Entire Net-Work

Reference

Version 6.5.2

April 2023

ADABAS & NATURAL

This document applies to Entire Net-Work Version 6.5.2 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: WCPMF-REF-652-20230324

Table of Contents

Statement and Command Reference

ADARUN Parameters	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.

1 Conventions

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.
vertical bars (|)	Vertical bars are used to separate mutually exclusive choices. **Note:** In more complex syntax involving the use of large brackets or braces,	**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.

Convention	Description	Example
	mutually exclusive choices are stacked instead.	
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

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.cloud**. Navigate to the desired product and then, depending on your solution, go to "Developer Center", "User Center" or "Documentation".

Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.softwareag.com**.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

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 ADARUN Parameters

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.

ADARUN parameters are fully described in the *Operations* section of your Adabas documentation.

Specifying 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.

The syntax of ADARUN parameters is:

ADARUN *parameter* = *value* **[**, *parameter* = *value* **]...**

ADARUN parameters 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* string; and
- Not extend beyond position 72 of a line.

ADARUN parameters may contain multiple lines. Each line must be specified as a separate parameter according to the rules above.

ADARUN parameters are fully described in your Adabas operations documentation.

CMADDR and CMLADDR Parameters: GETMAIN Memory Pool Start Address

These parameters apply only to BS2000 environments running Sockets versions less than 2.2.

Parameter	Specify . . .	Minimum	Maximum	Default
	the start address of the GETMAIN memory pool			
CMADDR	above the 16MB line	0, or X'1000000'	X'FE000000'	0
CMLADDR	below the 16MB line	0, or X'10000'	X'E00000'	0

Note: These parameters should only be used if required by an installation site, in which case the site will provide the needed value.

These parameters are involved with setting all required memory into a common memory pool for use by associated tasks on BS2000 above (CMADDR) or below (CMLADDR) the 16 megabyte line. They set the start address of this memory pool.

If the default value ("0") is used for a CMADDR or CMLADDR parameter, the start address depends on the value of the corresponding CMSIZE or CMLSIZE parameter:

- If the CMADDR or CMLADDR parameter value is "0" and the corresponding CMSIZE or CMLSIZE parameter is *not* "0", the operating system will set the start address at the next available megabyte boundary.
- If the CMADDR or CMLADDR parameter value is "0" and the corresponding CMSIZE or CMLSIZE parameter is *also* "0", required memory will not be obtained in a common memory pool but rather in class 6 memory.

Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the value of the CMADDR parameter to `X'01200000'`.

Examples

The following example allows for three (3) megabytes of user storage in the address range X'2000000' to X'4FFFFFF':

```
ADARUN PROG=ADANUC,CMADDR=2000000,CMSIZE=2500000
```

The following example allows for three (3) megabytes of user storage below 16 megabytes in the address range X'200000 to X'4FFFFF:

```
ADARUN PROG=ADANUC,CMLADDR=200000,CMLSIZE=25000000
```

CMFIX or CMLFIX Parameters: GETMAIN Memory Pool Fixed Location

These parameters apply only to BS2000 environments running Sockets versions less than 2.2.

Parameter	Specify . . .	Possible Values	Default
	whether the GETMAIN memory pool must have a fixed location or not		
CMFIX	above the 16MB line	YES | NO	NO
CMLFIX	below the 16MB line	YES | NO	NO

Note: Use these parameters only if required by an installation site, in which case the site will provide the needed value.

The ..FIX parameter is ignored if the corresponding ..SIZE parameter value is 0.

These parameters are involved with setting all required memory into a common memory pool for use by associated tasks on BS2000 above (CMFIX) or below (CMLFIX) the 16 megabyte line. This parameter specifies whether the memory pool is at a fixed location or not:

Value	The application's GETMAIN memory pool . . .
NO	need not be located at the same start address in the subtask as in the main task.
YES	must be located at the same start address in the subtask as in the main task.

Note: A subtask depends on the application. An example of this is Adabas Review in local mode.

Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the values of both the CMFIX and CMLFIX parameters to "YES".

Examples

The following example allows for three (3) megabytes of user storage above 16 megabytes in the address range X'2000000' to X'4FFFFFF'. This address range will also be used by the subtask:

```
ADARUN PROG=ADANUC,CMADDR=2000000,CMSIZE=2500000,CMFIX=YES
```

The following example allows for three (3) megabytes of user storage below 16 megabytes in the address range X'200000' to X'4FFFFF'. This address range will also be used by the subtask:

```
ADARUN PROG=ADANUC,CMLADDR=200000,CMSIZE=2500000,CMFIX=YES
```

CMSCOPE or CMLSCOPE Parameters: GETMAIN Memory Pool Scope

These parameters apply only to BS2000 environments running Sockets versions less than 2.2.

Parameter	Specify . . .	Possible Values	Default
	access to the GETMAIN memory pool		
CMSCOPE	above the 16MB line	GROUP | GLOBAL	GROUP
CMLSCOPE	below the 16MB line	GROUP | GLOBAL	GROUP

Note: These parameters should only be used if required by an installation site, in which case the site will provide the needed value.

The ..SCOPE parameter is ignored if the corresponding ..SIZE parameter value is 0.

These parameters are involved with setting all required memory into a common memory pool for use by associated tasks on BS2000 above (CMSCOPE) or below (CMLSCOPE) the 16 megabyte line. This parameter specifies accessibility to the memory pool:

Value	Meaning
GROUP	The GETMAIN common memory pool can only be accessed by other tasks using the same user ID.
GLOBAL	The GETMAIN common memory pool can only be accessed by all user IDs in the system.

Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the values of the CMSCOPE and CMLSCOPE parameters to "GROUP".

Examples

The following example allows for three (3) megabytes of user storage at the next available megabyte boundary above the 16 megabyte line accessible to tasks with the same user ID.

```
ADARUN PROG=ADANUC,CMSIZE=3M,CMSCOPE=GROUP
```

The following example allows for three (3) megabytes of user storage at the next available megabyte boundary below the 16 megabyte line accessible to tasks with the same user ID.

```
ADARUN PROG=ADANUC,CMLSIZE=3M,CMLSCOPE=GROUP
```

CMSIZE or CMLSIZE Parameters: GETMAIN Memory Pool Use and Size

These parameters apply only to BS2000 environments running Sockets versions less than 2.2.

Parameter	Specify . . .	Minimum	Maximum	Default
	whether the GETMAIN is performed in a common memory pool and if so, its size in bytes (decimal)			
CMSIZE	above the 16MB line	0	address-limit	0
CMLSIZE	below the 16MB line	0	address-limit	0

Note: These parameters should only be used if required by an installation site, in which case the site will provide the needed value.

These parameters are involved with setting all required memory into a common memory pool for use by associated tasks on BS2000 above (CMSIZE) or below (CMLSIZE) the 16 megabyte line. These parameters determine whether a common memory pool is used and if so, its size:

- If the ..SIZE parameter value is 0, the default, the required application memory is obtained in the class 6 memory of the system.
- If the ..SIZE parameter value is *not* 0, the required application memory is obtained in a common memory pool of the specified size rounded up to the next megabyte.

Specific Product Recommendations

For Entire Net-Work 6 (mainframe) running BS2000 Sockets earlier than version 2.2, Software AG recommends setting the CMSIZE parameter to "31M" and the CMLSIZE parameter to "4096K".

Examples

The following example allows for three (3) megabytes of user storage at the next available megabyte boundary above the 16 megabyte line accessible to tasks with the same user ID.

```
ADARUN PROG=ADANUC,CMSIZE=3M,CMSCOPE=GROUP
```

The following example allows for three (3) megabytes of user storage at the next available megabyte boundary below the 16 megabyte line accessible to tasks with the same user ID.

```
ADARUN PROG=ADANUC,CMLSIZE=3M,CMLSCOPE=GROUP
```

CT Parameter: Command Timeout Limit

Parameter	Specify . . .	Minimum	Maximum	Default
CT	the maximum time (seconds) for interregion communication of results from Adabas to the user.	1	2147483647	60

For Adabas Review, this is the maximum number of seconds (more precisely, units of 1.048576 seconds) that can elapse from the time an Adabas Review hub command has been completed until the results are returned to the user through the interregion communication (operating-system-dependent).

This parameter is used to prevent a command queue element and attached buffer from being held for a long period of time for a user who has terminated abnormally.

Possible causes of a command timeout are

- user region is swapped out or cannot be dispatched;
- user is canceled;
- user has low priority in high activity system.

If the CT limit is exceeded,

- the command queue element and attached buffer are released;
- a message ADAM93 is printed; and
- if the user has not terminated, response code 254 (ADARSP254) is returned to the user program.

Example

The following example permits about 30 seconds to obtain a result through interregion communication from the Adabas Review hub

```
ADARUN PROG=ADAREV,CT=30
```

FORCE Parameter: Allow Nucleus Database ID or Review Hub Table Entry Overwrite

Parameter	Specify . . .	Possible Values	Default
FORCE	whether the nucleus or Adabas Review hub can overwrite an existing ID table entry.	YES | NO	NO

If running Adabas Review, this indicates whether the Adabas Review hub can overwrite an existing ID table entry. When a Review hub starts up, ADARUN scans the ID table to ensure that no entry exists for the Review hub. You can use the FORCE parameter to indicate whether the Review hub can overwrite an existing ID table entry.

The ID table entry is derived from the database ID and the job name. For Adabas Review, the ID table entry is derived from the Review hub ID (REVIEW=). The ID table entry is deleted when the nucleus terminates normally.

The FORCE parameter allows the nucleus or Adabas Review hub to overwrite the existing ID table entry and access the database.

Caution: Do not use the FORCE parameter unless absolutely necessary, or the integrity of the database could be lost. Ensure that no nucleus or Review hub is active for the ID table entry being overwritten.

Value	Meaning
YES	The nucleus or Adabas Review hub that is starting can overwrite an existing ID table entry. FORCE=YES is required when restarting a session that terminated abnormally with an ADAM98 message. In this case, the ID table still contains an active entry for the nucleus or Review hub. Overwriting the existing entry by specifying FORCE=YES prevents further communication to the overwritten nucleus or hub and causes loss of cross-memory environment resources, which cannot be restored until the next IPL.
NO	(default) If the ID table contains an entry for the nucleus or Adabas Review hub that is starting, the nucleus is denied access to the database or the Review hub is not permitted to start.

Note: In an Adabas Cluster Services or Adabas Parallel Services environment, the FORCE parameter applies to the NUCID, rather than the database ID, because a cluster nucleus builds an ID table entry for the NUCID.

A data integrity block (DIB) entry will only be removed once the ID Table initialization has been successful. Therefore, you must set IGNDIB and FORCE both to "YES" if either of the following occur:

- You receive a PARM ERROR 26 after parameter settings IGNDIB=NO and FORCE=YES were applied;

- You receive a PARM ERROR 23 after parameter settings IGNDIB=YES and FORCE=NO were applied.

Examples

The following example specifies that if the ID table contains an active entry for DBID 7, overwrite the entry.

```
ADARUN PROG=ADANUC,FORCE=YES,DBID=7
```

The following example specifies that if the ID table contains an active entry for the Adabas Review hub, overwrite the entry.

```
ADARUN PROG=ADAREV,FORCE=YES,REVIEW=202
```

GROUPS Parameter: User Group Interprocess Communication

This parameter applies to the BS2000 operating system only.

Parameter	Specify . . .	Possible Values	Default
GROUPS	whether BS2000 interprocess communication is limited to users with the same logon ID.	YES | NO	NO

This parameter limits the accessibility of ADARER and its ID table (IDT) to a group of users with the same logon ID.

Value	Meaning
YES	Allows the user(s) access to the IDT that starts under the same logon ID as the Adabas nucleus. When initialized, the IDT is set to GROUPS=YES; all users (nuclei, utilities, Entire Net-work, and all user tasks) must therefore also specify GROUPS=YES in either the ADARUN statement or using the ADALNK parameters.
NO	The default setting. All users on one machine have access to this IDT, even if they are logged on under a different logon ID.

Note: More than one IDT with the same name can be used if one is set to GROUPS=NO (the default) and the rest specify GROUPS=YES.

Example

The following example allows BS2000 users access to the IDT that starts under the same logon ID as this Adabas nucleus.

```
ADARUN PROG=ADANUC,DBID=11,GROUPS=YES
```

IDTNAME Parameter: Define ID Table Name

This parameter applies to the BS2000 operating system only.

Parameter	Specify . . .	Possible Values	Default
IDTNAME	an alternate ID table.	ADA*ccccc*	ADABAS5B

Defines a name for an (alternate) Adabas environment. The name must be eight characters long and must begin with the characters "ADA".

Note: All users of the new ID table must include the IDTNAME in either the ADARUN statement or ADALNK parameters.

Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the value of the IDTNAME parameter to "ADABAS5B".

Example

The following example defines the name ADAOURDB for the second Adabas environment.

```
ADARUN PROG=ADANUC,IDTN=ADAOURDB
```

LU Parameter: Length of Intermediate User Buffer Area

Parameter	Specify . . .	Minimum	Maximum	Default
LU	the size of the intermediate user buffer area.	none	none (see note 1)	65535 (see note 2)

The LU parameter is optional. The size specified must be large enough to accommodate all Adabas control information (204 bytes), all user buffers (format, record, search, value, ISN) that may be required for any given Adabas command, plus any user information passed from Adabas link routines to nucleus user exits. If the ADALNK has user exits, the addresses of the ACB and each buffer passed will be stored in this space. Also, if the ADALNK has declared LUSIZE and has user exits, this buffer size plus 64 bytes will also be stored in this space.

If the multifetch/prefetch option or a utility that needs large record/ISN buffers is to be used during the session, the setting of LU must be large enough to contain the buffers needed.

If you are using the ADARPD IQETBBROKERID NET communication (used with Event Replicator for Adabas), make sure that the webMethods EntireX Broker NET definition parameter IUBL is set to a value as large as the setting of this LU ADARUN parameter.

Notes:

1. An error occurs if the LU parameter specifies a value greater than the byte count implied by the NAB (number of attached buffers) parameter. On z/OS systems, LU cannot exceed a value greater than that produced by the following calculation: `(NABvalue x 4096)`; on z/VSE and BS2000 systems, LU cannot exceed a value greater than that produced by the following calculation: `(NABvalue x 4096) - 256`. For more information about the NAB parameter, read *NAB: Number of Attached Buffers*, in *Adabas Operations Manual*.
2. Due to the length of the record buffer of the utilities that need the nucleus, e.g., ADAULD, the default value is set to 65,535. If the value of LU is less than 65,535 for an Adabas session, a response code will occur when such a utility is running.

The LU parameter syntax is:

```
LU={ n | 65535 }
```

Specific Product Recommendations

- For Event Replicator Server databases running with Adabas 8, the LU parameter must be greater than or equal to 167,000.

  In addition, if data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work LU parameter must be greater than or equal to the LU parameter setting for the Event Replicator Server itself (greater than or equal to 167,000).
- The ADACHK utility can use large record buffer lengths when making nucleus calls to verify spanned Data Storage records or an index structure with many levels. If this is the case, the settings of your LU and NAB ADARUN parameters may need to be increased.

Example

The following example runs the Adabas nucleus with an Adabas intermediate user buffer area of 20,000 bytes.

```
ADARUN PROG=ADANUC,LU=20000
```

NAB Parameter: Number of Attached Buffers

Parameter	Specify . . .	Minimum	Maximum	Default
NAB	the number of attached buffers to be used.	1	varies, depending on the amount of available virtual storage	16

The NAB parameter defines the number of attached buffers to be used during the session. An attached buffer is an internal buffer used for interregion communication. It is required in all environments. Adabas allocates an attached buffer pool with a size equal to the value of NAB multiplied by 4096 bytes.

Note: The allocation for buffers in the attached buffer pool is done in 256 byte slots; this means that each allocation is rounded to a multiple of 256. For example, if a size of 300 bytes is needed, the allocated space is 512 bytes.

You may specify as many attached buffers as fit into the available virtual storage.

In environments running in 31-bit addressing mode, the attached buffer pool space is allocated above the 16-MB line.

The NAB parameter syntax is:

```
NAB={ n | 16 }
```

Specific Product Recommendations

- For Event Replicator Server databases, set parameter NAB to a value greater than or equal to: 41 * 10 * *the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server*.

  For example, if one Adabas nucleus will be sending data to the Event Replicator Server, set the NAB parameter greater than or equal to 410 (for example NAB=420).
- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.
- If data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work NAB parameter must also be set to a value greater than or equal to: 41 * 10 * *the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server*.
- Users of the Adabas Review hub should read *Storage Requirements* in the *Adabas Review Concepts Manual* for more information about the space requirements of the Command Queue for Adabas Review.

- The ADACHK utility can use large record buffer lengths when making nucleus calls to verify spanned Data Storage records or an index structure with many levels. If this is the case, the settings of your LU and NAB ADARUN parameters may need to be increased.

Example

The following example runs the Adabas Review hub nucleus with 50 attached buffers.

```
ADARUN PROG=ADAREV,NAB=50
```

NC Parameter: Number of Command Queue Elements

Parameter	Specify . . .	Minimum	Maximum	Default
NC	the maximum number of command queue elements.	20	32767	200

The number of command queue elements (CQEs) established for the Adabas or Review hub session determines the maximum number of Adabas commands that may be queued or be in process at any one time during the session.

Each call from the Adabas nucleus is assigned a CQE. The CQE is released when the user receives the results of the command, the Adabas Review hub has processed the command, or the user has been timed out..

192 bytes are required for each CQE.

Software AG recommends that you set NC high enough to allow one command per active user for possible synchronization during execution of the online SAVE database function of the ADASAV utility.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

For more information about the space requirements of the Command Queue for Adabas Review, refer to *Storage Requirements* in the *Adabas Review Concepts Manual*.

Specific Product Recommendations

- For Event Replicator Server databases, set parameter `NC` to a value greater than or equal to: `10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server`. For example, if one Adabas nucleus will be sending data to the Event Replicator Server, set the `NC` parameter greater than or equal to 10 (for example NC=20).
- If data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work NC parameter must also be set to a value greater than or equal to: `10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server`.

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example:

Run the Adabas nucleus with a maximum of 500 elements in the command queue.

```
ADARUN PROG=ADANUC,NC=500
```

The following example runs the Adabas Review hub nucleus with a maximum of 500 elements in the command queue.

```
ADARUN PROG=ADAREV,NC=500
```

PROGRAM Parameter: Program to Run

Parameter	Specify:	Possible Values	Default
PROGRAM	the program to be executed.	see table below	USER

This parameter specifies what to execute. The possible values are described in the following table:

Specify:	To start:
ADACOM	an ADACOM task (used in Adabas Cluster Services and Adabas Parallel Services environments) For more information, refer to your Adabas Cluster Services and Adabas Parallel Services documentation.
ADANUC	an Adabas nucleus For more information about executing an Adabas nucleus, read *Adabas Session Execution*, in the *Adabas Operations Manual*.
ADAREV	an Adabas Review hub. Specify this in conjunction with the ADARUN REVIEW parameter. For more information, refer to your Adabas Review documentation.
NETWRK	an Entire Net-Work node For more information, refer to your Entire Net-Work documentation.
RENTUSER	a user program to be run using a reentrant Adabas batch/TSO link routine. For more information, refer to description of the Adabas TP monitor installation in your Adabas installation documentation.

Specify:	To start:
USER	a user program to be run using a non-reentrant Adabas batch/TSO link routine. For more information, read *Linking Applications to Adabas*, in the *Adabas Operations Manual*
utility-name	an Adabas utility Specify an Adabas utility for *utility-name*. For more information, refer to the *Adabas Utilities Manual*.

Examples

The following example specifies that an Adabas nucleus is running.

```
ADARUN PROGRAM=ADANUC
```

The following example specifies that an Adabas Review hub is running.

```
ADARUN PROGRAM=ADAREV, REVIEW=202
```

The following example specifies that an Entire Net-Work node is running.

```
ADARUN PROGRAM=NETWRK
```

SVC Parameter: SVC Number

This parameter applies to the operating environments z/OS and z/VSE only.

Parameter	Specify . . .	Possible Values	Default
SVC	the Adabas SVC number or Adabas Review hub SVC number to be used for the session.	see text	45 (z/VSE) 249 (z/OS)

The SVC number is specified as an integer. It must correspond to the number used for the Adabas SVC at your installation.

The Adabas SVC or Adabas Review hub SVC are used to perform various Adabas internal functions under z/OS and z/VSE.

Valid SVC values are as follows:

z/OS 200-255

z/VSE 45 is recommended; any free SVC value can be used. See the Adabas Installation documentation for information about finding free values for z/VSE.

Example

The following example runs an Adabas session under z/OS using SVC 202 for the Adabas SVC.

```
ADARUN PROG=ADANUC,SVC=202
```

The following example runs an Adabas Review hub session under z/VSE using SVC 45 for the Adabas Review hub SVC.

```
ADARUN PROG=ADAREV,SVC=45
```

TARGETID Parameter: Entire Net-Work Target ID

Parameter	Specify . . .	Minimum	Maximum	Default
TARGETID	the unique Entire Net-Work target ID for this node.	1	65535	1

The TARGETID parameter is an optional ADARUN parameter that specifies the unique Entire Net-Work target ID of a node. It is synonymous with the Adabas ADARUN DBID parameter.

All target IDs used by Entire Net-Work, Adabas (database IDs), Natural global buffer pools, 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. For more information about global target IDs, read *Target ID Handling in the Network*, in the *Entire Net-Work Administration Guide*.

Example

The following example specifies an Entire Net-Work target ID of 12 for a node.

```
ADARUN TARGETID=12
```

TASKCTGY Parameter: Adabas Batch/TP Task Category Control

This parameter applies in BS2000 operating environments only.

Parameter	Specify . . .	Possible Values	Default
TASKCTGY	the Adabas task category.	BATCH | TP	BATCH

BS2000 TP (interactive processing) environments favor TP-processing tasks over batch tasks. By default, Adabas has batch status. You can use the TASKCTGY parameter to assign TP-processing priority to the Adabas nucleus.

Value	Meaning
BATCH	(default) The nucleus retains batch status.
TP	Issues a TINF macro to force the Adabas nucleus to TP status.

Example

The following example forces the nucleus to TP-processing status.

```
ADARUN PROG=ADANUC,TASKCTGY=TP
```

TCPURL Parameter: TCP/IP Universal Resource Locator

Parameter	Specify . . .	Possible Values	Default
TCPURL	the universal resource locator (URL) for the TCP/IP link.	(see text)	none

If TCPIP=YES, you can specify the information required to activate the direct TCP/IP link to the Adabas nucleus. The parameter value is a 20-byte address that conforms to the RFC specification for universal resource locators (URLs):

```
TCPURL=api-name:[//]stackid:port-number[:logging-setting[:allowipv6-setting]]
```

where:

api-name	A required three-character value identifying the application programming interface (API) to use. APIs BS2, HPS, CNS, EZA, and OES are currently supported.
stackid	A one to eight-character value identifying the stack to use: ■ For BS2, no value is needed. ■ For CNS, no value is needed. ■ For EZA, specify the TCPI stack ID of the VSE TCP/IP job. ■ For HPS, specify the name of the TCP/IP started task or job. ■ For OES on systems running a single TCP/IP stack, no value is needed. On systems running multiple TCP/IP stacks, specify the name of the TCP/IP started task or job. (No value is needed on systems with only one TCP/IP stack.)
port-number	A one to five-digit number in decimal notation.
logging-setting	A one-character setting indicating whether or not logging should be performed. Valid values are "Y" or "N"; a setting of "Y" turns logging on.
allowipv6-setting	A one-character setting indicating whether or not IPv6 addresses can be used. Valid values are "Y" or "N"; a setting of "Y" indicates that IPv6 addresses can be used.

Note: The forward slashes ("//") are optional and cannot be specified in z/VSE environments.

Example 1

The following examples run Adabas with a direct TCP/IP link to the nucleus.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://:12216
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=HPS://TCPLPAR1:12213
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=EZA://66:2112:N:Y
```

Example 2

The following example runs Adabas with a direct TCP/IP link to the nucleus and with logging turned on.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://:12216:Y ↩
```

Example 3

The following example runs Adabas with a direct TCP/IP link to the nucleus and with logging turned on for a system with multiple TCP/IP stacks. In this example, the TCP/IP stack with the started task or job name of TCPIPMVS is requested.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://TCPIPMVS:12216 ↩
```

Example 4

The following example runs Adabas with a direct TCP/IP link to the nucleus and allows IPv6 addresses to be used.

Note: Note if you choose not to specify the logging setting (*logging-setting*), but to specify the IPv6 setting (*allowipv6-setting*), you must still specify the colons for both the logging and IPv6 settings. This is why there are two colons between "12216" and "Y" in the following example.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://:12216::Y
```

Example 5

The following example runs Adabas with a direct TCP/IP link to the nucleus in a z/VSE environment.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=CNS::12217
```

Example6

The following example runs Adabas with a direct TCP/IP link to the nucleus in a BS2000 environment.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=BS2://:12217
```

ZIIP Parameter: Activate Usage of Adabas for zIIP

This parameter is valid for z/OS operating environments only.

The ZIIP parameter controls whether Adabas activates Adabas for zIIP (AZPAD).

Note: Adabas for zIIP requires its own license file (AZPAD), in addition to the standard Adabas license file. If the AZPAD license file is missing or found invalid, Adabas will start but will run as if ZIIP=NO had been specified.

Parameter	Specify . . .	Possible Values	Default
ZIIP	whether or not to activate Adabas for zIIP.	YES | NO	NO

Value	Meaning
YES	Usage of Adabas for zIIP is enabled. Adabas offloads CPU time from the general processors to System z Integrated Information Processors (zIIP). The purpose of this configuration is to reduce the CPU consumption on the general processors.
NO	The default setting. Usage of Adabas for zIIP is disabled. Adabas runs without the option to offload CPU time to zIIPs.

Dynamic Modification

If Adabas was started with ZIIP=YES, the setting of the ZIIP parameter can be changed at runtime - that is, set to NO and subsequently back to YES - using the ZIIP command from the operator

console, the ADADBS OPERCOM ZIIP function, or the Modify Parameters menu of Adabas Online System.

If Adabas was started with ZIIP=NO, the setting of the ZIIP parameter cannot be changed to YES later in this session.

ZIIP may be set differently for different nuclei in a cluster (it is a local, modifiable parameter).

Example

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

```
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*).

4 Entire Net-Work Parameter Statements

This section describes the Entire Net-Work parameter statements that define the Entire Net-Work environment.

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. TRANSDEF (translation definition) statements allow enhanced data translation among heterogeneous platforms and are described in *Enhanced Translation Definitions* in the *Entire Net-Work Administration Guide*.

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 Parameters***). Entire Net-Work control statements must always begin with the NODE statement, followed by one or more DRIVER statements (one for each driver type), and the related LINK statements. DRIVER and LINK statements may be in any order, as long as no LINK statement precedes its related DRIVER statement.

One DRIVER statement for each line driver in use must be specified to indicate the operational parameters for the line driver's access method and default values for the related links.

Each link to a neighboring node must 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.

Examples

Following are two examples of Entire Net-Work parameter statements for a node that has two drivers; one driver has two links and the other driver has one link. In the first example, statements are grouped by driver and link. In the second example, driver statements are grouped together, followed by a group of link statements. Both sets of statements achieve the same results.

Example 1

```
NODE      CENTRAL  parameter=value,parameter=value, ...
DRIVER    VTAM  parameter=value,parameter=value, ...
LINK      TOSITEX  VTAM parameter=value,parameter=value, ...
LINK      TOSITEY  VTAM parameter=value,parameter=value, ...
DRIVER    IUCV  parameter=value,parameter=value, ...
LINK      TOVSE  IUCV parameter=value,parameter=value, ...
```

Example 2

```
NODE      CENTRAL  parameter=value,parameter=value, ...
DRIVER    VTAM  parameter=value,parameter=value, ...
DRIVER    IUCV  parameter=value,parameter=value, ...
LINK      TOSITEX VTAM parameter=value,parameter=value, ...
LINK      TOVSE IUCV parameter=value,parameter=value, ...
LINK      TOSITEY VTAM parameter=value,parameter=value, ...
```

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 | 0 } ] ) ]
              [CQTIMER = { secs | 60 } ]
              [DEFINE = { Y | N } ]
              [DOMAIN = domain-name ]
              [DUMP = { ALL | NONE | BLOCKS | TRACETAB | BUFFERS | LINKAREA | FORMAT } ]
              [ENDCMD = { N | Y } ]
              [LOG = { ON | OFF | YES | NO | FULL | LIMIT | SHORT } ]
              [LOGBUF = { ON | OFF | YES | NO } ]
              [LOGBUFSZ = [ { 0 | logbuffersize } ] ]
              [LOGSIZE = rptbuffersize ]
              [MAXPATH = { linkcount | 4 } ]
              [MSGFORM = { msg-format | 2 } ]
              [NID0 = { N | Y } ]
              [PASSWORD = password ]
              [REMCMD = { N | Y } ]
              [REPLYTIM = { secs | 60 } ]
              [SNAPERR = (msg1 [,msg2 ][,msg3 ][,msg4 ] ) ]
              [START = { Y | N } ]
              [TIMER = { seconds | 30 } ]
              [ { TRACE | TROFF | TRON} = tracetype [, tracetype ] ... ]
              [UCMSG = { N | Y } ]
              [ULINK = { N | Y } ]
              [ ZEDCINIT = { Y | N } ]
              [ ZEDCSZ = { 4097 | nnnnn } ]
```

NODE Statement Parameters

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

- nodename
- BUFFERS Parameter
- CQTIMER Parameter
- DEFINE Parameter
- DOMAIN Parameter
- DUMP Parameter
- ENDCMD Parameter
- LOG Parameter
- LOGBUF Parameter
- LOGBUFSZ Parameter
- LOGSIZE Parameter
- MAXPATH Parameter
- MSGFORM Parameter
- NID0 Parameter
- PASSWORD Parameter

nodename

```
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 Parameter

Note: This parameter is obsolete. It is available for compatibility reasons only. If it is specified, any value will be ignored.

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

This optional parameter is useful only for z/VSE nodes that use the IUCV driver. It should be specified as BUFFERS=(,,,*n*), where *n* is the size, in bytes, of the page-fixed storage to be used by the IUCV driver. The first three buffer storage areas (the first three operands) specified in the BUFFERS parameter are ignored on all operating systems. The fourth buffer storage area (the fourth operand or the page-fixed buffer pool size) is still supported on all platforms, but is only required for the IUCV line driver in VSE. However, if you need to specify a size for the fourth operand, you must also specify placeholders (commas) for the first three operands.

Valid sizes should be specified as decimal values for the operands ranging from 0 to 2147483647 bytes; or, optionally, a value followed by the multiplier "K" (x 1024). If followed by "K", the value must range 0 to 2097151. If the fourth operand 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 all operating 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.

The following operands are expected in the BUFFERS syntax:

Syntax Variable	Description
abufsize	The asynchronous buffer pool size. Specifications for this value are ignored on all operating systems.
ltbufsize	The long-term buffer pool size. Specifications for this value are ignored on all operating systems.
stbufsize	The short-term buffer pool size. Specifications for this value are ignored on all operating systems.
pfxbufsize	The page-fixed buffer pool size. **Note:** A page-fixed buffer pool is required only for the IUCV line driver in z/VSE. The CTCA line driver page-fixes all required storage itself; it does not use the Entire Net-Work Buffer Pool Manager's page-fixed storage. This optional subparameter specifies the bytes to reserve for the page-fixed buffer pool, from which all requests for page-fixed buffers are filled. If zero (the default) is specified, no page-fixed buffer pool is allocated. The definition of page-fixed buffers requires a real storage definition in z/VSE environments.

CQTIMER Parameter

```
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.

Notes:

1. This parameter can be changed during Entire Net-Work operation by the **SET CQTIMER command**.
2. This parameter cannot be specified in WCT mode.

DEFINE Parameter

```
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.

Note: This parameter cannot be specified in WCT mode.

DOMAIN Parameter

```
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).

Note: This parameter cannot be specified in WCT mode.

DUMP Parameter

```
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 Parameter

```
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.

Note: This parameter cannot be specified in WCT mode.

LOG Parameter

```
LOG={ ON | OFF | YES | NO | FULL | LIMIT | 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. LIMIT performs the same function as LOG=FULL, except that most log entries are limited to 256 bytes of data. In many cases, this makes the amount of data logged more manageable. 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. In addition, consider using the LOGDON, LOGDOFF, LOGLON, LOGLOFF, LOGTON, and LOGTOFF operator commands instead. These operator command limit logging to specific drivers, links, and targets. For more information, read *Entire Net-Work Operator Commands*, elsewhere in this guide.

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

LOGBUF Parameter

```
LOGBUF={ ON | OFF | YES | NO}
```

This parameter controls the destination of log data. If "ON" or "YES" are specified, log data is written to an internal wraparound buffer instead of DDPRINT/NETPRNT. The size of the buffer is specified by the LOGBUFSZ parameter. If LOGBUF is set to "OFF" or "NO", log data is written to NETPRNT if it exists, otherwise it is written to DDPRINT. The default is "OFF". Note that LOGBUF does not turn on or off any logging; it just controls the destination of any log data that is generated by other parameters such as the LOG parameter. The contents of the log buffer are displayed when a SNAP operator command is issued.

The LOGBUF parameter setting can be changed during Entire Net-Work operation by the SET LOGBUF command.

LOGBUFSZ Parameter

```
LOGBUFSZ=[{0 | logbuffersize}]
```

This parameter specifies the size of the log buffer. Log data is written to the log buffer rather than NETPRNT/DDPRINT if the LOGBUF parameter is set to "ON". The log buffer size can range from 4096 to 2147483647 bytes, which can also be specified in kilobytes such as 32K, or megabytes such as 4M. The default size is "0" bytes. This buffer is allocated the first time log data is attempted to be written to it.

The LOGBUFSZ parameter setting can be changed during Entire Net-Work operation using the SET LOGBUFSZ command. If a log buffer exists and LOGBUFSZ is set to "0" by an operator command, the buffer is freed, and LOGBUF is turned off. If a log buffer exists and the value of LOGBUFSZ is changed by an operator command, the existing buffer is freed, and a buffer of the new size will be allocated the first time log data is written. Note that in both cases, the contents of the log buffer will be lost when the buffer is freed. The log buffer is not freed when LOGBUF is set to "OFF".

LOGSIZE Parameter

```
LOGSIZE=rptbuffersize
```

This optional parameter defines the size of a report 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 Parameter

```
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. For alignment reasons, the MAXPATH value should be an even number. If the value specified is an odd number, it is incremented up to the next even number.

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.

Notes:

1. This parameter can be changed during Entire Net-Work operation by the **SET MAXPATH command**.
2. This parameter cannot be specified in WCT mode.

MSGFORM Parameter

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

NID0 Parameter

Note: This parameter is obsolete. It is available for compatibility reasons only. This parameter cannot be specified in WCT mode.

```
NID0={ 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 Parameter

```
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).

Notes:

1. 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.
2. This parameter cannot be specified in WCT mode.

REMCMD Parameter

```
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. This parameter can be changed during Entire Net-Work operation by the **SET REMCMD command**.
2. This parameter takes effect only when the PASSWORD parameter is specified.
3. This parameter cannot be specified in WCT mode.

REPLYTIM Parameter

```
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 (ADARSP224) is returned to the user.

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

SNAPERR Parameter

```
SNAPERR=(msg1[,msg2][,msg3][,msg4])
```

Note: The SNAPERR parameter does not work for all error messages. Use SNAPERR only under the direction of your Software AG support representative.

This optional parameter can be used for diagnostics. When specified correctly, it will take a snap dump whenever specific Entire Net-Work errors occur. Specify the first seven characters of an Entire Net-Work error message number for each *msg#* parameter in the syntax. A maximum of four message numbers can be specified. If more than one message number is specified, enclose the list in parentheses and separate them with commas.

Once this parameter is set, a snap dump will be taken every time one of the errors listed in the SNAPERR parameter is encountered.

In the following example, a snap dump will be taken whenever error NET0151 occurs.

```
SNAPERR=NET0151
```

In the following example, snap dumps are taken whenever errors NET0151 or NET0028 occur.

```
SNAPERR=(NET0151,NET0028)
```

To clear a SNAPERR setting, issue the operator command SET SNAPERR=OFF, or SET SNAPERR with no value specified.

START Parameter

```
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 Parameter

```
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, 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).

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, and TRON Parameters

```
{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
CTCA	CTCA and FCTC line driver tracing
IUCV	IUCV line driver tracing
MAIN	Mainline tracing
RQM	Receive Queue Manager tracing
TCPI	TCP/IP line driver tracing
TQM	Transmission Queue Manager tracing
VTAM	VTAM line driver tracing
XCFD	XCF line driver 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 Parameter

```
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 Parameter

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

Note: This parameter cannot be specified in WCT mode.

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.

ZEDCINIT Parameter

```
ZEDCINIT={ Y | N }
```

This optional parameter indicates whether zEnterprise Data Compression (zEDC) can occur. Valid values are "Y" or "N". If "Y" is specified, Entire Net-Work will try to initialize zEDC. If "N" is specified, Entire Net-Work will not attempt to initialize it. The default is "N".

zEnterprise Data Compression (zEDC) can occur only on z/OS operating systems. Consequently, ZEDCINIT=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)*.

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.

ZEDCSZ Parameter

```
ZEDCSZ={ 4097 | nnnnn }
```

This optional parameter identifies, in kilobytes, the smallest payload that may be compressed by zEDC compression. Payload size calculations do not include the Entire Net-Work headers. The default minimum payload size that can be compressed is 4,097K. Only payload greater than 4,097K will be compressed.

zEnterprise Data Compression (zEDC) can occur only on z/OS operating systems. Consequently, the ZEDCSZ 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)*.

The following table provides some examples:

ZEDCSZ Setting	Result
0	All buffers, regardless of actual size, will be compressed.
4	Entire Net-Work compresses buffers with sizes greater than 4K.
32	Entire Net-Work compresses buffers with sizes greater than 32K.

Entire Net-Work DRIVER Statement

The Entire Net-Work DRIVER control statement defines the line driver type (i.e., VTAM, CTCA, FCTC, IUCV, SSL, TCPI, TCPX, or XCF) to be loaded. With the exception of TCPI and TCPX, only one DRIVER statement may be specified for a given line driver type. The following is the syntax of the DRIVER statement:

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

where *drivername* is a four-character access method name for the driver type, as follows:

Driver	Module Name	Access Method
CTCA	NETCTCA	Channel-to-channel communications adapter
FCTC	NETFCTC	Fast channel-to-channel communications adapter
IUCV	NETIUCV	Inter-machine communication
SSL	NETSSL	For more information about Encryption for Entire Net-Work, contact your Software AG sales support representative.
TCPI	NETTCPI	TCP/IP access method
TCPX	NETTCPX	TCP/IP access method
TCPL	NETTCPL	TCP/IP access method exclusive to Entire Net-Work Light

Driver	Module Name	Access Method
VTAM	NETVTAM	Virtual Telecommunications Access Method
XCFD	NETXCF	IBM's Cross-System Coupling Facility

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.

- *CTCA and FCTC DRIVER Statements*
- *IUCV DRIVER Statement*
- *TCP/IP DRIVER Statement*
- *TCPX DRIVER Statement*
- *TCPL DRIVER Statement*
- *VTAM DRIVER Statement*
- *XCF DRIVER Statement*

Note: For more information about Encryption for Entire Net-Work, contact your Software AG sales support representative. The documentation for Encryption for Entire Net-Work is delivered separately from the other Entire Net-Work documentation.

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 the four-character name of the related line driver. 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.

- *CTCA and FCTC LINK Statements*
- *IUCV LINK Statement*
- *TCP/IP LINK Statement*
- *TCPX LINK Statement*
- *VTAM LINK Statement*
- *XCF LINK Statement*

Note: For more information about Encryption for Entire Net-Work, contact your Software AG sales support representative. The documentation for Encryption for Entire Net-Work is delivered separately from the other Entire Net-Work documentation.

5 Entire Net-Work Operator Commands

This section describes the Entire Net-Work operator commands.

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, BS2000 and z/VSE are provided.

The Entire Net-Work CTCA, FCTC, IUCV, TCPL, TCP/IP, TCPX, VTAM, and XCF line drivers have the ability to process operator commands that are directed to a specific link or directly to the driver. The operator commands that are specific to a line driver are described in the section about the line driver.

- *CTCA and FCTC Operator Commands*
- *IUCV Operator Commands*
- *TCP/IP Operator Commands*
- *TCPX Operator Commands*
- *TCPL Operator Commands*
- *VTAM Operator Commands*
- *XCF Operator Commands*

Operator Commands Summary

The following table summarizes the Entire Net-Work operator commands:

Use Command	Arguments	To
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	various	Display links, nodes, targets, user buffer blocks, paths or statistics.

Use Command	Arguments	To
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.
HALT	---	Terminate Entire Net-Work session.
HELP	---	List available operator commands.
LICREFRESH	---	Refresh Entire Net-Work license.
LOGDON	driver	Activates logging for a specific driver.
LOGDOFF	driver	Deactivates logging for a specific driver.
LOGLON	link	Activates logging for a specific link name. A wildcard can be used to activated logging for link names with certain characters in their names.
LOGLOFF	link	Deactivates logging for a specific link name. A wildcard can be used to deactivate link logging.
LOGTON	target	Activates logging for a specific target ID.
LOGTOFF	target	Deactivates logging for a specific target.
NETEND	---	Terminate Entire Net-Work session.
OPEN	driver	Reopen a driver after a close or access method failure. This command and the START command are currently synonymous.
OPEN NETPRNT	---	Open the NETPRNT file and route all trace and snap output to the NETPRNT file.
PROBE	nodename or node ID and bytes	Send a probe message to a node.
REFRESH	---	Refresh the Net-Work node's list of known targets.
RESUME	link	Resume sending messages via this link.
SET	various	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	Reopen a driver after a close or access method failure. This command and the OPEN command are currently synonymous.
STOP	---	Terminate Entire Net-Work session.
SUSPEND	link	Stop sending messages on this link.
TERMINATE	---	Terminate Entire Net-Work session.
TRANSLAT	various	Add, delete, or display translation definition statements.
VERIFY	target	Verify whether a target is still active or present in the network.

Operator Command Descriptions

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

- VERIFY Command

ADAEND, END, HALT, NETEND, STOP, and TERMINATE Commands

```
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 such as STOP TCPI 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 Command

```
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 Command

```
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 Command

```
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 Command

```
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.

Notes:

1. DEFINE commands will only be accepted if the **NODE statement DEFINE parameter** is set to"Y".
2. The DEFINE command is not applicable when operating in WCT mode.

Note: DEFINE commands will only be accepted if the **NODE statement DEFINE parameter** is set to"Y" .

The following example applies to the Entire Net-Work VTAM line driver:

```
DEFINE LINK TOPSYS VTAM APP=FSYSWK,ACQ=N,BL=Y,COMP=N, -
     STATB=Y,STATC=N,WE=10,MAXRU=85
```

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 Command

```
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.

Note: The DISABLE command is not applicable when operating in WCT mode

DISCONNECT Command

```
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.

Note: The DISCONNECT command is not applicable when operating in WCT mode.

DISPLAY Command

```
DISPLAY  { ALINKS | CPU | CQ | CQE | CSCI | DETAIL | LINKS | LOGGING | NODES | PATHS ↩
| STATS | TARGETS | UBQ |  ZAPS | ZSTATS }
                  [ {name | string* } ]
```

Displays current information about the specified network component. Only one component type (link, node, path, logging, statistics, target, or zaps) 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 Manual*.

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 LINKS and ALINKS, 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.

Note: The DISPLAY qualifiers CQ, CQE CSCI, NODES and PATHS are not applicable when operating in WCT mode.

DISPLAY ALINKS Example

The following is an example of DISPLAY ALINKS output. DISPLAY ALINKS lists currently active links only:

```
F NETWK,D AL
NET0120I: VTAM LINK LNKE TO NODE ENODE  STAT=ACTIVE
NET0120I: VTAM LINK LNKA TO NODE ANODE  STAT=ACTIVE
```

DISPLAY CPU Example

The DISPLAY CPU command displays the Central Processor Unit (CPU) time, the time that the Network has been active and the time the Network has been in wait mode within the active time. The units are Store Clock hours:minutes: seconds.microseconds. Here is an example of DISPLAY CPU output:

```
F NETWK,D CPU
NET0166I:  Duration           0:00:07.755607
NET0167I:  Wait-Time          0:00:05.000790
NET0168I:  CPU Time           0:00:00.485279
```

DISPLAY CQ Example

The DISPLAY CQ command displays all posted command queue elements (CQEs). It displays information about each CQE, including command code, database and file number, buffer length, sequence number, age, job name, user ID, and status. Here is an example of DISPLAY CQ output:

```
F NETWK,D CQ
NET0175I: Current CQ
NET0175I: Cmd=L1,Db=68,File=13,Len=100532,Seq-Nr=420
NET0175I: Time=0 sec ago,Job='USAKCTXG',TID=C1F0F0F200000000,Active
NET0175I: -- 1 CQEs displayed
```

DISPLAY CQE Example

The DISPLAY CQE command displays detailed information about a specific command queue element (CQE). It takes the command sequence number as input, which can be determined from the output of a DISPLAY CQ command. The command sequence number should be preceded by a blank, comma, or an equal sign (=).

Here is an example of DISPLAY CQE output. First this user enters D CQ to list all of the CQEs and gets this output:

```
NETI29 Oper cmd: D CQ
NET0175I KC1NET2A: Current CQ
NET0175I KC1NET2A: Cmd=L1,Db=68,File=13,Len=100532,Seq-Nr=567
NET0175I KC1NET2A: Time=6 sec agoJob='USAKCTXG',TID=C1F0F0F100000000,Active
NET0175I KC1NET2A: -- 1 CQEs displayed
```

Then the user decides to obtain details about the CQE with command sequence number 567 by specifying `D CQE=567` and receives this output:

```
NETI29 Oper cmd: D CQE=567
NET0177I KC1NET2A: CQE Seq-Nr=567,Status=Active
NET0177I KC1NET2A: Addr=150BF800,Cmd=L1,Db=68,File=13,Len=100532
NET0177I KC1NET2A: Bin/out=x1008,Flags=x1B80,AB=1029F000,UB=1034251C
NET0177I KC1NET2A: Job='USAKCTXG',TID=C1F0F0F100000000
NET0177I KC1NET2A: Uid=0009A10E2818000040404040404040000F9FB80C1F0F0F100000000
NET0177I KC1NET2A: Time=CC1DC707 0E15E18A; 15 sec ago ↩
```

For the specified command sequence number, DISPLAY CQE displays the sequence number, status, address of the CQE, command code, database and file numbers, length of attached buffers, buffer in/out flags, CQE flags, address of the attached buffers, address of the user buffer (UB), job name, terminal identifier (TID), user ID, the time in STCK format, and the age in seconds.

Note: `D CQE 567` and `D CQE,567` would have produced the same result.

DISPLAY CSCI Example

The following is an example of DISPLAY CSCI output:

```
F NETWK,D CSCI
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
```

DISPLAY DETAIL Examples

The DISPLAY DETAIL command shows details about local targets in ascending order, using the format shown in these examples. It allows you to display details about the node , its location, name and local targets. The local target information contains the target ID, target type, job name and the job number/ID that started it as well as product details.

The following example displays all target details:

```
F NETWK,D DE [ALL or *]

 NET0190I: Node=NET9990,Nodeid=9990,Host=DAEF
 NET0190I:  00011: REV V4.7.1, Isolated,Jobname=REV471,Jobid=14711
 NET0190I:  00063: ADA V8.2.6, Isolated,Jobname=DB0063,Jobid=14812
 NET0190I:  00081(1): ASM V8.2.6, Database,Jobname=CLNUC01,Jobid=14921
 NET0190I:  00081(10): ASM V8.2.6, Database,Jobname=CLNUC10,Jobid=14922
 NET0190I:  09990: WCP V6.3.2, Communicator,Jobname=NET9990,Jobid=14999
```

In this example, the network has the node name "NET9990" with the target number 9990 on the host/LPAR "DAEF". Target 11 is an Adabas Review hub, started with a job name "REV471" where the system runs it with job ID 14711 (J14711). Target 81 is a cluster with two nuclei DBIDs, (1) and (10).

The target types listed in the message can be any of the target types described in message NET0190I, in the *Entire Net-Work Messages and Codes Manual*.

The following example displays specific target details:

```
F NETWK,D DE 11,63
NET0190I: Node=NET9990,Nodeid=9990,Host=DAEF
NET0190I:  00011: REV V4.7.1, Isolated,Jobname=REV471,Jobid=14711
NET0190I:  00063: ADA V8.2.6, Isolated,Jobname=DB0063,Jobid=14812    ↩
```

The following example displays a range of target details:

```
F NETWK,D DE 20-100
NET0190I: Node=NET9990,Nodeid=9990,Host=DAEF
NET0190I:  00063: ADA V8.2.6, Isolated,Jobname=DB0063,Jobid=14812
NET0190I:  00081(1): ASM V8.2.6, Database,Jobname=CLNUC01,Jobid=14921
NET0190I:  00081(10): ASM V8.2.6, Database,Jobname=CLNUC10,Jobid=14922
NET0190I:  09990: WCP V6.3.2, Communicator,Jobname=NET9990,Jobid=14999          ↩
```

The following example shows errors that occur when no targets are in the requested range:

```
F NETWK,D DE 20-50
NET0190I: Node=NET9990,Nodeid=9990,Host=DAEF
NET0191I:  No target found
```

DISPLAY LINKS Examples

The following is an example of DISPLAY LINKS output:

```
F NETWK,D L
NET0120I: VTAM LINK LNKALS   TO NODE ALSNODE  STAT=DISC
NET0120I: VTAM LINK LNKE     TO NODE ENODE    STAT=ACTIVE
NET0120I: VTAM LINK LNKA     TO NODE ANODE    STAT=ACTIVE
NET0120I: VTAM LINK LNKVM    TO NODE UNKNOWN  STAT=OPEN
```

The following is an example of DISPLAY LINKS output for all links whose names begin with "TO":

```
F NET1,D L TO*
NET0120I: VTAM LINK TOSIX    TO NODE SIX      STAT=ACTIVE
NET0112I:              2 MSGS;              2 TR.BLKS
NET0120I: VTAM LINK TOTWO    TO NODE TWO      STAT=ACTIVE
NET0112I:              3 MSGS;              3 TR.BLKS
NET0120I: VTAM LINK TONINE   TO NODE UNKNOWN  STAT=CONSTA
NET0112I:              0 MSGS;              0 TR.BLKS
```

The following is an example of DISPLAY LINKS output with a qualifier for an FCTC link:

```
F NET1,D L NET2
NET0120I NET1    : FCTC link NET2     to node NET2     stat=active
NETC050I:  8155(W) Active         8154(R) Active
```

DISPLAY LOGGING Examples

The following is an example of DISPLAY LOGGING output showing the settings of the LOG, LOGBUF, LOGBUFSZ, and SNAPERR NODE parameters:

```
F node,DISPLAY LOGGING
NET0143I node : Selective Logging DRIVER NETTCPI  OFF.
NET0143I node : >> No Links being logged
NET0143I node : Selective Logging DRIVER NETTCPX  OFF.
NET0143I node : >> No Links being logged
NET0158I node : LOG=OFF
NET0159I node : LOGBUF=ON , LOGBUFSZ=0000065536
NET0160I node : SNAPERR=NET0001,NET0002,NET0003,NET0004  ↩
```

The following example of DISPLAY LOGGING output shows that selective logging was started for the VTAM line driver, link V246Y, and target 1001:

```
NET0143I: Selective Logging DRIVER NETXCF   OFF.
NET0143I: >> No Links being logged             .
NET0143I: Selective Logging DRIVER NETVTAM   ON.
NET0143I:                   LINK   V246Y     ON.
NET0143I: Selective Logging DRIVER NETTCPX  OFF.
NET0143I: >> No Links being logged             .
NET0143I: Selective Logging DRIVER NETTCPI  OFF.
NET0143I: >> No Links being logged             .
NET0143I: Selective Logging DRIVER NETCTCA  OFF.
NET0143I: >> No Links being logged             .
NET0143I: Selective Logging TARGET     1001  ON. ↩
```

DISPLAY NODES Examples

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
```

DISPLAY PATHS Example

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
```

DISPLAY STATS Examples

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 CPU/Active Time/Wait Time is also displayed .

The following is an example of DISPLAY STATS output. 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. The NET0166I, NET0167I, and NET0168I messages are for CPU and Active/Wait times.

```
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: ------------------------------------------------------------
NET0166I: Duration           0:00:07.755607
NET0167I: Wait-Time          0:00:05.000790
NET0168I: CPU Time           0:00:00.485279
```

The following additional statistics are shown on z/OS systems when messages have been compressed with zEDC. These statistics are issued when the DISPLAY STATS command is issued and at shutdown:

```
NET0180 -      97.819K messages were compressed
NET0181 -       1.036G bytes uncompressed data
NET0182 -      91.494M bytes compressed data
NET0183 -  91.38 % compression rate
          Message sizes     Smallest:      Largest:      Average:
NET0184 - Uncompressed:        10,496     1,000,496        11,109
NET0185 -   Compressed:           904        63,362           957
```

Additional statistics are shown on z/OS systems when the ADARUN parameter `ZIIP=YES` was specified. These statistics are also displayed via the `DISPLAY ZSTATS` operator command, and at session termination.

DISPLAY TARGETS Example

The DISPLAY TARGETS command lists the targets in ascending order. The following is an example of DISPLAY TARGETS output:

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

Substitutions for the values in parentheses (*x*-*y*) following the target ID in each message have specific meaning. Substitutions for the first value (*x*) are described in message NET0124I, in the *Entire Net-Work Messages and Codes Manual*.

Substitutions for the second value (*y*) can be:

- "T" (for types I, L, and V above, which use Adabas 7 translation); or
- "N" (normal).

DISPLAY UBQ Example

The DISPLAY UBQ command displays all the user buffer (UB) blocks associated with messages in process via ADALNK, the Adabas SVC, or the router to a target. The following are displayed for each user block: the command code, database and file number, the checksum (in hexadecimal), the user buffer flags and address in hexadecimal, the job name that sent it, the number of seconds elapsed since sending, and the 28-byte user ID in hexadecimal.

The following is an example of DISPLAY UBQ output:

```
F NETWK,D UBQ
NETI29       OPER CMD: D UBQ
NET0178I: UBQ Cmd=L1,Db=63,Fnr=3,Checksum=CC24F6274BCD0600
NET0178I:  Addr=01036E00,Flags=C510,Job='SDE     ',time ago=5
NET0178I:  Uid=0602200421500000E2C9F1F840404040000000000E4F0F0F100000000
           -- 1 UBs displayed
```

DISPLAY ZAPS Example

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:  NWTCPOES Date 2011-09-22, Version: 06.03, SP 01, Base WU631000
         Zaps WU631010
```

DISPLAY ZSTATS Example

The DISPLAY ZSTATS command can be used in a session that was started with ADARUN parameter `ZIIP=YES` to display statistics about the execution of Entire Net-Work in TCB mode and SRB mode and about the CPU time consumed on System z Integrated Information Processors (zIIP) and general processors (GP).

The statistics displayed by DISPLAY ZSTATS correspond and are equivalent to the zIIP-related statistics displayed by the **DISPLAY STATS** command, and at end of session.

See *Understanding the zIIP-Related Statistics* in the *Entire Net-Work for zIIP* documentation for detailed information about the various statistical figures.

DUMP Command

```
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 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.

Note: The ENABLE command is not applicable when operating in WCT mode.

HELP Command

```
HELP
```

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

LICREFRESH Command

```
LICREFRESH
```

Use the LICREFRESH command to:

- reload the license module or reread the license file from the library identified by the DDLIC JCL statement in the Entire Net-Work startup job
- display and check the license identified by the DDLIC JCL statement in the Entire Net-Work startup job

LOGDON Command

```
LOGDON {driver[,driver]... | (driver[,driver]...) | ALL }
```

Activates logging for the specified drivers. Valid values for *driver* include: TCPI, TCPX, VTAM, CTCA, FCTC, and XCFD. The driver names can be specified in parentheses or out of parentheses. At least one driver name or the term "ALL" must be specified. If you specify ALL, logging is activated for all drivers.

In the following example, logging for the TCPI, TCPX, and VTAM drivers is started.

```
LOGDON TCPI, TCPX, VTAM
```

LOGDOFF Command

```
LOGDOFF {driver[,driver]... | (driver[,driver]...) | ALL }
```

Deactivates logging for the specified drivers. Valid values for *driver* include: TCPI, TCPX, VTAM, CTCA, FCTC, and XCFD. The driver names can be specified in parentheses or out of parentheses. At least one driver name or ALL must be specified. If you specify ALL, logging is deactivated for all drivers.

In the following example, logging for the TCPI, TCPX, and VTAM drivers is stopped.

```
LOGDOFF (TCPI, TCPX, VTAM)
```

LOGLON Command

```
LOGLON {linkname[,linkname]... | (linkname[,linkname]...)  | ALL }
```

Activates logging for currently defined links that match the linknames specified in the command. The link names can be specified in parentheses or out of parentheses. At least one link name or "ALL" must be specified. If you specify "ALL", logging is activated for all links. You can use an asterisk (*) as a wildcard at the end of a link name ("linkname*") to specify a pattern of link names; logging will be activated for all currently defined links that match the pattern.

In the following example, logging is activated for links named XYZ and for links beginning with the characters "ABCD".

```
LOGLON XYZ, ACBD*
```

LOGLOFF Command

```
LOGLOFF {linkname[,linkname]... | (linkname[,linkname]...) | ALL }
```

Deactivates logging for currently defined links that match the linknames specified in the command. The link names can be specified in parentheses or out of parentheses. At least one link name or ALL must be specified. If you specify ALL, logging is deactivated for all links. You can use an asterisk (*) as a wildcard at the end of a link name ("linkname*") to specify a pattern of link names; logging will be deactivated for all currently defined links that match the pattern.

In the following example, logging is deactivated for links named XYZ and for links beginning with the characters "ABCD".

```
LOGLOFF (XYZ, ACBD*)
```

LOGTON Command

```
LOGTON {target-ID[,target-ID]... | (target-ID[,target-ID]...) }
```

Activates logging for the targets specified in the command. The target IDs can be specified in parentheses or out of parentheses. At least one target ID must be specified. Valid target ID values range from "1" through "65535".

In the following example, logging is activated for targets 12 and 181.

```
LOGTON 12, 181
```

LOGTOFF Command

```
LOGTOFF {target-ID[,target-ID]... | (target-ID[,target-ID]...) | ALL }
```

Deactivates logging for the targets specified in the command. The target IDs can be specified in parentheses or out of parentheses. At least one target ID or ALL must be specified. If you specify ALL, logging is deactivated for all targets. Valid target ID values range from "1" through "65535".

In the following example, logging is deactivated for targets 12 and 181.

```
LOGTOFF (12, 181)
```

OPEN Command

```
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 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 Command

```
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.

Note: The PROBE command is not applicable when operating in WCT mode.

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)
```

REFRESH Command

```
REFRESH
```

The REFRESH command updates Net-work's target table from the contents of the Adabas ID table on the same SVC as the Net-Work node. Target IDs present in the ID table that are not known to Net-Work will be added, and other Net-work nodes will be informed by broadcast messages. If a target is already known elsewhere in the network, an error message is issued and that target is ignored, in order to avoid target conflicts.

Note: When operating in WCT mode, the target is added without informing Net-Work nodes about the target and no check is done to ensure the target is not already known.

RESUME Command

```
RESUME  linkname
```

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

Note: The RESUME command is not applicable when operating in WCT mode.

SET Command

```
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. The parameters allowed for the SET command are a subset of those defined on the Entire Net-Work NODE statement. This section covers the following topics:

SET CQTIMER Command

```
SET CQTIMER=secs
```

The CQTIMER parameter of the SET command can be used to set 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, read about the **CQTIMER parameter** of the **NODE statement**, elsewhere in this guide.

SET DUMP Command

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

The DUMP parameter of the SET command can be used to set 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 Command

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

The LOG parameter of the SET command can be used to regulate control flow and logging of selected data areas to the printer data set. For more information, read about the **LOG parameter** of the **NODE statement**, elsewhere in this guide.

SET LOGBUF Command

```
SET LOGBUF={ ON | OFF | YES | NO}
```

The LOGBUF parameter of the SET command can be used to control the destination of log data. For more information, read about the **LOGBUF parameter** of the **NODE statement**, elsewhere in this guide.

SET LOGBUFSZ Command

```
SET LOGBUFSZ=[{0 | logbuffersize}]
```

The LOGBUFSZ parameter of the SET command can be used to specify the size of the log buffer. For more information, read about the **LOGBUFSZ parameter** of the **NODE statement**, elsewhere in this guide.

SET MAXPATH Command

```
SET  MAXPATH=linkcount
```

The MAXPATH parameter of the SET command can be used to set the maximum path link, specified in number of links, that a message from users on this node is expected to travel. For more information, read about the **MAXPATH parameter** of the **NODE statement**, elsewhere in this guide.

SET MSGFORM Command

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

The MSGFORM parameter of the SET command can be used to set the message format of console messages and DDPRINT output. For more information, read about the **MSGFORM parameter** of the **NODE statement**, elsewhere in this guide.

SET PASSWORD Command

```
SET  PASSWORD=password
```

The PASSWORD parameter of the SET command can be used to set the password that controls access to the Programmable Command Interface (PCI). For more information, read about the **PASSWORD parameter** of the **NODE statement**, elsewhere in this guide.

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

SET REMCMD Command

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

The REMCMD parameter of the SET command can be used to allow or disallow remote access to the Programmable Command Interface. For more information, read about the **REMCMD parameter** of the **NODE statement**, elsewhere in this guide.

SET REPLYTIM Command

```
SET  REPLYTIM=secs
```

The REPLYTIM parameter of the SET command can be used to set the time, in seconds, that this node is to wait for a reply to a user request before timing out. For more information, read about the ***REPLYTIM Parameter*** of the **NODE statement**, elsewhere in this guide.

SET SNAPERR Command

```
SET SNAPERR=[(msg1[,msg2][,msg3][,msg4]) | OFF]
```

Note: The SET SNAPERR operator command does not work for all error messages. Use SNAPERR only under the direction of your Software AG support representative.

This optional parameter can be used for diagnostics. When specified correctly, it will take a snap dump whenever specific Entire Net-Work errors occur. For more information, read about the ***SNAPERR Parameter*** of the **NODE statement**, elsewhere in this guide.

To clear a SNAPERR setting, issue the operator command SET SNAPERR=OFF, or SET SNAPERR with no value specified.

SET TRACE, TROFF, and TRON Commands

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

The TRACE, TROFF, and TRON parameters of the SET command can be used to set the trace control parameters for performing program traces. For more information, read about the **TRACE, TROFF, and TRON parameters** of the **NODE statement**, elsewhere in this guide.

SET UCMSG Command

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

The UCMSG parameter of the SET command can be used to control whether messages are issued in upper case or mixed case. For more information, see the **UCMSG parameter** of the **NODE statement**, elsewhere in this guide.

SET ULINK Command

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

The ULINK parameter of the SET command can be used to allow or disallow multiple links to an adjacent Entire Net-Work node. For more information, see the **ULINK parameter** of the **NODE statement**, elsewhere in this guide.

SET ZIIP Command

```
SET ZIIP={ ON | OFF | YES | NO}
```

Use the SET ZIIP operator command to turn on or off the use of System z Integrated Information Processors (zIIP) in the Entire Net-Work kernel. Issuing SET ZIIP=YES will tell the Entire Net-Work kernel to run in SRB mode when possible and enable the use of zIIPs.

Issuing SET ZIIP=NO will tell the Entire Net-Work kernel to stay in TCB mode and disable the use of zIIPs.

The SET ZIIP operator command may only be used in an Entire Net-Work kernel started with ADARUN `ZIIP=YES`.

Examples

For example, the following operator command:

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

is equivalent to the following NODE statement specification:

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

SNAP Command

```
SNAP  { BPH | CQ | CURRMSG |  LOGBUF | 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.
LOGBUF	Log buffer.
MAIN	Header of mainline module.
MYBLK	Central control block.
TRACE	Internal trace table.
UBQ	UB-Queue (currently active Adabas commands).

START Command

```
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 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.

Note: The SUSPEND command is not applicable when operating in WCT mode.

TRANSLAT Command

```
TRANSLAT parameters
```

The TRANSLAT command is used to add, delete, and display translation definition statements. For syntax and parameters, see the section *Enhanced Translation Definitions* in the *Entire Net-Work Administration Guide*.

VERIFY Command

```
VERIFY target-id
```

The VERIFY command is used to verify whether a target is still active or present in the network.

This command will report on the status of the target. If the target is not active but still locally in the network, it is removed and will no longer be seen in the DISPLAY TARGETS command. If the target is active and on another node, the target, target attributes and node will be displayed. If the target is active locally, but is not known to Entire Net-Work, Entire Net-Work becomes aware of the target and broadcasts this information throughout the network.

Specify a valid numeric target-ID in the range from 1 to 65535 with this command.

Note: This command potentially changes the status information within Entire Net-Work. You should therefore use it with care, and have a clear understanding of what's happening.

VERIFY command processing takes no action for virtual or relocatable targets managed by Adabas Cluster Services. It acts only on targets that are local to the Entire Net-Work node where the command is issued.

- If Entire Net-Work believes that the target is active on this node, but the target is not actually active, Entire Net-Work changes the target's current status on this node as no longer active and broadcasts this information throughout the network.
- If the target is active locally, but is not known to Entire Net-Work, Entire Net-Work becomes aware of the target and broadcasts this information throughout the network.
- If the target is not known in the network and is not active locally, error message NET0172 is issued and no action is taken.
- If the target is active locally, but Entire Net-Work believes it is active on another node, correct the situation in two steps:
  1. Issue the VERIFY command on the Entire Net-Work node where the target is believed to be active. This will make the target inactive in Entire Net-Work's view.
  2. Issue the VERIFY command on the Entire Net-Work node where the target is currently active. This will make Entire Net-Work aware of the target in the correct location.

Examples

For example, the following command verifies whether target 181 is still in the network:

```
VERIFY 181 ↩
```

When the target is active and local to the node, one of the following messages is issued:

```
NET0171I Verify target 00181 already active
NET0174I Verify target 00181 is relocatable or virtual
```

If the target is not present in the network, the following message is issued:

```
NET0172I Verify target 00181 not in this Net-Work
```

If the target is active and on another node, the following message is issued:

```
NET0124I: Target ttttt (a-a) active on node nnnnnnnn
```

where *nnnnnnnn* is the node name, *ttttt* is the target ID, and (*a-a*) are the attributes of the target.

If the target is locally active but is not known to Entire Net-Work, Entire Net-Work becomes aware of the target and issues the following broadcast message:

```
NET0042I  Local Target 000181 active
```