocol to be used for database 68.
ahost	The name of the host computer on which the server (Entire Net-Work with the Simple Connection Line Driver runs.
3001	The port where the server (the Simple Connection Line Driver) will listen. This value matches the `SERVERID` parameter of the TCPX DRIVER statement.
retry=3	One of multiple optional parameters that may be used. The first parameter is preceded by a question mark (?) and subsequent parameters, if any, are preceded by an ampersand (&).

Because all Software AG ADI-enabled products use the Adabas Directory Server, one may already be in place and configured for your network environment. Contact Software AG if you require additional assistance.

23 Simple Connection Line Driver Prerequisites

The following are prerequisites for using the Simple Connection Line Driver:

- The Adabas Directory Server (ADI) must be installed on the client application. ADI-enabled clients are clients that use the Adabas Directory Server. For complete information about Adabas Directory Server, read *Adabas Directory Server Documentation* in the *Software AG Directory Server Installation and Administration Guide*.
- The Adabas SVC must be for any supported Adabas release. For information about currently supported Adabas releases, read *End of Maintenance*, in the *Entire Net-Work Release Notes*.
- Mainframe Adabas data conversion (Universal Encoding Support) services must be enabled for all databases.

  When using the new Simple Connection Line Driver, Adabas conversion must be enabled for all the databases that will be called by requests coming through the driver. The Simple Connection Line Driver does not provide any data conversion itself. Adabas 7 Universal Encoding Support (UES) performs any required data conversion, such as converting data for Adabas buffers between different machine architectures (ASCII, EBCDIC, Big Endian, Little Endian). For information on enabling Adabas Universal Encoding Support, read *Connecting to UES-Enabled Adabas Databases* in the *Entire Net-Work Administration Guide*.

24 Connecting to Entire Net-Work 7

When a mainframe client makes an Adabas call to an open systems database, mainframe Entire Net-Work determines that the call needs to be delivered to the Entire Net-Work 7 node that serves the database. The TCPX line driver then sends the Adabas call to the Entire Net-Work 7 node. These Adabas calls may be made using the classic Adabas control block (ACB) or the extended Adabas control block (ACBX), which supports record buffer payloads greater than 32,767 bytes.

Note: Mainframe Entire Net-Work does not use the Software AG Directory Server directly. Instead, open systems Entire Net-Work (version 7 and later) notifies the mainframe node of the databases that Entire Net-Work 7 serves and the mainframe node then broadcasts this information throughout the network. Entire Net-Work 7 must have node IDs that are unique throughout the network.

Using the Simple Connection Line Driver, you can connect your Entire Net-Work on open systems (Entire Net-Work 7 or later) to databases on the Simple Connection Line Driver node. These connections can happen in any one of these ways:

1. They can happen automatically when you start systems with Entire Net-Work 7 (or later) installed. In this case, Entire Net-Work 7 initiates the connection, assuming the Entire Net-Work 7 connection definition Manual Connection setting is off (or not set) and its Reconnect settings are appropriately defined. For more information, refer to your Entire Net-Work open systems documentation.

   Note: If you do not have Entire Net-Work 7 installed, contact your Software AG technical support representative about acquiring it or about acquiring the Entire Net-Work Client available for download from Software AG's Empower web site (*https://empower.softwareag.com*).

   When an Entire Net-Work 7 or Simple Connection Line Driver system is started, it broadcasts that it is available to its internal list of known Entire Net-Work connections. In addition to indicating that the system is now available, the broadcast connection identifies which Adabas databases the system has access to and what their availability is. The system receiving the

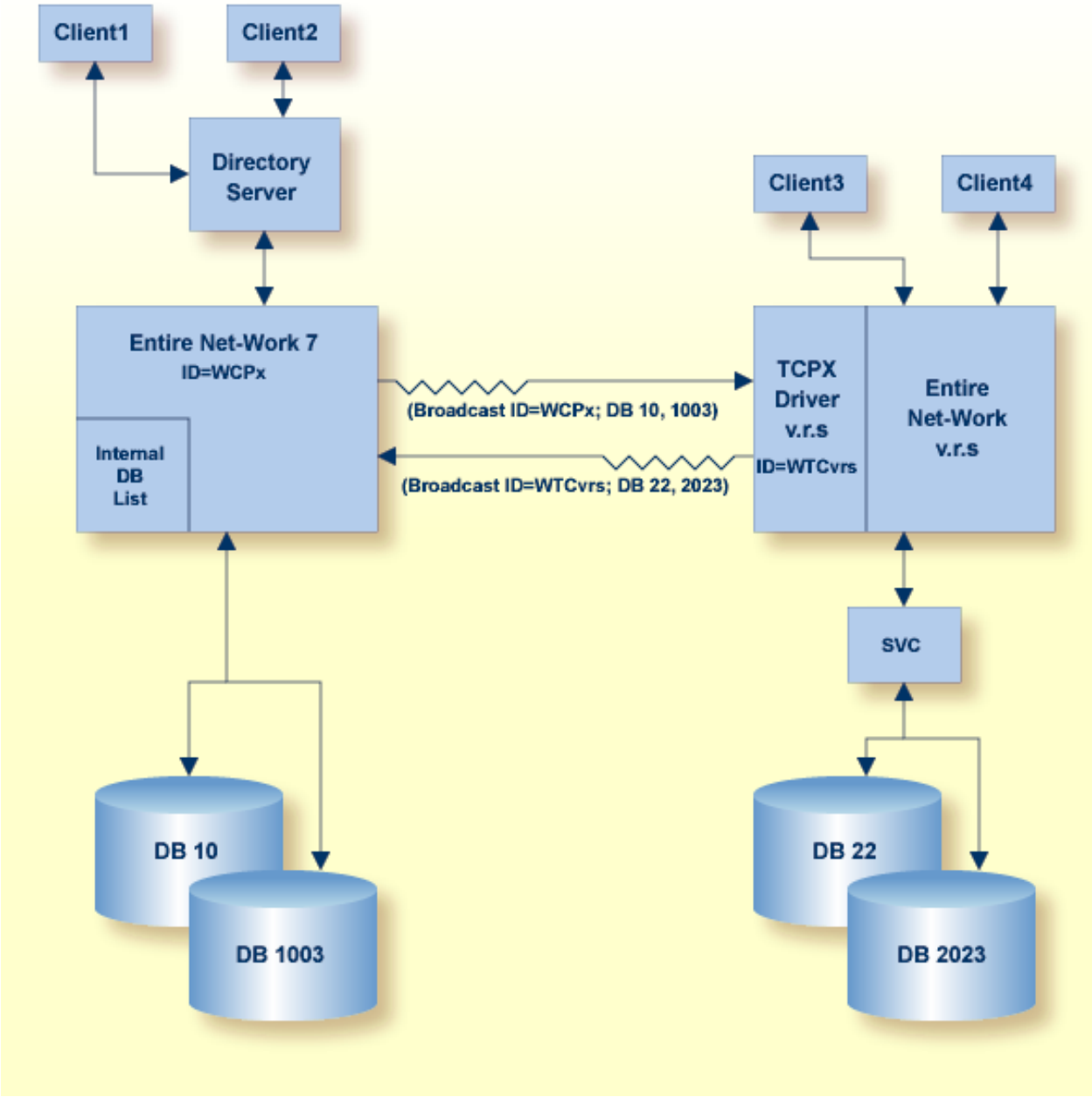
broadcast can then establish and store, through its own processing, the URL of the system that broadcasted. The end result is that Entire Net-Work 7 systems and Entire Net-Work systems with Simple Connection Line Driver installed can readily access each other's databases.

2. They can happen when you start other Entire Net-Work (mainframe) systems with the Simple Connection Line Driver installed. In this case, the Entire Net-Work Simple Connection Line Driver makes the connection. These connections can be made:

   - Manually via Simple Connection Line Driver CONNECT commands. The link must specify either the IP address or host name of the node running Entire Net-Work 7 as well as the port on which Entire Net-Work 7 lists for connections. For more information, read ***Link Commands***, elsewhere in this chapter.
   - Automatically, based on Simple Connection Line Driver LINK statement specifications and its ACQUIRE and RESTART parameter settings. The LINK statement must specify either the IP address or host name of the node running Entire Net-Work 7 as well as the port on which Entire Net-Work 7 lists for connections. For more information about the LINK statement and its ACQUIRE or RESTART parameters, read ***TCPX LINK Statement***, elsewhere in this chapter.

   Once a connection is established between two nodes, the nodes send broadcast messages back and forth that identify their databases and indicate the availability of those databases. The mainframe node then broadcasts this information throughout the network.

The following diagram depicts the interaction between systems with the Simple Connection Line Driver installed and Entire Net-Work 7 systems.



There are several restrictions and limitations to these connections.

- If you want point-to-point connections to the actual databases, a second Directory Server or a different partition in the existing Directory Server is needed. For more information, refer to your *Adabas Directory Server Documentation* in *Adabas Directory Server Administration*.
- Do not mix e-business and classic connections between open systems and mainframe Entire Net-Work nodes. All classic connections between open systems and mainframe databases should be converted to e-Business connections (TCPX on the mainframe).
- We do not recommend that you connect two mainframe nodes using the TCPX line driver.

25 TCPX DRIVER Statement

The TCPX DRIVER statement and its parameters are used to activate and define the characteristics of the local IBM mainframe node. The access method name TCPX instructs Entire Net-Work to load the line driver module NETTCPX.

In most cases, only one DRIVER statement needs to be coded in your Entire Net-Work startup parameters. However, multiple DRIVER statements can be defined to allow Entire Net-Work to listen on multiple ports.

DRIVER Statement Format

The TCPX DRIVER statement has the following format:

```
DRIVER  TCPX  [ACCEPTUI = { N | Y } ]
              [ ADI = { Y | N } ]
              [ ADIHOST = { hostname } ]
              [ ADIPORT = { port number } ]
              [ ALLOWIP6 = { Y | N } ]
              API = { CNS | EZA | HPS | OES }
              [ CONNQUE = { n | 10 } ]
              [ DRVCHAR = { character | # } ]
              [ DRVNAME = { driver-name | TCPX } ]
              [ KEEPALIV = { Y | N } ]
              [ MULTSESS = { N | Y } ]
              [ NODELAY = { N | Y } ]
              [ NUMUSERS = { number | 100 } ]
              [ OPTIONS1 = ( n , n , n , n , n , n , n , n , n , n ) ]
              [ OPTIONS2 = ( x , x , x , x , x ) ]
              [ PSTATS = { Y | N } ]
              [ RESTART = ( interval , retries } ]
              [ RSTATS = { Y | N } ]
              [ SERVERID = { n | 1996 } ]
              [ STATINT = { stat-interval | 3600 } ]
              [ SUBSYS = { subsys-name | VMCF } ]
              [ SUPMSGS = { Y | N } ]
              [ TRACE = { Y | N } ]
              [ TRACELEV = ( N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y, N | Y ) ]
              [ TRACESIZ = { size | 4096 } ]
              [ USERID = { { userid | TCPIP } } ]
                         { { nn |  00 } }
```

For more information about syntax conventions and rules used in this section, read *Conventions*.

Modifying the DRIVER Statement Parameters

The DRIVER statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. Some parameters can be modified when the line driver is open or closed. Others can be modified only when the line driver is closed. Read about the ALTER and CLOSE commands in *TCPX Operator Commands*. The open/closed requirement for each parameter is included in the parameter descriptions.

DRIVER Statement Parameters

This section describes all of the parameters that can be used for the TCPX DRIVER statement.

For more information about syntax conventions and rules used in this section, read *Conventions*.

ACCEPTUI Parameter

```
ACCEPTUI = { Y | N }
```

This optional parameter determines whether the line driver will accept connections from systems that have not been previously defined with LINK statements. The ACCEPTUI parameter can be modified when the line driver is open or closed.

Valid values are "Y" (Yes) or "N" (No).

- If "Y" is specified, Entire Net-Work will accept connection requests from an undefined system and the required control blocks are built dynamically. Normal "handshaking" procedures with the new connections are performed. This is the default.
- If "N" is specified, Entire Net-Work will reject incoming requests from unknown source nodes.

ADI Parameter

```
ADI = { Y | N }
```

Valid values are "Y" (Yes) or "N" (No).

This parameter specifies if Adabas Directory Server (ADI) support is enabled or disabled. Default is "N" for Net-work nodes, and "Y" for ADATCP.

- If "Y" is specified, Adabas Directory Server (ADI) support is enabled. This is the default for ADATCP.
- If "N" is specified, Adabas Directory Server (ADI) support is disabled. This is the default for Net-work nodes.

ADIHOST Parameter

```
ADIHOST = { hostname }
```

No default.

Valid values are 1-255 characters.

This parameter specifies the hostname of the Adabas Directory Server (ADI). The hostname is used to attempt to acquire the TCP/IP address of the system where the ADI resides. If used, ADIHOST and **ADIPORT** must both be specified.

ADIPART Parameter

```
ADIPART = {partition name}
```

No default.

Valid values are 1-32 characters.

This optional parameter specifies the partition name to be used with the Adabas Directory Server (ADI). If specified, the partition name will be included in all target entries added to the ADI by this session. Partitions are used to restrict database access; when an application queries the ADI for a target and specifies a partition, only entries with the same partition name are returned. Likewise, if the query does not specify a partition, only entries that do not have a partition are returned.

ADIPORT Parameter

```
ADIPORT = { port number }
```

No default.

Valid values are 1-65535.

This parameter specifies the port number used to communicate with the Adabas Directory Server (ADI). If used, **ADIHOST** and ADIPORT must both be specified.

ALLOWIP6 Parameter

```
ALLOWIP6 = { Y | N }
```

This optional parameter determines whether the line driver will accept connections using IPv6 communication. When the driver is opened, initialization for IPv6 communication is attempted. If the stack is not IPv6-enabled, IPv4 communication is used.

Valid values are "Y" (Yes) or "N" (No).

- If "Y" is specified, Entire Net-Work will attempt IPv6 communications when the driver is opened. ALLOWIP6=Y is only a valid specification if the API parameter has been set to one of the following values: BS2, HPS, OES, or EZA.

  Note: If ALLOWIP6=Y is set in a BS2000 environment, it is listening on the IPv6 address. Connection using an IPv4 address cannot be made.

- If "N" is specified, Entire Net-Work will not attempt IPv6 communications when the driver is opened.

API Parameter

```
API = { BS2 | CNS | EZA | HPS | OES }
```

This required parameter specifies the name of the TCP/IP application program interface being used. The API parameter can be modified only when the line driver is closed. Supported values are shown in the table below. There is no default.

Value	Description	Valid for Platforms
BS2	Loads the BS2000/OSD interface NWTCPBS2	BS2000
CNS	Loads the z/VSE interface NWTCPCNS	z/VSE
EZA	Loads the z/VSE interface NWTCPEZA. This option can be used only with the TCP/IP stack from Barnard Software, Inc.	z/VSE
HPS	Loads the IBM interface NWTCPHPS (High Performance Native Sockets)	z/OS and OS/390
OES	Loads the IBM OpenEdition sockets interface NWTCPOES	z/OS and OS/390

CONNQUE Parameter

```
CONNQUE = {n | 10}
```

This optional parameter specifies the number of connect queue entries. The value specified must accommodate the maximum number of simultaneous connection requests from remote nodes.

After the connection is accepted or rejected, connect queue entries are reused. If the value of this parameter is not high enough, the API routine is not able to process the incoming connection and the partner eventually will time out. Depending on the API being used, a message may be displayed indicating that an error has occurred. Values can range from 1 to 64; a value greater than 64 is reset to 64. The default value is 10. The CONNQUE parameter can be modified only when the line driver is closed.

DRVCHAR Parameter

```
DRVCHAR = {char | #}
```

This optional parameter specifies the special character used to designate that an operator command should be directed to this line driver rather than to a specific link. The DRVCHAR parameter can be modified only when the line driver is closed.

The default for this parameter is "#".

DRVNAME Parameter

```
DRVNAME = {name | TCPX}
```

This optional parameter specifies the 4-byte driver name. The DRVNAME parameter can be modified only when the line driver is closed.

The default for this parameter is "TCPX".

The DRVNAME parameter enables sites to make multiple TCP/IP API routines available at the same time. For example, the IBM APIs can be made available within the same Entire Net-Work address space. This parameter also allows two or more drivers to be defined so that Entire Net-Work can listen on multiple ports simultaneously.

KEEPALIV Parameter

```
KEEPALIV = { Y | N}
```

This optional parameter allows you to maintain connections when there is no other traffic with the remote links. Valid values are "Y" or "N."

- When this value is set to "Y", it causes internal TCP messages to be sent periodically to all remote links, thus maintaining the connections when there is no other traffic with the remote links. The amount of time between messages is determined by an initialization parameter in the TCP stack.
- When this value is set to "N", internal TCP messages are no longer sent periodically and the connections are not maintained.

The default for this parameter is "N".

KEEPALIV can also be set for individual remote links. For more information, read about the KEEPALIV parameter associated with the **TCPX LINK statement**.

MULTSESS Parameter

```
MULTSESS = { N | Y }
```

This optional parameter determines whether a connect request from a host that has an active connection is treated as a new link. This parameter can be modified when the line driver is open or closed.

A value of "Y" indicates that the connect request is treated as a new link; a value of "N" indicates that the connect request is rejected.

The default for this parameter is "Y".

NODELAY Parameter

```
NODELAY = { N | Y }
```

This optional parameter allows you to indicate whether the IBM socket option TCP_NODELAY is enabled or disabled for a link. TCP_NODELAY indicates whether data sent over the socket is subject to the Nagle algorithm (RFC 896). For more information, refer to your IBM documentation.

Valid values for this parameter are "Y" (the TCP_NODELAY option is enabled) or "N" (the TCP_NODELAY option is disabled). The default is "Y". When the NODELAY parameter on the DRIVER statement is specified, you do not need to specify the NODELAY parameter on the LINK statement. The value from the DRIVER statement is used. If the NODELAY parameter is not specified on either the DRIVER or LINK statements, a value of "Y" is assumed.

Note: The setting of this parameter is only effective if the API parameter is also set to "OES" or "HPS."

NUMUSERS Parameter

```
NUMUSERS = { number | 100 }
```

This parameter specifies the estimated maximum number of concurrently active clients. For performance reasons, a table of individual client entries is preallocated based on this number. During the Entire Net-Work session, if the number of active clients is exceeded, the table is automatically expanded by 50% of the current value. The size of each entry in the table is 256 bytes. The minimum value is 10, maximum is 32767. The default is 100.

Note: This parameter can only be altered when the driver is closed.

OPTIONS1 Parameter

```
OPTIONS1 = (n,n,n,n,n,n,n,n,n,n)
```

This optional parameter allows up to ten numeric API-specific options to be set. The values can be modified when the line driver is open or closed. There are no default values.

Not all APIs use the OPTIONS1 parameter.

The BS2000/OSD (BS2) API uses only two of the OPTIONS1 fields (prior to Entire Net-Work version 5.8, nine fields were used):

- The first, second, third, fourth, and fifth values are no longer used and must be set to 0.
- The sixth value is the Sockets task tracing level:
  - A value of 0 inhibits any tracing.
  - Values 1 and 2 give the corresponding level of high-level logic flow.
  - Values 3 through 9 log the transferred data and invoke the FSC sockets tracing.

  Tracing should be used only under the direction of Software AG.
- The seventh value is the maximum number of connections between the BS2 API and the Entire Net-Work partners. It is used for storage allocation by the Sockets task. The valid range is 2-2048 and the default value is 2048.
- The eighth and ninth values are no longer used and must be set to 0 or omitted.

OPTIONS2 Parameter

```
OPTIONS2 = (x,x,x,x,x)
```

This optional parameter allows up to five alphanumeric API-specific options to be set. The values can be modified when the line driver is open or closed. There are no default values.

Not all APIs use the OPTIONS2 parameter.

Beginning with Entire Net-Work version 5.8, the BS2000/OSD API (BS2) does not use the OPTIONS2 parameter at all.

PSTATS Parameter

```
PSTATS = { Y | N }
```

This optional parameter determines whether or not statistics are printed.

A value of "Y" indicates that statistics should be printed to DDPRINT when the statistics interval expires; a value of "N" indicates that the statistics should not be printed.

This parameter does not affect the STATS command and can be modified when the driver is open or closed.

The default for this parameter is "N".

RESTART Parameter

```
RESTART = (interval, retries)
```

This optional parameter specifies the retry interval in seconds (*interval*) and the number of retries (*retries*) that Entire Net-Work will attempt to reopen the access method with the API after a shutdown due to a failure. The RESTART parameter can be modified when the line driver is open or closed.

If RESTART is not specified, or *interval* is specified as zero, no retry is attempted. By specifying (*retries*) as zero, an infinite number of retries can be requested.

The RESTART parameter is particularly useful with the Simple Connection Line Driver when Entire Net-Work is started at IPL and communication with the API is unsuccessful because TCP/IP is not yet fully initialized. Using this parameter, you can instruct Entire Net-Work to reopen the TCP/IP session, thereby giving TCP/IP sufficient time to become active.

The TIMER parameter on the NODE statement affects the RESTART parameter (see the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*.) The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead.

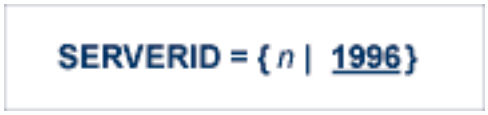
RSTATS Parameter

This optional parameter determines whether or not statistics are reset. A value of "Y" indicates that statistics should be reset when the statistics interval expires; a value of "N" indicates that the statistics should not be reset. The default is "N".

The RSTATS parameter can be modified when the line driver is open or closed.

The statistics interval value is set by link parameter `STATINT`. If LINK parameter `STATINT` is not define, the interval is set by the DRIVER parameter `STATINT`.

SERVERID Parameter



This optional parameter specifies a well known port number used by Entire Net-Work while awaiting connection requests from participating Entire Net-Work partners. Values may range from 1 to 65535. The SERVERID parameter can be modified only when the line driver is closed.

When specified in a DRIVER statement, the SERVERID parameter specifies the port number of the Entire Net-Work being initialized. If SERVERID is not specified for a link, the SERVERID specified for the driver is used as the default port for the link.

The default for this parameter is 1996. Only the DRIVER statement has a SERVERID parameter.

STATINT Parameter

```
STATINT = {interval | 3600}
```

This optional parameter specifies the amount of time, in seconds, before statistics are automatically printed or reset. The default is 3600. The STATINT parameter can be modified when the line driver is open or closed.

Acceptable values range from 1 to 2147483647. Any value outside this range is in error.

SUBSYS Parameter

```
SUBSYS = {name | VMCF}
```

This parameter specifies the name of the subsystem to be accessed by the API routines that use subsystem control blocks in interaddress space communications. The default value is VMCF. The SUBSYS parameter can be modified only when the line driver is closed.

In a z/OS environment, the IBM API routines communicate to the system address space by locating the subsystem control table and retrieving the information required to perform cross-memory communication. If the subsystem is specified incorrectly, the driver is not able to perform its open processing and no connections are possible.

This parameter is not used in a z/VSE or BS2000 environment.

SUPMSGS Parameter

```
SUPMSGS = { Y | N }
```

This optional parameter suppresses printout of the following messages in the TCPX DRIVER: NETP818I and NETP819I. This allows you to avoid cluttering your logs with these connection and disconnection messages.

Note: This functionality is available in Entire Net-Work TCP/IP Option 6.3 SP2 or later installations and only in 6.3 SP2 installations after applying zap WT632005.

Valid values are "Y" (Yes) or "N" (No):

- If "Y" is specified, the NETP0818I Connect, and NETP0819I Disconnect messages are suppressed (no longer output to the log).
- If "N" is specified, the NETP0818I and NETP0819I messages are output to the log.

TRACE Parameter

```
TRACE = { Y | N }
```

This parameter indicates whether tracing for this line driver should be active (Y) or not (N). When tracing is activated, trace information is placed in the trace table. The default is N (no). The TRACE parameter can be modified when the line driver is open or closed.

This is equivalent to specifying `TRACE=linedriver-code` or `TRON=linedriver-code` in the NODE statement (for example, `TRACE=CTCA`).

TRACELEV Parameter

```
TRACELEV = ( Y | N, Y | N, Y | N, Y | N, Y | N, Y | N, Y | N, Y | N, Y | N, Y | N )
```

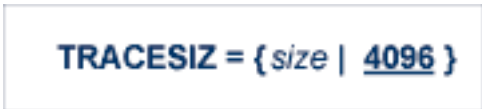
This optional parameter specifies the levels of tracing that the line driver will perform. It is a series of flags that determine which events are traced. The TRACELEV specification must be enclosed in parentheses. For example:

```
TRACELEV=(N,N,N,N,N,N,N,N,N,N)
```

Trace levels are positional within the parameter syntax and are set using Y (Yes) or N (No). It is recommended that all settings within the TRACELEV parameter be N. If your system experiences problems, contact your Software AG technical support representative for the settings that produce the appropriate trace information. The TRACELEV parameter can be modified when the line driver is open or closed.

Note: The tracing information provided is sent to the DDPRINT data set. In addition to setting the TRACELEV flags, the trace must also be turned on using either the DRIVER statement parameter `TRACE=Y` or the operator command `TRACE=linedriver-name`. Tracing dramatically affects the overall performance and throughput of Entire Net-Work.

TRACESIZ Parameter



This optional parameter specifies the size, in bytes, of the driver-specific trace table.

This parameter is also used as the default size of the link specific trace table when the LINK statement does not include a TRACESIZ specification.

Valid values can range from 4096 to 4194304. A value less than 4096 is reset to 4096; a value greater than 4194304 is reset to 4194304.

The TRACESIZ parameter can be modified only when the line driver is closed.

The default for this parameter is "4096".

USERID Parameter

This parameter's value can be modified only when the line driver is closed.

- For IBM APIs (i.e., the NWTCPIBM interface, the NWTCPHPS interface, and the NWTCPOES interface), the USERID parameter specifies the name of the started task, job, or virtual machine in which the IBM TCP/IP protocol stack is running. The value is 1-8 characters. The default value is TCPIP.
- For the CNS API (i.e., the NWTCPCNS interface), the USERID parameter is used to direct traffic to a particular TCP/IP stack. The value is a two-digit number that must match the ID= value in the PARM field of the TCP/IP stack. The default value is 00, which is also the default for the TCP/IP stack. The value of USERID is not validated; if its value does not match the ID= value of an active TCP/IP stack, the TCP/IP driver will fail to open and it will return messages similar to the following:

```
NETP571W TCP API ERROR ON OPEN - RC = 0008
NET0101I  TCPI DRIVER OPEN FAILED - RC = 0004
```

To use multiple TCP/IP stacks, one DRIVER statement must be provided for each stack, and each link must specify the driver associated with the stack it will use. When defining multiple drivers, copy the module NETTCPI.phase and change the last four characters of the name to match the DRIVER name. For example, to define the following, NETTCPI would be copied (not renamed) to NETTCP2:

```
*Links using Driver TCPI use TCP/IP stack with ID=00
DRIVER TCPI  API=CNS
LINK PC01 TCPI,INETADDR=(x,x,x,x)
```

```
*Links using Driver TCP2 use TCP/IP stack with ID=02
DRIVER TCP2 API=CNS,USERID=02,DRVNAME=TCP2
LINK PC02 TCP2,INETADDR=(x,x,x,x)
```

WCPPART Parameter

WCPPART is an alias of **ADIPART** and works in the same way as ADIPART.

26 TCPX LINK Statement

The LINK statement and its parameters are used to define the characteristics of the remote client or an Entire Net-Work 7 (or later) Kernel. With the Simple Connection Line Driver, links used by direct clients are not normally predefined; they are dynamically allocated as clients initiate communication. However, links may be predefined to override defaults or provide some control over clients. Links to Entire Net-Work 7 Kernels are normally predefined, allowing the connection to be initiated on the mainframe side.

Note: Mainframe-to-mainframe connections are not allowed via the Simple Connection Line Driver (TCPX).

LINK Statement Format

The TCPX LINK statement has the following format:

```
LINK linkname TCPX [ ACQUIRE = { Y | N } ]
                  [ ADJHOST = Internet-host-name ]

                    [ INETADDR = ( n1.n2.n3.n4 ) ]
                    [ V6IPADDR = x : x : x : x : x : x : x : x ]

                  [ KEEPALIV = { N | Y } ]
                  [ MULTSESS = { Y | N } ]
                  [ NODELAY = { N | Y } ]
                  [ PSTATS = { N | Y } ]
                  [ RESTART = ( interval , retries ) ]
                  [ RSTATS = { Y | N } ]
                  [ SAF = { Y | L | N } ]
                  [ SENDTIME = { n | 90 } ]
                  [ SERVERID = port-number ]
                  [ STATINT = { stat-interval | 3600 } ]
                  [ TRACESIZ = size ]
                  [ WEIGHT = { n | 256 } ]
```

For more information about syntax conventions and rules used in this section, read *Conventions*.

Modifying the LINK Statement Parameters

The LINK statement parameters are read from a sequential file during system startup, and can be modified after startup using the `ALTER` operator command. Some parameters can be modified when the link is open or closed. Others can be modified only when the link is closed. Read about the `ALTER` and `CLOSE` commands in the section *TCPX Operator Commands*. The open/closed requirement for each parameter is included in the parameter descriptions.

LINK Statement Parameters

This section describes all of the parameters that can be used for the TCPX LINK statement.

linkname Parameter

```
linkname
```

The required *linkname* parameter specifies the name by which this link is to be known. It is positional, and must be specified immediately after the LINK keyword and immediately before the driver name (TCPX); the link name must be unique on the node. All operator commands affecting the link must specify this name.

If the link name begins with the characters "MODEL", the link is defined as a model link. See the section ***Model Links***.

TCPX Parameter

```
TCPX
```

TCPX is required and specifies the protocol name that defines the driver associated with this link. It must be the same as the value specified for the **DRVNAME parameter** in the **TCPX DRIVER statement**. In most cases, this value is "TCPX". If DRVNAME is changed to a value other than "TCPX", this parameter must also be changed.

ACQUIRE Parameter

```
ACQUIRE={N | Y}
```

This optional parameter specifies whether or not a connection with the remote node should be attempted when the driver is opened for the first time (during system initialization). If "N" is specified, the link is connected manually using operator commands from the client or server node. If "Y" is specified, the link is attempted automatically when the system initializes. The default value is "N".

The ACQUIRE parameter can be modified only when the link is closed.

ADJHOST Parameter

```
ADJHOST=Internet-host-name
```

This optional parameter specifies the Internet host name of a node with which a connection is to be established. The name can be resolved to either an IPv4 or IPv6 address. Its value can be 1 - 255 characters. The ADJHOST parameter can be modified only when the link is closed.

The ADJHOST parameter uses Domain Name Services (DNS), as follows:

- The GetHostByName function is used to determine the IP address of a host name specified with ADJHOST. IP address is used both for connecting to another node and for locating the link for an incoming connection.
- The GetHostByAddr function is used to determine the host name of a node that is trying to connect to this node. This is necessary when the IP address of a host name specified with ADJHOST changes after the link has been opened.

Software AG recommends the use of the ADJHOST parameter for sites that assign IP addresses via the DHCP protocol. Entire Net-Work will use the GetHostByName function for every outgoing connection on nodes that have ADJHOST specified as long as INETADDR is not specified.

The following table lists the APIs that support Domain Name Services:

API	GetHostByName	GetHostByAddr
BS2	Yes	Yes
CNS	Yes	No
EZA	Yes	Yes
HPS	Yes	Yes
OES	Yes	Yes

For performance reasons, Software AG recommends that all LINK statements containing an ADJHOST value be defined after the LINK statements containing an INETADDR or V6IPADDR specification. If none of these parameters is specified, the link is not usable. If both the INETADDR and ADJHOST parameters are specified, the INETADDR parameter takes precedence. Likewise, if both the V6IPADDR and ADJHOST parameters are specified, the V6IPADDR parameter takes precedence. The INETADDR and V6IPADDR parameters are mutually exclusive; only one of them can be specified in the same link.

INETADDR Parameter

```
INETADDR=(n1.n2.n3.n4)
```

This optional parameter specifies the IPv4 (Internet Protocol) address of the remote host associated with this link. The INETADDR parameter can be modified only when the link is closed.

IP address is used both for connecting to another node and for locating the link for an incoming connection. It is provided to Entire Net-Work in the form of INETADDR=(n1,n2,n3,n4) or INETADDR=(n1.n2.n3.n4) where each value represents 8 bits of the 32-bit IP address. Acceptable values are between 0 and 255 and may be separated by commas or periods. For example:

```
INETADDR=(157,182,17,20) ↩
```

or

```
INETADDR=(157.182.17.20) ↩
```

On most UNIX-based machines, this address can be found in the /etc/hosts file. The following are examples of the information in the /etc/hosts file:

157.182.17.20 DALLAS dallas (VMS)

157.182.17.18 DENVER denver (UNIX)

Each host on the INTERNET is assigned a unique IP address which is used by the IP and higher level protocols to route packets through the network. The IP address is logically made up of two parts: the network number and the local address. This IP address is 32 bits in length and can take on different formats or classes. The class defines the length (number of bits) of each part. There are four classes (A, B, C, and D); the class is identified by the allocation of the initial bit.

For performance reasons, Software AG recommends that all LINK statements containing an INET-ADDR value be defined before the LINK statements containing an ADJHOST specification. If neither of these parameters or the V6IPADDR parameter are specified, the link is not usable. If both INETADDR and ADJHOST are specified, the INETADDR parameter takes precedence.

Note: Do not specify both V6IPADDR and INETADDR in the same link statement; they are mutually exclusive parameters.

KEEPALIV Parameter

```
KEEPALIV={Y | N}
```

KEEPALIV=Y (Yes) causes internal TCP messages to be sent periodically to the remote node, thus maintaining the connection when there is no other traffic with the node. The amount of time between messages is determined by an initialization parameter in the TCP stack. If no KEEPALIV value is specified for the link, it defaults to the KEEPALIV value on the **DRIVER statement**.

MULTSESS Parameter

```
MULTSESS={N | Y}
```

This optional parameter determines whether a connect request from a host that already has an active connection is treated as a new link. A value of "Y" indicates that the connect request is treated as a new link; a value of "N" indicates that the connect request is rejected. The default value is the value specified for the MULTSESS parameter in the **TCPX DRIVER statement** (N or Y, with Y as the default). The MULTSESS parameter can be modified when the link is open or closed.

NODELAY Parameter

This optional parameter allows you to indicate whether the IBM socket option TCP_NODELAY is enabled or disabled for a link. TCP_NODELAY indicates whether data sent over the socket is subject to the Nagle algorithm (RFC 896). For more information, refer to your IBM documentation.

Valid values for this parameter are "Y" (the TCP_NODELAY option is enabled) or "N" (the TCP_NODELAY option is disabled). The default is "Y". When the NODELAY parameter on the DRIVER statement is specified, you do not need to specify the NODELAY parameter on the LINK statement. The value from the DRIVER statement is used. If the NODELAY parameter is not specified on either the DRIVER or LINK statements, a value of "Y" is assumed.

Note: The setting of this parameter is only effective if the API parameter is also set to "OES" or "HPS."

PSTATS Parameter

```
PSTATS={N | Y}
```

This optional parameter determines whether or not (Y or N) statistics are printed to DDPRINT when the statistics interval expires. The default value is the value specified for the PSTATS parameter in the **TCPX DRIVER statement** (for which the default is N). This parameter does not affect the `STATS` operator command. The PSTATS parameter can be modified when the link is open or closed.

RESTART Parameter

```
RESTART= (i,n)
```

This optional parameter specifies the retry interval in seconds (*i*) and the number of retry attempts (*n*) that are made to start the connection to the remote node. If RESTART is not specified, or (*i*) is specified as zero, no retry is attempted. By specifying (*n*) as zero, an infinite number of retries can be requested.

The TIMER parameter on the NODE statement affects the RESTART parameter (see the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*. The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead.

The RESTART parameter can be modified when the link is open or closed.

RSTATS Parameter

```
RSTATS={N | Y}
```

This optional parameter determines whether or not (Y or N) statistics are automatically reset when the statistics interval expires. The default value is the value specified for the RSTATS parameter in the **TCPX DRIVER statement**. The RSTATS parameter can be modified when the link is open or closed.

SAF Parameter

```
SAF={Y | L | N}
```

If SAF=Y or SAF=L is specified, Entire Net-Work will call the SAF Interface for all incoming requests on this link; failure to load the Interface is considered a security violation and Entire Net-Work will shut down. If SAF=L, the calls are traced and the output directed to DDPRINT. An error code is transmitted to the user if access to SAF is denied. The SAF parameter can be modified when the link is open or closed. The default value is N (No).

SENDTIME Parameter

```
SENDTIME={time | 90}
```

This optional parameter specifies the time (in seconds) that the Simple Connection Line Driver allows for a send to complete. When this time is exceeded, the line driver writes a message to the operator console indicating a possible error condition on the remote node. The connection is considered severed and link disconnect processing is initiated. The default value is 90 seconds. The SENDTIME parameter can be modified when the link is open or closed.

SERVERID Parameter

```
SERVERID=port-number
```

This parameter specifies a well known port number associated with a remote partner. Entire Net-Work will attempt to use this port number when connecting to the remote partner. If this parameter is not specified, or is set to zero (0), the value specified for the SERVERID parameter on the **DRIVER statement** is used. The SERVERID parameter can be modified only when the link is closed.

STATINT Parameter

```
STATINT={ interval | 3600}
```

This optional parameter specifies the amount of time, in seconds, before statistics are automatically printed or reset. Acceptable values are 1 - 2147483647. Any value outside this range is in error. The default value is 3600. The STATINT parameter can be modified when the link is open or closed.

TRACESIZ Parameter

```
TRACESIZ=size
```

This optional parameter specifies the size of the TCP/IP link specific trace table. Value can be 4096 - 4194304. A value less than 4096 is reset to 4096. A value greater than 4194304 is reset to 4194304. The default value is the value specified for the TRACESIZ parameter in the **TCPX DRIVER statement**. The TRACESIZ parameter can be modified only when the link is closed.

V6IPADDR Parameter

```
V6IPADDR=x:x:x:x:x:x:x:x
```

This optional parm specifies the IPv6 address of the remote host associated with this link. The V6IPADDR parameter can be modified only when the link is closed. Specify the address as *x:x:x:x:x:x:x:x*, where each *x* represents a hexadecimal value of the eight 16-bit pieces of the address.

Standard abbreviated forms are allowed:

- Leading zeros in an individual value may be omitted;
- Groups of zeros may be eliminated, specifying them as a double colon '::'. A double colon may be used only once in an address.
- IPv4-compatible and IPv4-mapped IPv6 addresses can be specified.

The following are examples of valid IPv6 addresses:

```
2001:DB8:7654:3210:FEDC:BA98:7654:3210
2001:DB8:0:0:8:800:200C:417A
2001:DB8::8:800:200C:417A
::13.1.68.3
::FFFF:129.144.52.38
```

For performance reasons, Software AG recommends that all LINK statements containing a V6IPADDR value be defined before the LINK statements containing an ADJHOST specification. If neither of these parameters or the INETADDR parameter are specified, the link is not usable. If both V6IPADDR and ADJHOST are specified, the V6IPADDR parameter takes precedence.

Note: Do not specify both V6IPADDR and INETADDR in the same link statement; they are mutually exclusive parameters.

WEIGHT Parameter

```
WEIGHT={n | 256}
```

This parameter specifies the weight of this link with respect to other links going to the same node. If a given target can be reached by more than one path (chain of connected links), the path with the lowest weight is used. Slow or expensive links should be given a higher value than fast or inexpensive links. Values range from "1" to "999999". The default value is "256".

The WEIGHT parameter can be modified only when the link is closed.

27 Simple Connection Line Driver Operator Commands

Entire Net-Work's Simple Connection Line Driver has the ability to process operator commands that are directed to a specific link or directly to the driver.

Operator Command Syntax

Under z/OS, the Simple Connection Line Driver operator commands have the following format:

F NETWORK, TCPX *target cmd*

The following table describes this syntax.

Syntax Representation	Description
TCPX	Informs Entire Net-Work that the command is destined for the Simple Connection Line Driver. If more than one **TCPX DRIVER statement** exists, use the name specified on the DRVNAME parameter of the **DRIVER statement** instead of TCPX.
target	A value that informs Entire Net-Work what the target of the command is, as follows: ■ Specify an asterisk (*) if the target is all links. ■ Specify the DRVCHAR value ("#" is the default) if the target is the driver itself (see the DRVCHAR parameter on the ***TCPX DRIVER Statement***). ■ Specify the link name if the target is a specific link.
cmd1, *cmd2*, and *cmdx*	The operator commands to be carried out. Multiple commands can be specified in a single command statement. When the `ALTER` command is specified, it must be the last command in the statement, because everything following the `ALTER` command is treated as a DRIVER or LINK statement parameter. One or more DRIVER or LINK statement parameters must be specified.

Examples

The following are examples of Simple Connection Line Driver operator commands:

```
F NETWORK,TCPX * CLOSE
```

```
TCPX # STATS
```

Driver Commands

The Entire Net-Work Simple Connection Line Driver supports the commands listed in the following table when the target is the driver. The underlined portion of the command is the minimum abbreviation.

Command	Action
ALTER *driver-parms*	Dynamically changes the driver configuration. The ALTER command is followed by the driver configuration parameters to be altered. The driver configuration parameters are the same as those specified in the DRIVER statement. For example: `TCPX # ALTER ACCEPTUI=Y`
CLOSE	Disconnects and closes all links that are connected to other nodes. Releases all resources held by the driver as well as all open links. Closes the driver.
OPEN	Reopens the driver after it is closed with the CLOSE operator command or because of an access method failure. Allocates all the resources needed by the driver to communicate with TCP/IP. Also attempts to resolve any unresolved host names.
RESET	Resets all statistics for the driver. Statistics are printed only if the STATS command precedes the RESET command.
SHOW	Displays the current configuration of the driver. The current configuration is always shown automatically following an ALTER command.
SNAP	Causes all control blocks specific to the link to be snapped (printed in hexadecimal). Driver-specific control blocks and Entire Net-Work specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. This command has no effect on the next automatic printing of statistics. To print and reset statistics, specify RESET immediately after the STATS command. For example: `TCPX # STATS RESET`
STATUS	Displays the current status of the driver as well as a count of messages received and sent.
TRACE	Causes the Simple Connection Line Driver to format and print the driver-specific trace table. The trace table is formatted and printed in hexadecimal automatically when the SNAP command is processed.
USERS	Displays the Adabas user ID in character and hexadecimal formats, the Context ID and Context Verifier values (these are part of the internal message header and can be used to help identify the client), and the number of database calls received for the client.

Note: When the driver is closed, it does not recognize the commands CLOSE, STATS, or RESET.

Link Commands

The Entire Net-Work Simple Connection Line Driver supports the commands listed in the following table when the target is a link or all links. The underlined portion of the command is the minimum abbreviation.

Command	Action
ALTER *link-parms*	Dynamically changes the link configuration. The ALTER command is followed by the link configuration parameters to be altered. The link configuration parameters are the same as those specified on the LINK statement. For example: `TCPX LINK1 ALTER ADJHOST=DALLAS`
CLOSE	Disconnects the link if it is connected to another node and releases all resources held by the link.
CONNECT ↩	Attempts to establish one or more TCPX sessions with the target link(s). If the link is already connected or is in the process of connecting, the command is ignored.
DISCONNECT	Starts the disconnect sequence for the target link(s). If the link is already disconnected or is in the process of disconnecting, the command is ignored.
LOGLON *linkname*	Turns on selective logging for the specified link.
LOGLOFF *linkname*	Turns off selective logging for the specified link.
OPEN	Allocates all the resources needed by the link to communicate with TCPX. Does not initiate a connect to the remote node. The status of the link displayed via the SHOW operator command is not affected by the OPEN request.
RESET	Resets all statistics for the link. Statistics are printed only if the STATS command precedes the RESET command.
RESUME	Restarts processing on a link that was temporarily stopped due to a SUSPEND command.
SHOW	Displays the current configuration of the link. The current configuration is always shown automatically following an ALTER command.
SNAP	Causes all link-specific control blocks and the link-specific trace table to be snapped (printed in hexadecimal). Driver-specific control blocks and Entire Net-Work-specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. This command has no effect on the next automatic printing of statistics. To print and reset statistics, specify RESET immediately after the STATS command. For example: `TCPX LINK1 STATS RESET`
STATUS	Displays the current status of the link as well as a count of messages received and sent.

Command	Action
SUSPEND	Temporarily stops all processing on a link. Processing can be restarted with the RESUME command.
TRACE	Causes the link-specific trace table to be formatted and printed. The trace table is formatted and printed in hexadecimal automatically when the SNAP command is processed.
USERS	Displays the Adabas user ID in character and hexadecimal formats, the IP address for the link, the Context ID and Context Verifier values (these are part of the internal message header and can be used to help identify the client), and the number of database calls received for the client.

28 Model Links

The Simple Connection Line Driver supports dynamically added links, thus reducing the time required to set up and maintain the TCPX LINK statements. Model links can be coded and used to define new links as they are added.

The Simple Connection Line Driver is permitted to add links dynamically if ACCEPTUI=Y is coded on the DRIVER statement. A new link block is created and is used to control all further communications on the link. The link block can be initialized with default values that will be applied to each new link. Alternatively, one or more model links can be defined to override the values contained in the link block.

The model link statement is identical to other LINK statements, except that the link name begins with the characters 'MODEL'. Most of the model link parameters, such as PSTATS and RSTATS, are copied into the dynamically built link block. Some parameters, such as INETADDR, are not copied because they are truly link-specific.

29 Simple Connection Line Driver Statistics

The Entire Net-Work Simple Connection Line Driver issues API calls to communicate with TCP. To help tune each link and the driver itself, the Simple Connection Line Driver provides the statistics shown below. The statistics for a link and the driver are identical, with the exception of the title line (a).

```
     +-------------------------------------------------------------------------+
(a)  + Statistics For Driver   TCPX  Period     0:02:42 (    162.325 Secs)  +
     + -------------  ---Bytes----  --Messages--  -Api Calls--  ----------- +
(b)  + Writes             496.394K         3,412         3,412  Total       +
(c)  +                      3.063K            21            21  Per Second  +
(d)  + Reads                0.000K             0         3,451  Total       +
(e)  +                      0.000K             0            21  Per Second  +
     + -------------  ---Total----  ----Task----  ---Other----  ----------- +
(f)  + Write Cmd's                0             0             0  Total       +
(g)  +                            0             0             0  Per Second  +
(h)  + Read Cmd's                 0             0             0  Total       +
(i)  +                            0             0             0  Per Second  +
     +-------------------------------------------------------------------------+
```

This multiple line display is produced when the STATS operator command is issued either for the TCPX driver or its links. This display is also produced when the automatic statistics interval expires and the PSTATS=Y is specified in the TCPX DRIVER or LINK statement. Values are displayed and updated asynchronously; therefore, the totals displayed may not always be accurate. The contents are as follows:

Line	Shows the . . .
a	name of the link or driver and the length of time since statistics were last reset or the link was last connected. Length of time is displayed in *hours:minutes:seconds* and in *seconds:milliseconds*.
b	cumulative number of bytes and messages written, and the cumulative number of API calls.
c	average number of bytes and messages written, and the average number of read API calls per second.
d	cumulative number of bytes and messages read, and the cumulative number of read API calls.

Line	Shows the . . .
e	average number of bytes and messages read, and the average number of read API calls per second.
f	cumulative number of WRITE commands that occurred. The total number of WRITEs is equal to the number of WRITEs from the Entire Net-Work task plus the average number of WRITEs from asynchronous routines.
g	average number of WRITE commands that occurred per second. The total average number of WRITEs is equal to the average number of WRITEs from the Entire Net-Work task plus the average number of WRITEs from asynchronous routines.
h	cumulative number of READ commands that occurred. The total number of READs is equal to the number of READs from the Entire Net-Work task plus the number of READs from asynchronous routines.
i	average number of READ commands that occurred per second. The total average number of READs is equal to the average number of READs from the Entire Net-Work task plus the average number of READs from asynchronous routines.

Index

K

L

M

N

O

P

R

S

T

U

V

W

Z