CTC Option Administration Guide

Administration

Version 6.5.1

October 2019

ADABAS & NATURAL

This document applies to Entire Net-Work CTC 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: WCC-DOC-651-20191115

Table of Contents

Preface

This document provides information for administrators responsible for configuring and running the Entire Net-Work CTCA line driver or the FCTC line driver once Entire Net-Work is installed.

Note: The CTCA line driver and FCTC line driver are provided in the Software AG product option called the Entire Net-Work CTC Option (product code WCC), which is an add-on to the Entire Net-Work product and must be ordered separately.

The CTCA line driver and FCTC line driver documentation is organized as follows:

Entire Net-Work CTC Option Release Information	Provides a high-level explanation of this release of Entire Net-Work CTC Option.
Overview	Introduces you to the CTCA line driver and the FCTC line driver.
Installing the CTCA Line Driver or FCTC Line Driver in z/OS	Explains how to install the CTCA line driver and the FCTC line driver in z/OS environments.
FCTC line driver Compatibility	Describes FCTC line driver compatibility and how to activate it.
Sample JCL and Parameter Statements	Provides examples of JCL and parameter statements used to run the CTCA line driver and the FCTC line driver.
Multiple CTCA Configurations	Describes multiple CTCA configurations.
CTCA and FCTC DRIVER Statements	Describes the syntax and parameters of the CTCA DRIVER and FCTC DRIVER statements.
CTCA and FCTC LINK Statements	Describes the syntax and parameters of the CTCA LINK and FCTC LINK statements.
CTCA and FCTC Operator Commands	Describes the operator commands you can use with the CTCA line driver and the FCTC line driver.

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 CTC Option Release Information

Read this chapter carefully before installing and using Entire Net-Work CTC Option version 6.4.

For complete information about the installation media for Entire Net-Work 6.4 and about new and changed features for Entire Net-Work 6.4, read *Entire Net-Work 6.4 Release Information* in the *Entire Net-Work Release Notes*.

New Features for Version 6.3 and above

This section summarizes the changes made to the Entire Net-Work CTC Option in this release:

1. Effective with version 6.3 SP2, zEnterprise Data Compression (zEDC) is supported on z/OS systems. This support comes in the form of two new LINK parameters for both the CTCA and FCTC line drivers: ZEDC and ZEDCLOG. The ZEDC parameter indicates whether zEDC compression can occur for the link; the ZEDCLOG parameter identifies the level of trace data that should be logged for zEDC compression processing. For more information, read ***CTCA and FCTC LINK Statements***(elsewhere in this guide).
2. The following table summarizes the new Entire Net-Work CTC Option DRIVER and LINK parameters introduced in Entire Net-Work 6.3:

Parameter	DRIVER or LINK Statement?	New or Changed?	Description	Introduced in Release
ZEDC	LINK	New	This new parameter indicates whether zEnterprise Data Compression (zEDC) compression can occur for the link.	6.3 SP2
ZEDCLOG	LINK	New	This new parameter indicates what level of trace data will be logged for zEDC compression processing.	6.3 SP2

Compatibility with Earlier Versions of the Line Drivers

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

As of version 6.3, the FCTC line driver is incompatible with previous releases of FCTC. To address this, specifying `COMPAT=Y` enables compatibility mode, which detects a connection to a prior version of FCTC and automatically converts that connection to a compatible 6.2 SP2 connection.

For complete information about the COMPAT parameter, read ***FCTC Line Driver Compatibility*** and ***CTCA and FCTC DRIVER Statements***, elsewhere in this guide.

4 Overview

Entire Net-Work channel-to-channel line drivers provide high speed, high volume data transfer capabilities using:

- channel-to-channel adapters (CTCAs);
- IBM 3088 Multisystem Channel Communication Units;
- suitably configured ESCON or FICON channels; or
- compatible products from other vendors.

The devices (or device address exposures) used by Entire Net-Work are dedicated and cannot be shared with other users, such as VTAM, JES, or GRS. However, any such CTCA or FCTC (described later) users can run in parallel with Entire Net-Work as long as each is using separate device addresses. It is also possible to use a CTCA with different products at different times.

In practice, the number of channel-to-channel links that Entire Net-Work can establish is limited only by storage constraints or by the capacity of z/OS systems to attach peripheral devices.

The Entire Net-Work CTCA line driver observes its own proprietary channel programming protocol that is not compatible with other software using CTCAs. It is compatible with earlier supported versions of the Entire Net-Work channel-to-channel line driver. As of version 6.3 SP1, FCTC channels are not compatible with previous versions.

In the CTCA line driver, channel-to-channel adapters are driven by attention interrupts. Entire Net-Work's CTCA line driver includes a line driver module and an attention routine module. Both modules are unloaded from the installation media into your Entire Net-Work load library.

Note: The CTCA and FCTC line drivers page-fix all required storage themselves; they do not use the Entire Net-Work buffer pool manager's page-fixed storage.

The Entire Net-Work fast channel-to-channel line driver (FCTC) is a newer version of the CTCA line driver. In the FCTC line driver, changes have been made in the channel program logic flow to improve performance. Once connection is established, the channel designated as the READ channel will issue READ CCWs in anticipation of WRITEs from the other side. This reduces disconnect time on the WRITE side of the channel, and greatly reduces the number of interrupts, thus improving performance. When installing and using the FCTC line driver, you must pay attention to differences in the parameters and parameter settings used by the CTCA line driver.

Note: The FCTC driver *requires* that two channels be assigned and paired. If two channels cannot be assigned, do not attempt to use this driver. The existing CTCA driver must be used for single channel arrangements.

FCTC LINKS Operating in Impaired Mode

FCTC LINKs can function even if only one unit is active; this is referred to as operating in *impaired mode*. In this case, the active unit reverts to bidirectional communication similar to a CTCA link.

- If one unit is unavailable when the link is connected, the connection will succeed and the link will function in impaired mode. Note however that both the UNITREAD and UNITWRT parameters must always be specified on the FCTC LINK statement.
- If a unit fails while the link is active, for example due to a hardware error, the other unit reverts to bidirectional communication, allowing the link to continue to function in impaired mode. Normally this happens seamlessly, and the message NETC126W is issued. However, in rare cases, depending on the state of the channels and the messages that were in progress at the time of the failure, the link must be disconnected and reconnected in order to properly clean up. The reconnect can be done by operator command or automatically via the RESTART parameter. The link will then function impaired until the problem can be resolved.

If a link is active in impaired mode, after the failed unit is repaired, the link must be disconnected and reconnected in order for both units to become active again. Note that if the unit was previously offline, the connect may need to be initiated on the side that was offline, in order for the unit to be properly allocated.

Tip: Use the DISPLAY LINKS operator command or the FCTC **STATUS** operator command to review the status of the individual units. For more information (including the syntax of the commands), read *DISPLAY Command*(in the *Entire Net-Work Reference Guide*) and ***CTCA and FCTC Operator Commands***(elsewhere in this guide).

5 Installing the CTCA Line Driver or FCTC Line Driver in z/OS

The Entire Net-Work CTCA line driver and FCTC line driver currently support the z/OS operating system only. The following table describes most of the libraries included on the release 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 CTC 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
WCC*vrs*.LOAD	The z/OS load library for Entire Net-Work CTC Option. The *vrs* in the library name represents the *version* of Entire Net-Work CTC Option.

If Entire Net-Work is not already installed, follow the instructions in the section *Entire Net-Work Installation* in the *Entire Net-Work Installation Guide*. Then perform the following additional steps:

1. Using IEBCOPY, unload the Entire Net-Work CTCA line driver or FCTC line driver library from the installation media into your Entire Net-Work load library.

   Refer to the cover letter accompanying your installation media and to the *Software AG Product Delivery Report* for information about media format.

   The library contains the following modules:

NETCTCA	The CTCA line driver module
NCHXAR	The CTCA line driver attention routine module
NETFCTC	The FCTC line driver module
NCHXARF	The FCTC line driver attention routine module

2. Add DRIVER and LINK statements that specify your requirements to the Entire Net-Work parameter data set.

   For an example, see the section ***Sample JCL and Parameter Statements***.

3. The CTCA line driver and FCTC line driver require APF authorization. To make Entire Net-Work APF-authorized, ensure that:

   - all load libraries in your STEPLIB concatenation have been defined to z/OS as authorized libraries; and
   - module ADARUN has been linked with AMODE(31) and AC(1).

4. Start Entire Net-Work (on multiple nodes) and run verification tests, such as executing operator commands and accessing remote databases.

6 FCTC Line Driver Compatibility

The 6.3 FCTC line driver is incompatible with previous releases of FCTC. To address this incompatibility in version 6.3 SP1, an Entire Net-Work CTC Option L001 library was made available to you that enabled compatible FCTC connections between 6.3 and 6.2 SP2 nodes. This L001 library provided code that allowed a 6.3 node to detect a connection to a prior version of FCTC and automatically convert that connection to a compatible 6.2 SP2 connection.

This FCTC compatibility processing has been automatically included in Entire Net-Work CTC Option 6.3 SP2. You no longer need the L001 library. Activation of compatibility mode is controlled using an FCTC DRIVER parameter called COMPAT. For complete information about the COMPAT parameter, read ***CTCA and FCTC DRIVER Statements***, elsewhere in this guide.

No changes to your 6.2 SP2 nodes are required. However, if FCTC line driver compatibility has been activated, Entire Net-Work must *not* be started with REUSASID=YES.

Activating FCTC Line Driver Compatibility Processing

By default, FCTC line driver compatibility processing is already activated when you install version 6.3 SP2 or later. However, you can follow these steps to verify that activation is in place.

To activate FCTC line driver compatibility processing, complete the following steps:

1 Edit your FCTC line driver DRIVER statement and verify it includes the COMPAT parameter set to "Y". (To deactivate FCTC line driver compatibility processing, you would set the COMPAT parameter to "N".) For more information, read ***CTCA and FCTC DRIVER Statements***, elsewhere in this guide.

2 Start Entire Net-Work 6.3.

Note: If FCTC line driver compatibility has been activated, Entire Net-Work must *not* be started with REUSASID=YES.

The FCTC line driver compatibility code is activated and ready to be used.

Note: No changes are necessary to your existing Entire Net-Work version 6.2 SP2 nodes.

How It Works

This section describes the internal process used by the Entire Net-Work CTC Option to handle FCTC connections once compatibility is activated and a 6.3 FCTC line driver is defined.

When an Entire Net-Work version 6.3 node initializes, both the 6.3 FCTC line driver and a version 6.2 SP2 FCTO line driver are initialized. The FCTO line driver uses the same parameters that were specified for the FCTC driver.

Caution: Do NOT define the FCTO line driver in the Entire Net-Work startup parameters.

When a connection attempt with an Entire Net-Work 6.2 SP2 node (the partner connection) is detected, the connection is automatically converted to the FCTO line driver, as follows:

1. The version 6.3 link is closed and marked disabled.
2. If a pair of matching FCTO links is found, these are used for the connection. If no matching FCTO links exist, a pair of FCTO links is created. The parameters for these links are determined from the parameters used for the version 6.3 link; so no formal FCTO line driver definition should be attempted.

   The name of the new link is based on the length of the version 6.3 link name and checks for duplicate link names. If the 6.3 link name is less than eight characters long, an "R" is appended to the link name to create the FCTO read link name and a "W" is appended to create the FCTO write link name. If the 6.3 link name is eight characters long, the eighth character is replaced by "R" or "W" to create the read and write FCTO link names. If the newly created FCTO links have names that are duplicated by others in the system, then the link name is generated in the format FCTO*xnnn*, where *x* is "R" or "W" (to identify the read or write link), and *nnn* is a unique positive integer.
3. The connection is established using the pair of FCTO links.

When the partner node is upgraded to Entire Net-Work 6.3, the connection is automatically converted back to a version 6.3 connection as follows:

1. The pair of FCTO links are closed and marked disabled.
2. The original version 6.3 FCTC link is located and reopened.
3. The connection is established using the original 6.3 FCTC link. You do not need to restart the existing 6.3 node after upgrading the partner node; the connection happens automatically.

Important Usage Notes

The following important usage notes should be remembered:

1. Do *not* define the FCTO line driver in your Entire Net-Work parameters, the way you would for other line drivers. The FCTO line driver parameters are initialized automatically as they are needed when FCTC compatibility is activated.
2. Do *not* make any changes to your existing Entire Net-Work 6.2 SP2 nodes to implement compatibility with version 6.3 nodes.
3. Operator commands can be issued to the FCTO line driver and links. However, do not use operator commands to attempt to convert between the two types of connections; this is handled automatically.
4. You do *not* need to restart the version 6.3 node once the connection partner has also upgraded to 6.3. The connection will automatically be converted back to a 6.3 connection.
5. You *cannot* use the FCTO line driver to establish connections between two version 6.3 nodes.
6. If FCTC line driver compatibility is *not* activated, and an FCTC connection to a version 6.2 SP2 node is attempted, the connection will fail with error NETC121E.
7. If FCTC line driver compatibility has been activated, Entire Net-Work must *not* be started with REUSASID=YES.

7 Sample JCL and Parameter Statements

This section provides examples of JCL and parameter statements for running the CTCA line driver and the FCTC line driver.

Sample procedure for Entire Net-Work with CTCA line driver and the FCTC line driver:

```
//NETWORK     PROC     F=N
//NETWORK     EXEC     PGM=ADARUN,TIME=1440
//STEPLIB      DD      DSN=NETWRK.Vvrs.LOAD,DISP=SHR
//             DD      DSN=WAL.Vvrs.LOAD,DISP=SHR
//DDPRINT      DD      SYSOUT=*
//MPMDUMP      DD      SYSOUT=*
//SYSDUMP      DD      SYSOUT=*
//DDCARD       DD      DISP=SHR,DSN=NETWORK.SOURCE(DDCARD)
//             DD      DISP=SHR,DSN=NETWORK.SOURCE(FORCE&F)
//DDKARTE      DD      DISP=SHR,DSN=NETWORK.SOURCE(DDKARTE)
```

Sample parameter statements (DDKARTE) for Entire Net-Work with the CTCA line driver:

```
NODE ONE
*
DRIVER CTCA TRACE=Y,IORETRY=2
LINK TOTWO  CTCA UNIT=380,WE=1
LINK TOFOUR CTCA UNIT=3C0,WE=1
```

Sample parameter statements (DDKARTE) for Entire Net-Work with the FCTC line driver:

```
DRIVER FCTC TRACE=N,IORETRY=5
LINK CTC-1 FCTC IORETRY=0,                    -
               UNITR=xxxx,                    -
               UNITW=xxxx,                    -
               MAXIOTIM=300,                  -
               RCVBLKSZ=4K,RCVBLKCT=80,       -
               SNDBLKSZ=4K,SNDBLKCT=80,       -
```

```
                PSTATS=N,RSTATS=N,STATINT=100, -
                WEIGHT=10,                     -
                RESTART=(300,5)
```

Both the CTCA and the FCTC line drivers require proper tuning of the RCVBLKSZ, RCVBLKCT, SNDBLKSZ and SNDBLKCT parameters for optimum performance. Larger xxxBLKSZ values enable multiple messages to be blocked together, and larger xxxBLKCT values reduce the likelihood of delays due to buffer shortages.

8 Multiple CTCA Configurations

The channel-to-channel adapter is a fast device, and Entire Net-Work provides even better performance by driving it directly rather than through some other access method.

Where applications require even better performance than can be provided by a single CTCA, Entire Net-Work allows you to use multiple CTCAs in parallel. The following discussion is provided to help you design your Entire Net-Work environment to meet your needs.

In single CTCA configurations, contention is one of the major reasons for performance problems when the load is increased. Since both nodes must send their messages over the same CTCA, they will occasionally attempt to do so simultaneously. In this case:

- only one node is allowed to send;
- the other will receive a busy indication, requeue its send buffer, perform the steps necessary to receive the other node's message, then attempt to resend its own message.

This attempt may fail again, however, if the other node gets its next message out first. The busier the CTCA, the more often this problem occurs.

Contention creates overhead for requeuing and retrying the failing I/Os. The problem is too much load on the device. The answer is to use multiple CTCAs. To use multiple CTCAs, parallel links can be specified.

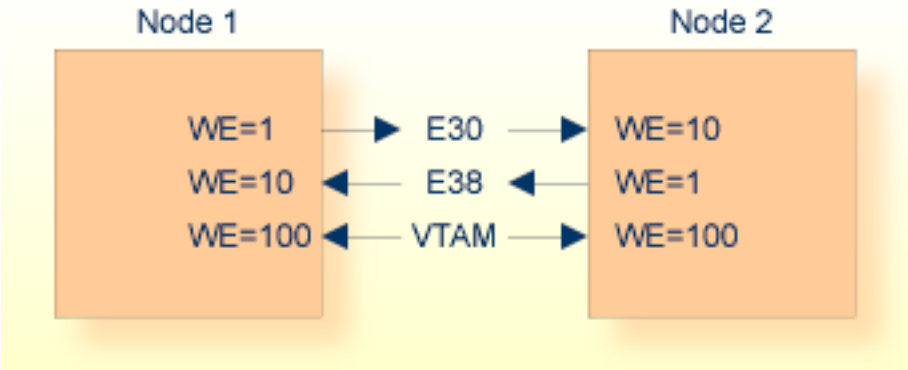
Parallel Links

Parallel links are no different from other links except that they connect the same two nodes. With CTCA parallel link routing, the link with the lowest WEIGHT parameter value is always selected as the sending link; the link with the higher WEIGHT parameter value is the receiving link. With FCTC, the UNITREAD and UNITWRT parameters are used to define which units are used for receiving and sending. Any other links are considered backups; that is, they are not to be used unless the "best" link goes down. From then on, the "next best" link is selected. When the "best" link again becomes available, it is automatically selected for data traffic.

Because only one link is ever selected as the sending link, there is no need to define more than two links in parallel except to use as backup links. To reduce system paging overhead, the number of receive buffers defined can be reduced where a lower number is deemed adequate. (See the RCVBLKCT parameter in the section ***CTCA and FCTC LINK Statements*** .)

CTCA Parallel Link Example

The following example shows a two-node network with a pair of CTCAs configured as parallel unidirectional links, and a VTAM link as backup (to be used only if both CTCAs stop working).



Parameters for Node 1:

```
NODE ONE ...
*
DRIVER CTCA
LINK TO2A   CTCA UNIT=E30,WE=1
LINK TO2B   CTCA UNIT=E38,WE=10
*
DRIVER VTAM  APP=NETWRK1
LINK TO2V    VTAM APP=NETWRK2,WE=100
```

Parameters for Node 2:

```
NODE TWO ...
*
DRIVER CTCA
LINK TO1A   CTCA UNIT=E30,WE=10
LINK TO1B   CTCA UNIT=E38,WE=1
*
DRIVER VTAM  APP=NETWRK2
LINK TO1V    VTAM APP=NETWRK1,WE=100
```

The same unit addresses are used for both nodes because, in this example, device addresses are the same on both systems.

Each of the two nodes will select only the link with weight WE = 1. Thus

- traffic from Node 1 to Node 2 is sent over CTCA E30 (link TO2A); while
- traffic from Node 2 to Node 1 is sent over CTCA E38 (link TO1B).

If one of the two CTCAs malfunctions, traffic will be routed automatically through the other CTCA, causing some performance degradation due to contention. Only if both CTCAs become unavailable is the VTAM link used.

If the weight of the VTAM link were set to WE=5, the priorities would be reversed: if either of the CTCAs malfunctioned, the respective node would immediately start using the VTAM link instead. Running one CTCA with contention generally provides better performance, even if VTAM also uses a CTCA for its connection.

9 CTCA and FCTC DRIVER Statements

The CTCA DRIVER statement defines the operating characteristics of the standard channel-to-channel line driver (CTCA) for the Entire Net-Work Communicator, while the FCTC DRIVER statement defines the operating characteristics of the fast channel-to-channel driver (FCTC). The DRIVER statement must follow an Entire Net-Work NODE statement, but can be preceded by non-CTCA and non-FCTC DRIVER and related LINK statements. Only one CTCA DRIVER statement can be specified for a node; likewise, only one FCTC DRIVER statement can be specified for a node. However, both a CTCA DRIVER and an FCTC DRIVER statement can be specified for the same node.

DRIVER Statement Format

The CTCA and FCTC DRIVER statements have the following format:

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

DRIVER Statement Parameters

This section describes all of the parameters that can be used for a CTCA or FCTC DRIVER statement.

- COMPAT Parameter
- IORETRY Parameter
- TRACE Parameter

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

COMPAT Parameter

This parameter applies only to the FCTC line driver; it cannot be used with the CTCA line driver.

Use this parameter to activate FCTC line driver compatibility mode, allowing version 6.3 nodes to make FCTC connections to nodes running prior versions. For details about FCTC line driver compatibility, read *FCTC Line Driver Compatibility*, elsewhere in this guide.

Valid values for this parameter are "Y" (activate compatibility) or "N" (deactivate compatibility). The default is "Y".

Note: If FCTC line driver compatibility has been activated, Entire Net-Work must *not* be started with REUSASID=YES.

IORETRY Parameter

```
IORETRY = {number | 3}
```

This parameter applies to both the FCTC and CTCA line drivers.

Use this parameter to specify the number of times the driver will retry an I/O operation that it decides can be retried. The valid range is 0-15. The default value is 3.

TRACE Parameter

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

This parameter applies to both the FCTC and CTCA line drivers.

Use this parameter to indicate 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`).

10 CTCA and FCTC LINK Statements

The CTCA LINK statement is used to specify the channel-to-channel connections to other Entire Net-Work nodes, while the FCTC LINK statement is used to specify fast channel-to-channel connections to other Entire Net-Work nodes. Multiple links can be specified between any pair of nodes. See the section ***Multiple CTCA Configurations*** for information about configuring more than one CTCA between two nodes to increase throughput.

The total amount of real storage required for each CTCA or FCTC link is calculated from the values specified for the send and receive buffers (the size of the buffers multiplied by the number of buffers) and the size of the trace table. A buffer equal to the calculated amount is obtained with GETMAIN and is page-fixed. This amount of real storage is required as long as the link is open.

Notes:

1. FCTC links in Entire Net-Work 6.3 cannot connect to FCTC links from previous releases of Entire Net-Work. If connections to previous releases are required, use the CTCA line driver or another line driver.
2. FCTC links can function even if only one unit is active; this is referred to as operating in *impaired mode*. For more information, read ***FCTC LINKS Operating in Impaired Mode***, elsewhere in this guide.

CTCA LINK Statement Syntax

The CTCA LINK statement has the following format:

```
LINK linkname    CTCA    [ IORETRY = { retries | DRIVER statement value } ]
                         [ MAXIOTIM = { seconds | 60 } ]
                         [ PSTATS = { N | Y } ]
                         [ RCVBLKCT = { buffercount | 4 } ]
                         [ RCVBLKSZ = { buffersize | 8192 } ]
                         [ RESTART = { ( interval , retries | (0, 83647) } ]
                         [ RSTATS = { Y | N } ]
                         [ SAF = { Y | L | N } ]
                         [ SNDBLKCT = { buffercount | 4 } ]
                         [ SNDBLKSZ = { buffersize | 8192 } ]
                         [ STATINT = { interval | 3600 } ]
                         [ TRACESIZ = { tablesize | 8192 } ]
                         UNIT = unitaddress
                         [ WEIGHT = { linkvalue | 256 } ]
                         [ ZEDC = { Y | N } ]
                         [ ZEDCLOG = { F | L | N } ]
```

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

FCTC LINK Statement Syntax

The FCTC LINK statement has the following format:

```
LINK linkname    FCTC    [ IORETRY = { retries | DRIVER statement value } ]
                         [ MAXIOTIM = { seconds | 60 } ]
                         [ PSTATS = { N | Y } ]
                         [ RCVBLKCT = { buffercount | 4 } ]
                         [ RCVBLKSZ = { buffersize | 8192 } ]
                         [ RESTART = { ( interval , retries | (0, 83647) } ]
                         [ RSTATS = { Y | N } ]
                         [ SAF = { Y | L | N } ]
                         [ SNDBLKCT = { buffercount | 4 } ]
                         [ SNDBLKSZ = { buffersize | 8192 } ]
                         [ STATINT = { interval | 3600 } ]
                         [ TRACESIZ = { tablesize | 8192 } ]
                         UNITREAD = unitreadaddress
                         UNITWRT = unitwriteaddress
                         [ WEIGHT = { linkvalue | 256 } ]
                         [ ZEDC = { Y | N } ]
                         [ ZEDCLOG = { F | L | N } ]
```

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 the operator commands ALTER and CLOSE in the section *CTCA and FCTC Operator Commands* .) The open/closed requirement for each parameter is included in the following descriptions.

LINK Statement Parameters

This section describes the CTCA and FCTC LINK statement parameters in alphabetic order.

All parameters apply to both CTCA and FCTC line drivers, with the exception of the UNITREAD and UNITWRT parameters (which apply only to the FCTC line driver) and the UNIT parameter (which applies only to the CTCA line driver).

- ZEDCLOG Parameter

IORETRY Parameter

IORETRY = { *retries* | DRIVER statement value }

This parameter is used to specify the number of times the driver will retry an I/O operation that it thinks can be retried. The valid range is 0 - 15. If no value is specified, the value from the DRIVER statement is used.

MAXIOTIM Parameter

MAXIOTIM = { *seconds* | 60 }

This parameter specifies the maximum amount of time, in seconds, in which an I/O must complete before a link is considered hung and is automatically disconnected. The default value is 60 seconds.

PSTATS Parameter

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.

RCVBLKCT Parameter

RCVBLKCT = { *buffercount* | 4 }

This parameter specifies the number of receive buffers defined with the parameter RCVBLKSZ.

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

RCVBLKSZ Parameter

RCVBLKSZ = { *buffersize* | 8192 }

This parameter specifies the block size for the receive buffers used in I/O operations. The value specified must be in multiples of 4,096 (4K) bytes. Other values are rounded automatically. The minimum value is 4,096; the maximum value is 61,440. Values outside these limits are reset automatically to the accepted minimum or maximum value. No message is produced when values are automatically rounded or reset. The value may optionally be followed by the multiplier "K" (which indicates the value is multiplied by 1024).

The value of RCVBLKSZ must equal the value of SNDBLKSZ that is specified on the remote side of the link.

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

RESTART Parameter

RESTART specifies first the retry interval in seconds (*interval*) and the number of retries (*retries*) to be used if you want Entire Net-Work to attempt to reconnect the link following a shutdown due to a failure. If RESTART is not specified, or the interval is specified as zero, no retry is attempted. By specifying the number of retries 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*section.) 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.

Note: Unlike other line drivers, this driver does not support a RESTART parameter on the DRIVER statement. The reason for this is that there is no global resource (an access method, such as VTAM, for example) that could fail.

RSTATS Parameter

The RSTATS parameter specifies whether or not to reset link statistics automatically at intervals specified by the STATINT parameter. RSTATS=N (No) is the default value.

SAF Parameter

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).

SNDBLKCT Parameter

SNDBLKCT = { *buffersize* | 4 }

This parameter specifies the number of send buffers defined with the parameter SNDBLKSZ.

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

SNDBLKSZ Parameter

SNDBLKSZ = { *buffersize* | 8192 }

This parameter specifies the block size for the send buffers used in I/O operations. The value specified must be in multiples of 4,096 (4K) bytes. Other values are rounded automatically. The minimum value is 4,096; the maximum value is 61,440. Values outside these limits are reset automatically to the accepted minimum or maximum value. No message is produced when values are automatically rounded or reset. The value may optionally be followed by the multiplier "K" (which indicates the value is multiplied by 1024).

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

The value of SNDBLKSZ must equal the value of RCVBLKSZ that is specified on the remote side of the link.

STATINT Parameter

The STATINT parameter specifies the time interval, in seconds, between automatically printing or resetting statistics. The default value is 3600 seconds. See the PSTATS parameter and the RSTATS parameter.

TRACESIZE Parameter

TRACESIZE = { *tablesize* | 8192 }

The TRACESIZE parameter specifies the size of the trace table for each link, in multiples of 64 bytes. The minimum value is 4096 bytes; there is no maximum value. TRACESIZE = 8192 is the default value. The TRACESIZE parameter can be modified only when the link is closed.

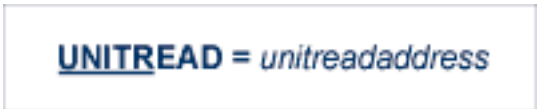
UNIT Parameter

UNIT = *unitaddress*

This parameter specifies the unit address of the CTCA link. The unit address is 3 or 4 characters, which can be any combination of 0-9 and A-F. The valid range is 0000-FFFF. The UNIT parameter can be modified only when the link is closed.

This parameter does not apply to FCTC links.

UNITREAD Parameter



This parameter is required for FCTC links; it does not apply to CTCA links. It specifies the address of the unit that will be used to read data. The unit address is 3 or 4 characters, which can be any combination of 0-9 and A-F. The valid range is 0000-FFFF. The UNITREAD parameter can be modified only when the link is closed.

UNITWRT Parameter

This parameter is required for FCTC links; it does not apply to CTCA links. It specifies the address of the unit that will be used to send data. The unit address is 3 or 4 characters, which can be any combination of 0-9 and A-F. The valid range is 0000-FFFF. The UNITWRT parameter can be modified only when the link is closed.

WEIGHT Parameter

The WEIGHT parameter allows a performance- or cost-based value to be assigned to each link, in order to provide Entire Net-Work with information on which to base path selection if more than one path to a destination is available. A higher value should be used for slower links and a lower value for faster links. Values may 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)*.

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

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)*.

11 CTCA and FCTC Operator Commands

Entire Net-Work's CTCA line driver and FCTC line driver have the ability to process operator commands that are directed to a specific link. The operator commands allow you to dynamically reconfigure any link and to monitor link operations.

Operator Command Syntax

Under z/OS, the operator commands have the following format:

The following table describes this syntax.

Syntax Representation	Description
CTCA	Informs Entire Net-Work that the command is destined for the CTCA line driver.
FCTC	Informs Entire Net-Work that the command is destined for the FCTC line driver.
target	A value that informs CTCA or FCTC what the target of the command is, as follows: ■ Specify an asterisk (*) if the target is all links. ■ Specify the link name if the target is a specific link.
cmd	The operator command to be carried out.

Examples

The following are examples of operator commands:

```
F NETWORK,CTCA * CLOSE
```

```
F NETWORK,CTCA link3 CONNECT
```

Description

The following table lists the CTCA and FCTC operator commands. The underlined portion of the command is the minimum abbreviation.

Command	Action
ALTER *parameters*	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: `F NETWORK,CTCA LINK1 ALTER STATINT=180` 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 a CTCA or FCTC session with the target link. If the link is already connected or is in the process of connecting, the command is ignored.
DISC	Starts the disconnect sequence for the target link. If the link is already disconnected or is in the process of disconnecting, the command is ignored.
OPEN	Allocates all the resources needed by the link to communicate with CTCA or FCTC. Does not initiate a connect to the remote node.
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: `STATS RESET`
STATUS	Causes the immediate printing of the link status as well as the number of messages sent and received. For FCTC links, the status of each unit is also displayed.
SUSPEND	Temporarily stops all processing on a link. Processing can be restarted with the RESUME command.

Command	Action
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.

Index

S

T

U

V

W

Z