Entire Net-Work Administration

Entire Net-Work Reference

Version 5.9.1

September 2009

Entire Net-Work

This document applies to Entire Net-Work Administration Version 5.9.1 and to all subsequent releases.

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

Table of Contents

1 Statement and Command Reference

ADARUN Control Statements	Describes the ADARUN function, which is used to invoke Entire Net-Work.
Entire Net-Work Parameter Statements	Describes the Entire Net-Work parameter statements that define the Entire Net-Work environment.
Entire Net-Work Operator Commands	Describes the Entire Net-Work operator commands.
Diagnostic Utilities	Describes the Entire Net-Work diagnostic utilities provided in the Entire Net-Work load library.

2 Conventions

Notation *vrs* or *vr*: When used in this documentation, the notation *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 = { IBM | HPS | BS2 | CMS | CNS | FUJ | ILK | OES } In the example above, you must select IBM, HPS, BS2, CMS, CNS, FUJ, ILK, 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 brackets, one and only one of the elements must be supplied.	DRIVER TCPI API = { IBM | HPS | BS2 | CMS | CNS | FUJ | ILK | OES } In this example, one of the following values is required for the API parameter: IBM, HPS, BS2, CMS, CNS, FUJ, ILK, 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, DRIVER, or TRANSDEF) must be specified as the first non-blank item on the statement.
- The node name, driver name, link name, or translation definition function 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, it 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, . . .
```

3 ADARUN Control Statements

This section covers the following topics:

ADARUN Function

The ADARUN function is used to invoke Entire Net-Work. ADARUN invokes the Entire Net-Work control program and:

- loads the module ADAIOR, which performs all operating system-dependent functions;
- reads and interprets all ADARUN parameter statements;
- loads the modules needed to execute the functions specified by the ADARUN parameters;
- performs any necessary modifications to those load modules, based on the specified parameters;
- passes control to Entire Net-Work.

The ADARUN statement is fully described in the Operations documentation of Adabas.

ADARUN Statement Format

The syntax of ADARUN statement and parameters is:

ADARUN *parameter = value,...*

ADARUN statements must:

- Contain the word "ADARUN" in positions 1-6, followed by "parameter=value" strings of one or more entries;
- Have one or more blanks, beginning in position 7, between "ADARUN" and the first "parameter=value"; and
- Not extend beyond position 72 of a line.

ADARUN control statements may contain multiple statement lines. Each line must be specified as a separate statement according to the rules above.

The ADARUN statement is fully described in the Operations documentation of Adabas.

Setting ADARUN Parameters

When specifying ADARUN parameters:

- Ensure that the correct program to be executed is specified (see the PROGRAM parameter);
- Ensure that the correct target ID is specified (see the TARGETID parameter); and
- Determine which settings for the following parameters are applicable for the session:
  - FORCE (overwrite active target ID)
  - SVC (Adabas SVC number)

Each ADARUN parameter has a default value that ADARUN uses if the parameter is not explicitly specified. Parameters can be abbreviated, but the abbreviation must be unique; that is, not the same as those of other ADARUN parameters.

The Entire Net-Work session statistics can be used to determine the best settings for each parameter. The statistics can be displayed using Entire Net-Work operator commands during the session; they are also printed automatically at the end of a session.

Statement Example

The following is an example of an Entire Net-Work ADARUN statement:

```
ADARUN PROG=NETWRK,TARGETID=3333,NAB=20,NC=50,LU=65535,SVC=251
```

For this node, Entire Net-Work:

- runs with a target ID of 3333;
- allocates an attached buffer pool for 20 interregion communication buffers;
- processes as many as 50 requests simultaneously;
- uses the SVC 251; and
- ensures adequate buffer size for Adabas compatibility by setting the value of the LU parameter to 65535. For more information about the LU parameter, read about it in the Adabas Operations documentation (see the *Adabas Operations Manual*).

ADARUN Parameter Summary

The following table summarizes some useful Entire Net-Work ADARUN parameters for Entire Net-Work. Details for each parameter can be found in the Adabas Operations documentation (see the *Adabas Operations Manual*).

Caution: If recommended numeric values are supplied in this table, do not specify smaller values; if recommended non-numeric values are supplied in this table, do not specify other values.

Parameter	Specifies	Minimum	Maximum	Recommended	Default
CT	The maximum number of seconds that can elapse from the time an Adabas command has been completed until the results are returned to the user through the interregion communication (operating-system-dependent).	1	16777215	---	60
FORCE	Whether the nucleus can overwrite an existing ID table entry.	---	---	---	NO
LU	The size of the intermediate user buffer area.	none	none	---	65535
NAB	The number of attached buffers to be used.	1	varies, depending on the amount of available virtual storage	---	16
NC	The maximum number of command queue elements.	20	32767	---	200
PROGRAM	The program to be run.	NETWRK	NETWRK	NETWRK	(none)
SVC	The Adabas SVC number to be used for the session.	---	---	---	249
TARGETID (see note below)	Entire Net-Work target ID.	1	65535	---	1

Note: The TARGETID parameter is an optional ADARUN parameter that specifies the unique Entire Net-Work target ID of this node. It is synonymous with the Adabas DBID parameter. All target IDs used by Entire Net-Work, Adabas (database IDs), Adabas TPF, Natural global bufferpools, etc., must be unique throughout all Entire Net-Work nodes. In particular, the Entire Net-Work target ID must not coincide with any database ID used in the network, with the exception of isolated databases that are defined for local availability only and are therefore unknown to Entire Net-Work.

4 Entire Net-Work Parameter Statements

This section describes the Entire Net-Work parameter statements that define the Entire Net-Work environment. It covers the following topics:

Overview

To communicate with other nodes, Entire Net-Work requires a definition of its own operating environment, access method information, and peer node characteristics. This is accomplished with the Entire Net-Work parameter statements. The NODE, DRIVER, and LINK statements are described in this section.

The NODE statement specifies the global operating parameters for a specific Entire Net-Work node. (Note that all interregion communication information is specified by ***ADARUN Control Statements***). Entire Net-Work control statements must always begin with the NODE statement, followed by a TCPX DRIVER statement, and the related LINK statement. DRIVER and LINK statements may be in any order, as long as no LINK statement precedes its related DRIVER statement.

Each link to an Entire Net-Work Clientmust be defined to Entire Net-Work with a LINK statement that specifies the operating parameters for the individual link. Each LINK statement must be associated with a previously-specified DRIVER statement.

Entire Net-Work NODE Statement

The NODE statement must be the first Entire Net-Work control statement. It defines the node's name and operating characteristics.

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

Statement Format

The following is the format of the NODE statement:

```
NODE  nodename   BUFFERS =(abufsize,ltbufsize,stbufsize,pfxbufsize),-
                     CQTIMER=secs,-
                     DEFINE=N/Y,-
                     DOMAIN=domain-name, -
                     DUMP =storage-area
                     ENDCMD=N/Y, -
                     LOG=N/Y-
                     LOGSIZE=logbufsize, -
                     MAXPATH=linkcount, -
                     MSGFORM =message-form, -
```

```
                    NIDO=N/Y, -
                    PASSWORD=password, -
                    REMCMD=N/Y, -
                    REPLYTIM=secs, -
                    START=Y/N,-
                    TIMER=secs, -
                    TRACE=trace-type[, trace-type] ...-
                    UCMSG=N/Y,-
                    ULINK=N/Y
```

NODE Statement Parameters

The NODE statement parameters, what they do, and their accepted values and defaults are as follows. Underlined arguments are default values that are in effect if the parameter is not specified.

```
nodename
```

A 1-8 character name for this node. Nodename must be specified immediately after NODE, separated by at least one blank space. All Entire Net-Work nodes must have unique names. Choose a meaningful name. Entire Net-Work uses the node name when referring to the node for operator messages and when logging.

```
BUFFERS=(abufsize,ltbufsize,stbufsize,{ pfxbufsize | 0 })
```

Specifies the amount of working storage reserved for internal buffer management. The values required depend on the number and types of line drivers and links, and on the message volume; these values can be adjusted according to usage statistics printed at the end of each Entire Net-Work session, or from issuing the operator command "D S" (DISPLAY STATISTICS).

Specify decimal values for the operands ranging from 0 to 2147483647 bytes; or, optionally, a value followed by either the multiplier "M" (x 1048576) or "K" (x 1024). If followed by "M", the specified value must range 0 to 2047; if followed by "K", the value must range 0 to 2097151. If a subparameter is omitted or zero is specified, no corresponding buffer pool will exist for this node. The requested storage space must be available; if the space is not available, Entire Net-Work ends with error message NET0013.

For z/OS systems, the buffer pool manager initializes the common buffer pool with a subpool of 256K. Additional subpools can be created dynamically, and all subpools can be expanded or contracted as needed. The only size limitation for the buffer pool is the size of the region or partition. The BUFFERS specification on the NODE statement remains the same, even though the first three values (abufsize, ltbufsize, and stbufsize) are ignored by the new buffer pool manager. The fourth value (pfxbufsize) is used to set the size of the Page-Fixed Buffer Pool.

```
abufsize (Asynchronous Buffer Pool Size)
```

Specifies the storage space to reserve for the Asynchronous Buffer Pool, used whenever asynchronously scheduled routines (exit routines, interrupt handlers) require storage. All requests to this pool are serialized. An abufsize value must be specified.

```
ltbufsize (Long-Term Buffer Pool Size)
```

Specifies the storage space to reserve for the Long-Term Buffer Pool, from which comparatively small buffers are allocated for relatively long-term use (for example, dynamically created control blocks). An ltbufsize value must be specified.

```
stbufsize (Short-Term Buffer Pool Size)
```

Specifies the storage space to reserve for the Short-Term Buffer Pool, from which comparatively large individual buffers are allocated for short time periods (for example, message and transmission buffers). An stbufsize value must be specified.

```
pfxbufsize (Page-Fixed Buffer Pool Size)
```

This optional subparameter specifies the bytes to reserve for the Page-Fixed Buffer Pool, from which all requests for page-fixed buffers are filled. The use of this pool is similar to that of the Short-Term Buffer Pool; that is, relatively large buffers are allocated for short time periods. If zero (the default) is specified, no Page-Fixed Buffer Pool is allocated..

Because the segment size of the Page-Fixed Buffer Pool is identical to the page size of the operating system, more page-fixed storage than short-term storage may be required.

```
CQTIMER={ secs | 60 }
```

This optional parameter specifies the approximate waiting time in seconds allowed for a user or application to retrieve command results with a router-16-call before timeout occurs. Specify a practical decimal value, depending on the node system's environment; Entire Net-Work accepts values ranging from 1 (one second) to 2147483647 (approximately 68 years-effectively, no timeout will occur). The default value is 60 (approximately one minute).

The purpose of the CQTIMER timeout is to prevent an Entire Net-Work Request Queue Element (RQE) and the attached buffer from becoming irretrievable if the user has ended abnormally. This parameter performs the same function as the ADARUN CT parameter.

Note: This parameter can be changed during Entire Net-Work operation by the SET CQTIMER command.

```
DEFINE={ Y | N }
```

This optional parameter determines whether the DEFINE operator command can be used to define links during Entire Net-Work operation. If Y is specified, the DEFINE operator command is accepted and executed. If N is specified, the DEFINE operator command is rejected. The default value is N.

```
DOMAIN=domain-name
```

This optional parameter allows you to subdivide the network into multiple domains. Using domains simplifies network management and limits administrative message traffic.

Specify a 1-6 character name. The default value is blank (no domain name).

```
DUMP={ ALL | NONE  | BLOCKS | TRACETAB | BUFFERS | LINKAREA | FORMAT }
```

Specifies the areas of storage to be printed after an abnormal termination of Entire Net-Work. The information is printed to the NETPRNT file if it is open. Otherwise, it is printed to the DDPRINT file. The DUMP parameter can be used to reduce the amount of output generated during an abend, especially on large Entire Net-Work systems. This parameter cannot be abbreviated.

In general, the default value of ALL should be used so that all diagnostic information is available to Software AG support.

Multiple values can be specified, separated by commas and surrounded by parentheses. For example:

```
DUMP=(BLOCKS, TRACETAB, FORMAT)
```

If values conflict, the last value specified is used. In the following, for example, the value NONE is used:

```
DUMP=(BLOCKS, TRACETAB, NONE)
```

Value	Description
ALL	All storage areas are dumped. This is the default value.
NONE	No storage areas are dumped.
BLOCKS	The major control blocks are dumped.
TRACETAB	The internal trace table is dumped.
BUFFERS	All internal buffer areas are dumped.
LINKAREA	All storage areas related to a driver and link are dumped.
FORMAT	The driver and link trace tables are formatted.

Note: This parameter can be changed during Entire Net-Work operation by the SET DUMP command.

```
ENDCMD={N | Y }
```

This optional parameter determines whether an operator command to end Entire Net-Work operation, e.g., NETEND, will be accepted when issued using the Programmable Command Interface. If Y is specified, the operator command is accepted and executed. If N is specified, the operator command is rejected. The default value is N.

```
LOG={ ON | OFF | YES | NO | FULL | SHORT }
```

This is a test parameter for recording control flow and for logging selected data areas. The information is written to the NETPRNT file if it is open. Otherwise, it is written to the DDPRINT file. OFF and NO are synonyms meaning that logging is not to be done during this Entire Net-Work node's session. ON, YES, and FULL cause logging of both the node's checkpoint records and data areas. SHORT causes logging of the checkpoint records only. The default value is NO.

Normally, logging should not be used because of the extra system resources required. The LOG function is intended primarily as a diagnostic tool; it is recommended that you use the LOG function only with the assistance of your Software AG technical support representative.

Note: This parameter can be changed during Entire Net-Work operation by the SET LOG command.

```
LOGSIZE=logbufsize
```

This optional parameter defines the size of a wrap-around log buffer which is used to hold the last records written to DDPRINT. The buffer can then be retrieved through the Programmable Command Interface. As many DDPRINT records as will fit are kept in the buffer. When new records are inserted, they replace the oldest records in the buffer. The valid range is 50 - 32000, which can also be specified in kilobytes, such as 30K. The suggested value is 10K. The default value is 0 (zero) -- no log is created. The value specified for this parameter must not be greater than 32000.

Note: This parameter takes effect only when the PASSWORD parameter is also specified.

```
MAXPATH={ linkcount | 4 }
```

The maximum path length, specified in links, that a message is expected to travel in the network. Specify a decimal value ranging from 1 to 32767. The default value is 4, resulting in a stack large enough for four node IDs.

The Communicator uses this optional value to build a list of two-byte entries for tracking each message. This list, called a Node Stack, is included in the message header. As the message passes through nodes en route to its target, each node's ID is added to the stack.

If the specified MAXPATH value results in a Node Stack that is larger than needed, messages will be unnecessarily long. If the MAXPATH value is too small, Entire Net-Work automatically copies the message, increasing the Node Stack size; this causes unnecessary processor overhead.

Note: This parameter can be changed during Entire Net-Work operation by the SET MAXPATH command.

```
MSGFORM={ message-format | 2 }
```

This optional parameter defines the message format of console messages and DDPRINT output. Value can be 1, 2, 3, or 4, as follows. The default value is 2.

Format	Description
1	is compatible with the message format used by Entire Net-Work Version 5.2. For example: NET0090 BUFFER USAGE STATISTICS
2	provides a severity letter (I, W, or E) with the message number. For example: NET0090I: BUFFER USAGE STATISTICS
3	provides the message number followed by the node name of the issuing Entire Net-Work node, padded with blanks to a length of 8. For example: NET0090I NODE2 : BUFFER USAGE STATISTICS
4	provides the message number followed by the node name, not padded, of the issuing Entire Net-Work node. For example: NET0090I NODE2: BUFFER USAGE STATISTICS

Notes:

1. 1. Individual line drivers may not recognize this parameter.
2. 2. Values between 5 and 255 will be accepted, but they have no meaning and are not valid.
3. 3. This parameter can be changed during Entire Net-Work operation by the SET MSGFORM= command.

```
NIDO={ N | Y }
```

This parameter can be used to force a node ID of 0 (zero) for all unsolicited connections. The default value is NIDO=N. If NID0=Y is specified, all nodes that attempt to connect and are not explicitly defined are assigned a node ID of 0. No Adabas servers on those nodes are broadcast through the network.

```
PASSWORD=password
```

This optional parameter is used to control access to the Programmable Command Interface (PCI). If a password value is specified, only PCI calls that supply a matching password are accepted. If

the special password ALL is specified, all PCI calls are accepted without password checking. If the PASSWORD parameter is omitted, or no password value is specified, all PCI calls are rejected. The default value is blank (no password).

Note: This parameter can be changed during Entire Net-Work operation by the SET PASSWORD command. For security reasons, this command is accepted only through the Programmable Command Interface.

```
REMCMD={ N | Y }
```

This optional parameter determines whether Programmable Command Interface (PCI) calls that originate at remote Entire Net-Work nodes will be accepted. If N is specified, only calls from local applications, i.e., calls that reach Entire Net-Work through the local Adabas Router, are allowed. If Y is specified, calls that reach the local node as messages from other nodes are also accepted. The default value is N.

Notes:

1. 1. This parameter can be changed during Entire Net-Work operation by the SET REMCMD command.
2. 2. This parameter takes effect only when the PASSWORD parameter is specified.

```
REPLYTIM={ secs | 60 }
```

This optional parameter specifies the approximate waiting time, in seconds, allowed for a user request to complete before timeout occurs. A request is considered complete when the originating node receives a reply.

Specify a practical decimal value, depending on the node system's operation; Entire Net-Work accepts values ranging from 1 (one second) to 2147483647 (approximately 68 years; effectively, no timeout will occur). The default value is 60 (approximately one minute).

In the event that a message is "stranded" (that is, a reply cannot be returned to the originating node), REPLYTIM specifies a time after which a response code 224 is returned to the user.

Note: This parameter can be changed during Entire Net-Work operation by the SET REPLYTIM command.

```
START={ Y | N }
```

This optional parameter is used to determine whether Entire Net-Work starts normal operations automatically. The default value is Y.

- If Y is specified, Entire Net-Work automatically starts all line drivers and initiates connections for all links that have the parameter ACQUIRE=Y specified.
- If N is specified, Entire Net-Work initializes line drivers but does not start them, nor does it connect any links. Line drivers can be started individually by using the START operator command.

```
TIMER={ n | 30 }
```

This optional parameter defines the interval between handling of time-dependent requests; that is, every *n* seconds, Entire Net-Work scans its tables for any time-dependent action that needs to be taken. The TIMER value determines the precision of all time-dependent Entire Net-Work services.

Specify a practical decimal value depending on the node operation. On all operating systems, except GCS, Entire Net-Work accepts values ranging from 1 to 16777215 seconds (effectively, no timing supervision will occur, even if other timing parameters, such as REPLYTIM, CQTIMER, or ADARUN CT are set). On GCS systems, Entire Net-Work accepts values ranging from 1 to 86399 seconds.

There is an interaction between TIMER and other timing parameters. If the TIMER interval is greater than the individual CQTIMER and REPLYTIM intervals, the specified action may not be started until the TIMER interval has expired. The default value is 30 seconds.

```
{TRACE | TROFF | TRON}=tracetype[, tracetype]...
```

Specifies trace control parameters for performing program traces. Tracing should not be active during normal operation. Tracing is intended as a diagnostic tool; it is recommended that you use tracing only with the assistance of your Software AG technical support representative.

All diagnostic information from tracing is written to the NETPRNT file if it is open. Otherwise, it is written to the DDPRINT file.

The standard values for *trace-type* are shown in the following table.

Note: If you want to trace an Entire Net-Work line driver and you have specified a non-default driver name for that line driver (using the DRVNAME parameter), you must use the driver name you defined in the DRVNAME parameter as the *trace-type* in the NODE statement's TRACE parameter.

Default trace type	Performs
ADA	Limited Adabas call tracing
BPM	Buffer Pool Manager tracing
MAIN	Mainline tracing
RQM	Receive Queue Manager tracing
TQM	Transmission Queue Manager tracing

TRACE=ADA allows you to trace Adabas calls without using Entire Net-Work full tracing. That is, logging can be turned off (see the LOG parameter), thus reducing the amount of overhead required.

Line driver traces can be requested for installed line drivers on the local node only. TRACE and TRON are synonyms to either start or resume tracing of the specified events. TROFF stops tracing. If this parameter is not specified, no tracing will occur.

Note: Values set by this parameter can be changed during Entire Net-Work operation by the SET TRACE, TROFF, or TRON commands.

```
UCMSG={ N | Y }
```

This optional parameter determines whether messages are issued in uppercase (Y) or mixed case (N). The default value is N.

Notes:

1. 1. Individual line drivers may not recognize this parameter.
2. 2. This parameter can be changed during Entire Net-Work operation by the SET UCMSG command.

```
ULINK={ N | Y }
```

This optional parameter determines whether multiple links are allowed between two Entire Net-Work nodes. If Y is specified, Entire Net-Work ensures that each connection to an adjacent node is unique; incoming connection requests from adjacent nodes that are already known as active will be rejected. If N is specified, multiple links between two Entire Net-Work nodes are allowed. The default value is N.

In networks with many PCs, two PCs may be assigned the same node name and ID by mistake. If both PCs are simultaneously connected to Entire Net-Work, they are perceived as one Entire Net-Work node that is connected by two different links. As a result, one of the PCs may receive a reply to a call that originated on the other PC.

To avoid this type of situation, specify ULINK=YES. When the second PC tries to connect, it is rejected. The integrity of the network is maintained and the duplicate node name and ID can be identified. This parameter can be changed during Entire Net-Work operation by the SET ULINK command.

Entire Net-Work DRIVER Statement

The Entire Net-Work DRIVER control statement defines the TCPX line driver to be loaded. The following is the syntax of the DRIVER statement:

```
DRIVER     drivername     parameter=value,...
```

where *drivername* is "TCPX". This invokes module NETTCPX, which supports the TCPX access method.

The line driver itself must exist in the libraries defined for the related job step, and must have the name:

```
NET drivername
```

where *drivername* is the same name specified in the DRIVER statement's *drivername* field.

The keyword parameter or parameters:

```
parameter=value,...
```

are specific to the line driver. For more information, read *TCPX DRIVER Statement* .

Entire Net-Work LINK Statement

Each link to another node must be defined with a LINK statement. Each link uses a specific communications access method, as defined by a related DRIVER statement. LINK statements must specify the related driver by name, and follow the related DRIVER statement in the Entire Net-Work statement order.

The following is the syntax of a LINK statement:

```
LINK     linkname     drivername     parameter=value,...
```

where *linkname* is a unique one- to eight-character name identifying the link, and drivername is "TCPX". A DRIVER statement for "drivername" must precede this LINK statement.

Note: If more than 8 characters are entered for *linkname*, only the first 8 characters are used. The connection is issued correctly and no error message is generated.

The keyword parameters:

```
parameter=value,...
```

are specific to the line driver. For more information, read *TCPX LINK Statement*.

5 Entire Net-Work Operator Commands

This section describes the Entire Net-Work operator commands. It covers the following topics:

Overview

Although Entire Net-Work operates automatically, there are operator commands available to display or modify the status of the network and control the local Entire Net-Work node.

The Entire Net-Work commands described in this section are similar to Adabas operator commands. A summary and description of the operator commands for z/OS are provided.

The Entire Net-WorkTCPX line driver has the ability to process operator commands that are directed to a specific link or directly to the driver. For more information about TCPX line driver operator commands, read *TCPX Operator Commands*.

Operator Commands Summary

The following table summarizes the Entire Net-Work operator commands for z/OS:

Command	Argument	Action
ADAEND		Terminate Entire Net-Work session.
CLOSE	driver	Disconnect all links of a driver, then close the driver.
CLOSE NETPRNT		Close the NETPRNT file and route all trace and snap output to DDPRINT.
CONNECT	link	(Re-) connect a link after a disconnect or handshake error.
DEFINE	link	Dynamically define a new link.
DISABLE	link	Disable a link (link cannot accept connects).
DISCONNECT	link	Disconnect a link.
DISPLAY	parameter	Display link, nodes, targets, paths or statistics.
DUMP		Snap data areas, then terminate the Entire Net-Work session.
ENABLE	link	Enable a link (the link can accept connects).
END		Terminate Entire Net-Work session.
FORCE	node	Broadcast a "node down" message.
HALT		Terminate Entire Net-Work session.
HELP		List available operator commands.
NETEND		Terminate Entire Net-Work session.
OPEN	driver	Reopen a driver after a close or access method failure.
OPEN NETPRNT		Open the NETPRNT file and route all trace and snap output to the NETPRNT file.

Command	Argument	Action
PROBE	node	Send a probe message to a node.
RESUME	link	Resume sending messages via this link.
SET	parameter	Change the values of Entire Net-Work parameters. Note: The minimum abbreviation for SET is the null string (zero characters long).
SNAP		Snap data areas to DDPRINT.
START	driver	(Re-) start driver (then connect all links of driver).
STOP		Terminate Entire Net-Work session.
SUSPEND	link	Stop sending messages on this link.
TERMINATE		Terminate Entire Net-Work session.

Operator Command Descriptions

This section describes each of the operator commands in detail. The underlined portion of the command is the minimum abbreviation.

```
ADAEND
END
HALT
NETEND
STOP
TERMINATE
```

Any one of the above commands can be used to terminate an Entire Net-Work session normally. The STOP operator command (for example, STOP taskid or P taskid) can be used in z/OS environments.

A check is made for any additional parameters. If one is found, the command is rejected and message NET0115 is issued. Thus, erroneous commands are rejected and an accidental termination of Entire Net-Work is avoided.

Once the termination command has been accepted by Entire Net-Work, no more requests are selected from the request queue. Message NET0999 is displayed on the operator console confirming that normal termination procedures have been started.

Note: The DUMP command also ends Entire Net-Work operation after performing a snap dump of pertinent data areas.

```
CLOSE   driver
```

Terminate all activities of the driver by disconnecting and closing all links related to the driver, then closing the driver itself.

The effect of this command can be reversed by issuing the OPEN or START command for the driver, and CONNECT commands for the links (as appropriate).

```
CLOSE   NETPRNT
```

Close the NETPRNT file and route all trace and snap output to the DDPRINT file. When the NETPRNT file is closed, the data set can be copied for sending to Software AG support, without shutting down Entire Net-Work. The file must be allocated SHR. This command cannot be abbreviated.

```
CONNECT   linkname
```

Attempt to connect link linkname. The link name specified must match that used on the LINK statement. If the link was disconnected after a "handshaking" conflict, the CONNECT command can be used to retry the procedure. If the link is disabled, the CONNECT command can be used to enable it.

```
DEFINE   LINK   linkname={ link statement | LIKE linkname }
```

Define a link during Entire Net-Work operation. The link statement must adhere to the format described in the section about the related line driver.

The LIKE linkname parameter can be used instead of the link statement to define a link by copying the parameters specified for a previously-defined link. For example:

```
DEFINE LINK TOPSYS LIKE BOTSYS
```

Note: DEFINE LINK, is permitted only if DEFINE=Y is specified on the NODE statement

```
DISABLE  linkname
```

Instructs the specified link not to accept any connections from other Entire Net-Work nodes. If the link is connected, it is disconnected and then disabled.

```
DISCONNECT   linkname
```

Disconnect the specified link, which is connected to this node. The link name specified must be the same as that used on the LINK statement.

```
DISPLAY  { ALINKS | CSCI | LINKS | NODES | PATHS | STATS | TARGETS | ZAPS }
                 [ {name | string* } ]
```

Displays current information about the specified network component. Only one component type (link, node, path, or target) can be specified in a single DISPLAY command. The information is

displayed in the form of Entire Net-Work messages. For more information, see the section *Messages and Codes* in the *Entire Net-Work Messages and Codes*.

The optional second parameter serves to qualify the display request, thereby limiting the information displayed. At the same time, additional information is displayed for qualified DISPLAY LINK or DISPLAY NODES requests.

The possible qualifier values and their meanings depend on the type of request. A link name, node name, or (numeric) target ID may be specified. Alternatively, a string ending in a "wild card" character (*) may be used to indicate all links or nodes whose names start with the specified string. The asterisk (*) alone may be used to produce a display of all links or nodes, but additional information is shown only for qualified display requests.

The following is an example of DISPLAY CSCI output:

```
F NETWK,D C
NETQ002I:  Csi Server  -ESG111- Act Targ(00039) Srv(00013)
NETQ002I:  Csi Server  ESQSRV   Act Targ(00039) Srv(00012)
NETQ002I:  Csi Server  TESTNAT  Act Targ(01001) Srv(00011)
NETQ002I:  Csi Server  KSPS2    Act Targ(01001) Srv(00010)
NETQ002I:  Csi Server  KSPS1    Act Targ(01001) Srv(00009)
NETQ002I:  Csi Server  -DAEKCO- Act Targ(01014) Srv(00008)
NETQ002I:  Csi Server  KCOSRV4  Act Targ(01014) Srv(00007)
NETQ004I:  Registered Servers Display Function Complete
```

The following is an example of DISPLAY NODES output:

```
F NETWK,D N
NET0122I: NODE FNODE     (50752) LOCAL
NET0122I: NODE ALSNODE   (54080) DIST 000040 VIA LINK LNKE
NET0122I: NODE ANODE     (49472) DIST 000020 VIA LINK LNKA
NET0122I: NODE ENODE     (50496) DIST 000020 VIA LINK LNKE
```

A qualifier is used in the following example:

```
F NETWK,D N A*
NET0122I: NODE ALSNODE   (54080) DIST 000040 VIA LINK LNKE
NET0123I: TARGETS: 00025 00171 00194 00175 00173 00018 00009
NET0123I: TARGETS: 00177
NET0122I: NODE ANODE     (49472) DIST 000020 VIA LINK LNKA
NET0123I: TARGETS: 00125 00192
```

The following is an example of DISPLAY TARGETS output:

```
F NETWK,D T
NET0124I: TARGET 09777 (C-N) ACTIVE ON NODE ALSNODE
NET0124I: TARGET 00009 (I-T) ACTIVE ON NODE ALSNODE
NET0124I: TARGET 02048 (L-N) ACTIVE ON NODE ANODE
NET0124I: TARGET 00237 (I-N) ACTIVE ON NODE ANODE
NET0124I: TARGET 00238 (I-N) ACTIVE ON NODE ANODE
NET0124I: TARGET 09888 (C-N) ACTIVE ON NODE ANODE
NET0124I: TARGET 00234 (I-N) ACTIVE ON NODE ANODE
NET0124I: TARGET 55769 (C-N) ACTIVE ON NODE ANODE
```

The following is an example of DISPLAY PATHS output:

```
F NETWK,D P
NET0122I: NODE ALSNODE  (54080) DIST 000080 (001) VIA LINK LNKA
NET0122I: NODE ALSNODE  (54080) DIST 000040 (002) VIA LINK LNKE
NET0122I: NODE ANODE    (49472) DIST 000020 (001) VIA LINK LNKA
NET0122I: NODE ANODE    (49472) DIST 000040 (002) VIA LINK LNKE
NET0122I: NODE ENODE    (50496) DIST 000040 (002) VIA LINK LNKA
NET0122I: NODE ENODE    (50496) DIST 000020 (001) VIA LINK LNKE
```

The DISPLAY STATS command produces the same type of information found in the statistics displayed at the end of an Entire Net-Work session. A qualifier parameter, if given, would have no effect. The buffer usage statistics displayed depend on the operating system being used.

The following is an example of DISPLAY STATS output for a GCS/CMS system:

```
F NETWK,D STATS
NET0090I: BUFFER USAGE STATISTICS:
NET0091I: ASYNCH. BUFFERS:     000016 (= 24.2 %) OF 000064 K USED
NET0091I: LONG TERM BUFFERS:   000000 (=  0.4 %) OF 000064 K USED
NET0091I: SHORT TERM BUFFERS:  000000 (=  6.1 %) OF 001025 K USED
NET0091I: ATTACHED BUFFERS:    000000 (= 11.9 %) OF 000080 K USED
NET0091I: REQUEST QUEUE:       000000 (=  6.0 %) OF 000050 RQES USED
NET0087I: 0000010847 REQUESTS FROM LOCAL RQ
```

The following is an example of DISPLAY STATS output for a z/OS system. It includes a NETB001I and a NETB009I for each active buffer pool, a set of NETB008I, NETB010I, and NETB012I for each subpool within the buffer pools, and a NETB013I for each operator command issued.

```
F NETWK,D STATS
NETB000I: ------------------------------------------------------------
NETB001I:        Statistics For Buffer Pool COMN Loc = ANY
NETB000I: ------------------------------------------------------------
NETB008I: Req =(      13,     0,    10,     0)
NETB010I: ELM =(     512,   512,   512,   512), Sz = 512 B
NETB011I: Str =(     256,   256,   254,   252 ) K
NETB012I: Exp =(       0,     1,     0,     0)
NETB000I: ------------------------------------------------------------
NETB008I: Req =(       1,     0,     0,     0)
NETB010I: Elm =(      10,    10,    10,     2), Sz =   1 K
NETB011I: Str =(      15,    15,    13,    13) K
NETB012I: Exp =(       0,     1,     0,     0)
NETB000I: ------------------------------------------------------------
NETB008I: Req =(       1,     0,     0,     0)
NETB010I: Elm =(       1,     1,     1,     1), Sz =  14K
NETB011I: Str =(      14,    14,     0,     0) K
NETB012I: Exp =(       0,     1,     0,     0)
NETB000I: ------------------------------------------------------------
NETB009I: High Allc=   285  Curr Allc =   285   Curr Avail =   267 K
NETB000I: ------------------------------------------------------------
NETB001I:       Statistics For Buffer Pool PGFX Loc = ANY
NETB000I: ------------------------------------------------------------
NETB008I: Req =(       0,      0,      0,    0)
NETB010I: Elm =(      64,     64,     64,    0), Sz =  4K
NETB011I: Str =(     256,    256,    256,  256) K
NETB012I: Exp =(       0,      0,      0,    0)
NETB000I: ------------------------------------------------------------
NETB009I: High Allc=  256  Curr Allc =   256   Curr Avail =   256 K
NETB000I: ------------------------------------------------------------
NETB013I: Combined Buffer Pools Size                    541 K
NETB000I: ------------------------------------------------------------
```

The DISPLAY ZAPS command lists, for each Entire Net-Work module, its name, assembly date, system maintenance level, and zap level. If zaps were applied after initial shipment, their numbers are listed as "Additional Zaps". The following is an excerpt from a DISPLAY ZAPS example:

```
F NETWK,D Z
NET0037I: NETCLF (1998/11/10 SM=0001) ZAP LEVEL 0000
```

```
DUMP
```

Issue a snap dump, then end the Entire Net-Work session. DUMP is equivalent to the SNAP command followed by an ADAEND (or synonymous) command.

```
ENABLE  linkname
```

Revokes a previously entered DISABLE command. The specified link is instructed to accept incoming connect requests. Enabling a disconnected link does not connect the link.

```
FORCE  { nodename | nodeid }
```

Broadcasts a control message through the network to notify all Entire Net-Work nodes that the specified node is no longer available. This command is provided for diagnosis and exception handling, and should be used only on the advice of your Software AG technical support representative.

```
HELP
```

Lists the available Entire Net-Work operator commands with a short explanation of their function.

```
OPEN  driver
```

Reopen an installed/defined line driver that was stopped due to an access method or other network or system failure, or by the CLOSE operator command. The driver name must be the same as that specified on a DRIVER statement. Note that this command is currently a synonym for the START command. For further information, see the explanation of the START command.

```
OPEN   NETPRNT
```

Open the NETPRNT file and route all trace and snap output to NETPRNT. This command is necessary only after a CLOSE NETPRNT command has been used. It opens the NETPRNT file when Entire Net-Work is initialized. If the file is allocated SHR or OLD it will be erased when opened. This command cannot be abbreviated.

```
PROBE  { nodename | nodeid } [ nnnn ]
```

The PROBE command verifies that the specified node is available and can be reached. Entire Net-Work issues internal probe commands for the same purpose during normal operation. PROBE routes an internal message to the specified node and back. If the node cannot be reached, this information is sent to all active nodes, updating the node status.

The optional second parameter specifies that nnnn bytes of random user data (64512 bytes maximum) are to be appended to the actual probe message. The exact length of the message sent can be calculated as follows:

```
70 + (nodestack size) + nnnn
```

where nodestack size is twice the number specified by the NODE statement parameter MAXPATH=, rounded up to the next multiple of 4. For example, with MAXPATH=4 (the default value) the following command results in a message of length 1078:

```
PROBE nodename 1000
```

The result of the operation is displayed on the operator console, as shown in the following example:

```
F NET1,PROBE TWO
NET0136I: PROBE MESSAGE SENT
NET0135I: PROBE FOR NODE TWO      (0001.711 SEC)
NET0120I: NODE TWO      (62194) DIST 000030 VIA LINK TOFIVE
NET0140I: VERSION v.r.s (1999/11/10)
```

```
RESUME  linkname
```

Revokes a SUSPEND command for the specified link. The link's status changes to "active" and the link resumes sending queued messages.

```
SET  parameter=value, [ ... ]
```

The SET command can be used to change Entire Net-Work parameter settings dynamically without interrupting network operations. The SET command itself may be omitted.

Multiple parameters can be specified with one SET command.

For example, the following:

```
F NODEA,SET  CQTIMER=180, TRACE=OFF
```

is equivalent to

```
F NODEA,CQTIMER=180, TRACE=OFF
```

The parameters allowed for the SET command are a subset of those defined on the Entire Net-Work NODE statement. They are:

```
SET CQTIMER=secs
```

The approximate waiting time allowed for a user or application to retrieve command results with a router-16-call before timeout occurs. For more information, see the CQTIMER parameter.

```
SET DUMP={ ALL | NONE | BLOCKS | TRACETAB | BUFFERS | LINKAREA | FORMAT }
```

Sets the storage areas to be included in a dump when Entire Net-Work terminates abnormally. The information is printed to the NETPRNT file if it is open. Otherwise, it is printed to the DDPRINT

file. SET DUMP can be used to reduce the amount of output generated during an abend, especially on large Entire Net-Work systems. This command cannot be abbreviated.

In general, the default value of ALL should be used so that all diagnostic information is available to Software AG support.

Multiple values can be specified, separated by commas and surrounded by parentheses. For example:

```
SET DUMP=(BLOCKS, TRACETAB, FORMAT)
```

If conflicting values are specified, the last value specified is used. In the following, for example, the value used is NONE:

```
SET DUMP=(BLOCKS, TRACETAB, NONE)
```

Value	Description
ALL	All storage areas are dumped. This is the default value.
NONE	No storage areas are dumped.
BLOCKS	The major control blocks are dumped.
TRACETAB	The internal trace table is dumped.
BUFFERS	All internal buffer areas are dumped.
LINKAREA	All storage areas related to a driver and link are dumped.
FORMAT	The driver and link trace tables are formatted.

```
SET LOG={ ON | OFF | YES | NO | FULL | SHORT }
```

Regulates control flow and logging of selected data areas to the printer data set. For more information, see the LOG parameter.

```
SET _MAXPATH=linkcount
```

Sets the maximum path link, specified in number of links, that a message from users on this node is expected to travel. For more information, see the MAXPATH parameter.

```
SET  MSGFORM=message-format
```

Sets the message format of console messages and DDPRINT output. For more information, see the MSGFORM parameter.

```
SET  PASSWORD=password
```

Sets the password that controls access to the Programmable Command Interface (PCI). For more information, see the PASSWORD parameter.

Note: For security reasons, this command is accepted only through the Programmable Command Interface.

```
SET  REMCMD={ N | Y }
```

Allow or disallow remote access to the Programmable Command Interface. For more information, see the REMCMD parameter.

```
SET  REPLYTIM=secs
```

Sets the time, in seconds, that this node is to wait for a reply to a user request before timing out. For more information, see the REPLYTIM parameter.

```
SET  { TRACE | TROFF | TRON }={ trace | (trace,...) }
```

Sets the trace control parameters for performing program traces. For more information, see the TRACE parameters.

```
SET  UCMSC={ N | Y }
```

Controls whether messages are issued in upper case or mixed case. For more information, see the UCMSG parameter.

```
SET  ULINK={ N | Y }
```

Allows or disallows multiple links to an adjacent Entire Net-Work node. For more information, see the ULINK parameter.

```
SNAP  { BPH | CQ | CURRMSG |  MAIN  | MYBLK |  TRACE |  UBQ }
```

Issue a snap dump of selected data areas to the DDPRINT file and continue processing. (Under certain circumstances, a snap dump is performed internally at either normal or abnormal session end.)

The optional parameters are used to snap one or more specific data areas:

Parameter	Area
BPH	Buffer pool headers.
CQ	Command queue.
CURRMSG	Message that Entire Net-Work mainline is currently working on.
MAIN	Header of mainline module.
MYBLK	Central control block.
TRACE	Internal trace table.
UBQ	UB-Queue (currently active Adabas commands).

```
START  drivername
```

Restart an installed line driver that was stopped due to an access method or other network or system failure, or by the CLOSE operator command. The driver name must be the same as that specified on a DRIVER statement. The START command is a synonym for the OPEN command.

```
SUSPEND  linkname
```

Instructs the specified link to not send any more messages. However, Entire Net-Work can still queue messages on this link. The SUSPEND command is valid only if the link is active.

6 Diagnostic Utilities

This section describes the Entire Net-Work diagnostic utilities, which are provided in the Entire Net-Work load library:

The Adabas modules required by the utilities are provided in the WAL*vrs* library distributed with Entire Net-Work. See the section *Product Code Description* in the *Entire Net-Work Release Notes* and the *Report of Tape Creation*.

NETPFIL1 Utility

The NETPFIL1 utility is used to select the information to be printed from the Entire Net-Work NETPRNT file. The Entire Net-Work NETPRNT file contains tracing, logging, and dump output. The NETPRNT file should be used as input to the NETPFIL1 utility for the NETFILE file.

NETPFIL1 can be used to select

- only those records that fall within a certain date and time frame,
- only logging and dump records that match the title or any desired portion of the title,
- only trace records that match the title or any desired portion of the title, or
- any combination of the above.

The output from NETPFIL1 can be used as input to the ***NETPFIL2 Utility*** to perform additional filtering.

NETPFIL1 Parameters

This section describes the parameters for the NETPFIL1 utility.

```
DATE=yyyymmdd
```

Only records created on the specified date are written to DDPRINT. The default value is all dates.

```
STARTTIME=hhmmssth
```

Records created before the specified time are not written to DDPRINT and are not checked for other selection criteria. The default is no restriction on the start time.

```
ENDTIME=hhmmssth
```

Records created after the specified time are not written to DDPRINT and are not checked for other selection criteria. The default is no restriction on the end time.

```
LOG=log-title
```

Log and dump records with a title that matches the specified log title are written to DDPRINT.

The log title must be specified exactly as it appears in the NETPRNT file; the value specified can be limited to the number of characters necessary to select the desired records. All characters following the last nonblank character are considered wild cards. Spaces are valid in the log title, but the check is performed only up to the last nonblank character.

LOG= selects all log and dump records. The default value is that no log or dump records are written.

```
TRACE=trace-title
```

Trace records that have a title matching the specified trace title are written to DDPRINT.

The trace title must be specified exactly as it appears in the NETPRNT file; the value specified can be limited to the number of characters necessary to select the desired records. All characters following the last nonblank character are considered wild cards. Spaces are valid in the trace title, but the check is performed only up to the last nonblank character.

TRACE= selects all trace records. The default value is that no trace records are written.

Anything else is written to the output file as a comment along with the parameters specified to the beginning of the DDPRINT file.

An asterisk (*) in column 1 causes this parameter record to be ignored.

Examples

The following selects all records for a five minute period with a comment:

```
ALL INFORMATION FROM 2:23 to 2:29
STARTTIME=14230000
ENDTIME=14290000
LOG=
TRACE=
```

The following selects log records for IDDDATA and IDDBLK using implied wild cards:

```
LOG=IDD
```

The following selects log records for IDDDATA only:

```
LOG=IDDDATA
```

The following selects dump records for M Y B L K, D R I V B L K, and L I N K B L K:

```
LOG=M Y B L K
LOG=D R I V B L K
LOG=L I N K B L K
```

NETPFIL2 Utility

The NETPFIL2 utility is used to search for a control block or storage area snapped by either the logging function or a dump. When the desired dump entries are found, they are printed.

Optionally, NETPFIL2 can also print all non log entries, including the trace and time stamp records. NETPFIL2 is therefore suitable as a second step in a job where NETPFIL1 is run first.

To summarize, NETPFIL2 can be used to

- find and print a control block or data area that contains a specified value at a specified offset,
- optionally print all non logging and dump records, and
- find multiple control blocks and multiple values.

The Entire Net-Work NETPRNT file contains tracing, logging, and dump output. The NETPRNT file should be used as input to the NETPFIL2 utility for the NETFILE file.

Parameters

This section describes the parameters for the NETPFIL2 utility.

```
NONLOG={Yes | No}
```

The NONLOG parameter specifies whether all non log and dump records should be written to DDPRINT.

NONLOG=Yes	Writes all non log and dump records.
NONLOG=No	Ignores all non log and dump records. This is the default value.

```
FIND=log-title (OFFSET=xxx {VALUE=characters | VALUE=X`hex values'})
```

Log and dump records that have a log title matching the specified log title are evaluated to determine whether the hexadecimal offset matches either the character or hexadecimal value specified. If it matches, the whole dumped area is written to DDPRINT.

The log title must be specified exactly as it appears in the NETPRNT file; the value specified can be limited to the number of characters necessary to select the desired records. All characters fol-

lowing the last non blank character are considered wild cards. Spaces are valid in the log title, but the evaluation is performed only up to the last non blank character. To select all log and dump records, use the LOG= parameter.

OFFSET=xxx must be specified as a valid hexadecimal number. This number specifies the first byte of the location in the dumped area to compare. The comparison is performed in such a way that the value must start at this location. Each additional character of the value is checked at the next logical position in the dump, even if the next logical position is on the next record.

VALUE=characters specifies the right hand portion of the value, which is in character format. The character form of the value is compared to the interpreted part of the dumped records.

VALUE=X`hex values' specifies the left hand portion of the value, following the address and offset. The hexadecimal form of the value is compared to the hexadecimal part of the dumped records.

Example JCL

Running in z/OS Environments

The following example JCL can be used to run NETPFIL1 or NETPFIL2 in a z/OS environment.

```
//YOURJOB  JOB (0),'NET-WORK',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//FILTER1  EXEC PGM=NETPFIL1                          <--- see Note 1 //STEPLIB
 DD  DSN=NETWRK.vrs.LOAD,DISP=SHR                <--- see Note 2
//         DD  DSN=WAL.vrs.LOAD,DISP=SHR                <--- see Note 3
//NETFILE  DD  DSN=NETWRK.vrs.NETPRNT,DISP=SHR             <--- see Note 4
//DDCARD   DD  *                                          <--- see Note 5
  THIS COMMENT WILL BE PRINTED AT THE BEGINNING OF THE OUTPUT
  DATE=20000316
  STARTTIME=12570000
  ENDTIME=12580000
  LOG=T R A C E
* LOG=IDDDATA
  TRACE=
* TRACE=SENDOUT
//DDPRINT  DD  SYSOUT=*,DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
/*
```

Notes:

1. 1. Specify the utility program you want to run.
2. 2. This is the same load library you use to run Entire Net-Work.
3. 3. This is the same Adabas or WAL load library you use to run Entire Net-Work.

4. 4. This is the file that was created by Entire Net-Work with DDNAME NETPRNT.
5. 5. This is the parameter file and may be any sequential file with LRECL=80.

Index