

Adabas Cluster Services

Reference

Version 8.1.3

June 2008

Adabas Cluster Services

This document applies to Adabas Cluster Services Version 8.1.3 and to all subsequent releases.

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

Copyright © Software AG 2008. All rights reserved.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

Table of Contents

1 Reference	1
2 Initialization Parameters	3
3 ADACOM Initialization Parameters	5
CMDMGR - ADACOM Use	6
DBID - Database Identification	7
FORCE - Force Restart after Abnormal Termination	7
NU - Number of User Table Elements	8
SVC - Interregion Communication Number	9
4 About ADARUN Parameters for Cluster Nuclei	11
Global ADARUN Parameters	12
ADARUN Parameter Usage in Cluster Environments	13
5 Entire Net-Work (NETWRK) Initialization Parameters	19
Guidelines	21
CT - Command Timeout Limit	21
FORCE - Overwrite ID Table Entry	22
LU - Length of Intermediate User Buffer	23
NAB - Number of Attached Buffers	23
NC - Number of Request Queue Elements	24
SVC - Interregion Communication Number	24
TARGETID - Entire Net-Work Target Identification	25
6 Useful ADARUN Parameters	27
CLOGMRG : Cluster Command Log Merge	28
CLUCACHENAME : Cluster Cache Structure/Area Name	29
CLUGROUPNAME : Cluster Group Name	30
CLULOCKNAME : Cluster Lock Structure/Area Name	30
CLUSTER : Adabas Nucleus Cluster Session Control	31
DIRRATIO / ELEMENTRATIO : Cluster Cache Directory-to-Data Elements Ratio	32
LRDP : Length of the Redo Pool	33
MXCANCEL: Timeout Threshold for Canceled Peer Nucleus	34
MXCANCELWARN : Timeout Threshold for Canceled Peer Nucleus Warning	35
MXMSG: Timeout Threshold for Internucleus Command Processing	36
MXMSGWARN : Timeout Threshold for Internucleus Command Processing Warning	37
MXSTATUS : Member-Level XCF Status Monitoring Heartbeat Interval	38
MXWTOR : Self-Termination Operator Query Interval	39
NUCID : Adabas Cluster Nucleus ID	40
PROGRAM : Program to Run	41
7 Entire Net-Work Parameter Statements	43
Conventions	44
XCF NODE Statement	47
XCF DRIVER Statement	57

- XCF LINK Statement 61
- 8 Adabas Cluster Operator Commands 65
 - ADACOM Operator Commands 66
 - Adabas Cluster Nucleus Operator Commands 71
- 9 Entire Net-Work Operator Commands 109
 - Overview of Entire Net-WorkCommands for z/OS Environments 111
 - Command Descriptions 112
 - XCF Line Driver Commands 125
- 10 Entire Net-Work Diagnostic Utilities 131
 - NETPFIL1 Utility 132
 - NETPFIL2 Utility 134
 - Example JCL 136
- Index 137

1 Reference








This documentation provides reference information for Adabas initialization parameters, cluster operator commands as well as the Adabas Online System screens pertinent to Adabas Cluster Services.



This documentation provides information on the operator commands for ADACOM and for an Adabas cluster nucleus; it also describes the operator commands for Entire Net-Work and for its XCF line driver.

This documentation provides detailed information on the initialization parameters needed for an Adabas Cluster environment.

It also describes the parameter statements required for enabling Entire Net-Work to communicate with other nodes.

This document is organized as follows:

 <i>Initialization Parameters</i>	Describes the initialization parameters needed for an Adabas Cluster Services environment.
 <i>ADACOM Initialization Parameters</i>	Describes the ADACOM initialization parameters needed for an Adabas Cluster Services environment.
 <i>ADANUC Initialization Parameters for Cluster Nuclei</i>	Describes the ADARUN parameters needed for each nucleus in a cluster.
 <i>Entire Net-Work (NETWRK) Initialization Parameters</i>	Describes the Entire Net-Work parameters needed for each nucleus in a cluster.
 <i>Useful ADARUN Parameters</i>	Provides a reference for ADARUN parameters useful to Adabas Cluster Services processing.
 <i>Entire Net-Work Parameter Statements</i>	Describes the initialization parameters needed for Entire Net-Work.
 <i>Adabas Cluster Operator Commands</i>	Describes operator commands for ADACOM and for an Adabas Cluster Services cluster nucleus.

 <i>Entire Net-Work Operator Commands</i>	Describes operator commands for Entire Net-Work.
 <i>Entire Net-Work Diagnostic Utilities</i>	Describes the Entire Net-Work diagnostic utilities, provided in the Entire Net-Work load library.





2 Initialization Parameters

This documentation provides detailed information on the initialization parameters needed for an Adabas cluster environment. It also describes the parameters required for enabling Entire Net-Work to in cluster environments.

ADARUN parameters are used to customize the Adabas environment. For Adabas Cluster Services environments, three types of initialization parameters must be specified, based on the setting of the ADARUN **PROGRAM** parameter:

- To run an Adabas nucleus, the name must be ADANUC.
- To set up an Entire Net-Work node, the name must be NETWRK.
- To set up an ADACOM task, the name must be ADACOM (no other ADARUN parameters are recognized for the ADACOM program run).

The chapter is organized in the following parts:

 <i>ADACOM Initialization Parameters</i>	Describes the ADACOM initialization parameters required by Adabas Cluster Services.
 <i>About ADARUN Parameters for Cluster Nuclei</i>	Describes the Adabas nucleus (ADANUC) initialization parameters required by Adabas Cluster Services.
 <i>Entire Net-Work (NETWRK) Initialization Parameter</i>	Describes the Entire Net-Work (NETWRK) initialization parameters for Entire Net-Work in Adabas Cluster Services environments.
 <i>Useful ADARUN Parameters</i>	Provides a reference for all of the ADARUN parameters useful to Adabas Cluster Services processing.

3

ADACOM Initialization Parameters

■ CDMGR - ADACOM Use	6
■ DBID - Database Identification	7
■ FORCE - Force Restart after Abnormal Termination	7
■ NU - Number of User Table Elements	8
■ SVC - Interregion Communication Number	9

An unlimited number of Adabas nucleus clusters, each with up to 32 members sharing a common database, can be defined per parallel sysplex.

The ADACOM initialization task:

- must be run on any operating system image that has cluster users but no cluster nuclei to set up the environment.
- can optionally be used to monitor and control the nuclei of one or more sysplex clusters.

Parameters specify the SVC/DBID combinations (sets) that the ADACOM is to manage. The DBID identifies the external physical database shared by a particular cluster of nuclei and known to the application. The SVC value may vary within a cluster and the same SVC may be used for different clusters. Other ADACOM parameters are discussed in the following sections.

Although a single ADACOM job can run all SVC/DBID sets in a sysplex environment, it is possible to run multiple ADACOM tasks simultaneously with the same, mixed, or completely different SVC/DBID sets. A subtask is attached to the ADACOM task for each SVC/DBID set occurring in the input of ADACOM. All work pertaining to an SVC/DBID set is performed by the subtask.

The COMPRINT data set prints global messages that apply to all SVC/DBID sets defined to an ADACOM task. In addition, a SYSOUT data set is dynamically allocated for command output to each SVC/DBID set. The format of this data set DD-name is *Pssddddd* where *ss* is the last two digits of the SVC and *ddddd* is the DBID.

For a sample job to run ADACOM, see the section *Create a Startup Procedure for ADACOM*.

CMDMGR - ADACOM Use

Parameter	Specify...	Possible Values	Default
<u>C</u> MDMGR	whether ADACOM remains in use as a command manager after initialization or quiesces.	<u>Y</u> ES <u>N</u> O	YES

The CMDMGR parameter specifies whether the ADACOM task remains in operation as a command manager for the cluster nuclei (CMDMGR=YES) or whether it runs like a batch job and terminates after processing the commands given in DD/KARTE (CMDMGR=NO).

DBID - Database Identification

Parameter	Specify...	Minimum	Maximum	Default
DBID	the database to be used.	1	65000	none

The DBID parameter designates the ID of an Adabas cluster's external physical database; that is, the database number that the user programs address to send commands to the single physical database of an Adabas nucleus cluster.

This number must be unique among all Adabas database IDs, NUCIDs, Natural buffer pool IDs, Entire Net-Work target IDs (suggested range 65482-65535), and internal target IDs used by Adabas Cluster Services (restricted range 65001-65481).

FORCE - Force Restart after Abnormal Termination

Parameter	Specify...	Possible Values	Default
FORCE	whether ADACOM forces a restart after an abnormal termination.	<u>Y</u> ES <u>N</u> O	NO

If Adabas believes there are still active nuclei, ADACOMs, or users on an image after a termination, an NU parameter value that is different from the NU value already in effect will not be recognized on restart. If you are certain that Adabas is wrong in its belief, you can use FORCE=YES to force a clean restart.

Note, however, that if a sysplex cluster nucleus or an ADACOM with the same SVC number is active on the image where FORCE=YES is used, it will crash.



Notes:

1. FORCE=YES works only when the NU parameter value is being changed and has the effect of re-setting the environment.
2. The nucleus ADARUN parameter FORCE is neither seen nor used by the Adabas Cluster Services SVCCLU. The ADACOM parameter FORCE may only be used to change the NU that is valid for the cluster.
3. If you use the ADACOM FORCE parameter, remember to remove it from ADACOM after you have reset the environment.

NU - Number of User Table Elements

Parameter	Specify...	Minimum	Maximum	Default
<u>NU</u>	the number of users that can be active in parallel on sysplex cluster nuclei in the local image.	20	16 777 215	200

In the extended CSA (ECSA), the SVCCLU maintains a user table with entries (UTES) containing information about every active user in the sysplex cluster nuclei on the local image. A UTE is assigned when a user issues an `OP` command or (if the user did not issue an `OP` command) at the first Adabas command. The UTE is released when the user issues a `CL` command or when the assigned cluster nucleus otherwise deletes the user's UQE (e.g., due to timeout).

The ADACOM `NU` parameter specifies the number of concurrent users (UTES) allowed for all cluster nuclei on the local operating system image. The first cluster nucleus or ADACOM started governs the value for `NU`: different values set for subsequent nuclei or ADACOMs are ignored.

To manually change the `NU` value, you must stop all sysplex cluster nuclei and ADACOM initialization tasks (if used) in the local image that have the same SVC number, modify the `NU` value for all the nucleus and ADACOM jobs, and then restart.

When the `NU` parameter is set to zero, any space allocated for the user table is freed, the Adabas Cluster Services control block (PLXCB) is freed, and the ADACOM subtask for this SVC/DBID set terminates, regardless of the CMDMGR parameter setting. However, if the ADACOM subtask believes that there are active nuclei or users on this database, and `FORCE=YES` is not used, no action is taken and the subtask terminates abnormally with user ABEND 40 (hex). The ADACOM main task and any subtasks handling other SVC/DBID sets are not affected.

Example:

Run the Adabas cluster nuclei on the local image with 500 elements in the user table.

```
NU=500
```

SVC - Interregion Communication Number

Parameter	Specify...	Possible Values	Default
SVC	the supervisor call number to be used for the Adabas Cluster Services session.	200-255	249

The `svc` is used to perform various Adabas internal functions; the number is used to communicate between the users and the database.

The `svc` number is specified as an integer and must correspond to the number used for the version 8.1 Adabas SVC (ADASVC). Valid SVC values are 200-255.

Example:

Execute an Adabas Cluster Services session under an OS/390 or z/OS image using SVC 202.

```
SVC=202
```


4 About ADARUN Parameters for Cluster Nuclei

■ Global ADARUN Parameters	12
■ ADARUN Parameter Usage in Cluster Environments	13

Software AG recommends that you use your existing values (or the default settings) of the Adabas ADARUN parameters for each Adabas nucleus in a cluster, and then tune the values after analyzing the performance of the cluster. For more information about expected differences, read *Performance and Tuning*.

Session statistics can be used to determine the best settings for each parameter. The statistics are printed automatically at the end of a session, but can also be displayed using nucleus or ADACOM operator commands during the session.

For your convenience, ADARUN parameters that are most significant to Adabas Cluster Services usage are described in [Useful ADARUN Parameters](#).

This chapter covers the following topics:

Global ADARUN Parameters

ADARUN parameters that must be the same for all nuclei in the cluster are called *global*.

Some global parameters are set at nucleus startup and cannot be changed during the ensuing session; other global parameters can be changed during a session:

- When you set a value for a global parameter that cannot be changed after the first nucleus in an Adabas cluster has started, intracuster messages are used to communicate the ADARUN parameter settings of the first nucleus to all subsequent cluster nuclei. Each subsequent nucleus receives this information during initialization and determines whether its global nonchangeable parameters are equal to those of the first nucleus.

If they are not equal, the nucleus fails with a parameter error. The nonequal global changeable parameters are reset to the value retrieved from the intracuster messages and a corresponding message is printed.

- If you change the value of a global parameter that can be changed during a running session, the nucleus on which you make the change acquires a "parameter change lock", makes the changes in its local parameter area, and communicates the changes to the other cluster nuclei using intracuster messages.

All other nuclei in the cluster receive the intracuster messages containing the global parameters that have changed, change the parameters in their local parameter area, and send an "acknowledge" message.

ADARUN Parameter Usage in Cluster Environments

A cluster nucleus makes use of:

- *global* parameters, whose values are enforced by Adabas Cluster Services to be equal for all nuclei in a cluster. Some of these parameters can be modified (globally modifiable) during a session using an operator command or the Adabas Online System (`NISNHQ`, `NONDES`, and `AOSLOG` are only modifiable using AOS); others are fixed (globally fixed) and cannot be modified while the nuclei in the cluster are running.
- *local* parameters, which can be different for each nucleus. Some of these parameters are modifiable (locally modifiable) using an operator command or the Adabas Online System; others are fixed (locally fixed) and cannot be modified.

A few Adabas ADARUN parameters are not available to a cluster nuclei (No).

In the following table, the "No" column indicates which ADARUN parameters are not available to a cluster nucleus, the "LF" column indicates whether the parameter is a locally fixed parameter, the "LM" column indicates whether the parameter is a locally modifiable parameter, the "GF" column indicates whether the parameter is a globally fixed parameter, and the "GM" column indicates whether the parameter is a globally modifiable parameter.

Parameter	Usage	No	LF	LM	GF	GM
<code>AOSLOG</code>	Log to DDPRINT commands issued by AOS or ADADBS OPERCOM that modify the active nucleus			X		
<code>AREXCLUDE</code>	Exclude file(s) from autorestart		X			
<code>ARMNAME</code>	Name used to activate ARM		X			
<code>ASSOCACHE</code>	Controller caching control for the Associator component		X			
<code>ASYTVS</code>	Asynchronous buffer flush based on volser			X		
<code>CACHE</code>	Load ADACSH (Adabas Caching Facility)		X			
<code>CACTIVATE</code>	RABN range activation (Adabas Caching Facility)			X		
<code>CASSODSP</code>	Associator RABNs cached for data space (Adabas Caching Facility)			X		
<code>CASSOEXT</code>	Associator RABNs cached for extended memory (Adabas Caching Facility)			X		
<code>CASSOHSP</code>	Associator RABNs cached for hiperspace (Adabas Caching Facility)			X		
<code>CASSOV64</code>	Associator RABNs cached for virtual 64 (Adabas Caching Facility)			X		
<code>CASSOMAXS</code>	Associator cache space unit size (Adabas Caching Facility)			X		

Parameter	Usage	No	LF	LM	GF	GM
CBUFNO	Read buffer count for concurrent I/O (Adabas Caching Facility)			X		
CCTIMEOUT	Cache space area inactivity time (Adabas Caching Facility)			X		
CDATADSP	Data Storage RABNs cached for data space (Adabas Caching Facility)			X		
CDATAEXT	Data Storage RABNs cached for extended memory (Adabas Caching Facility)			X		
CDATAHSP	Data Storage RABNs cached for hiperspace (Adabas Caching Facility)			X		
CDATAV64	Data Storage RABNs cached for virtual 64 (Adabas Caching Facility)			X		
CDATAMAXS	Data Storage cache space unit size (Adabas Caching Facility)			X		
CDEMAND	Lowest acceptable Adabas buffer efficiency level (Adabas Caching Facility)			X		
CDISPSTAT	RABN range statistic display location (Adabas Caching Facility)			X		
CDXnn	Collation descriptor user exit(s)				X	
CEXCLUDE	Excluded command types (Adabas Caching Facility)			X		
CFILE	File or file range to be cached (Adabas Caching Facility)			X		
CLOGDEV	Multiple command log device		X			
CLOGLAYOUT	Define command log format		X			
CLOGMRG	Automatic command log merge control in a cluster environment					X
CLOGSIZE	Multiple command log size (blocks)		X			
CLUCACHENAME	Cluster cache structure name (Adabas Cluster Services only)				X	
CLUCACHESIZE	Cluster cache area size (Parallel Services only)				X	
CLUGROUPNAME	Cluster group name (Adabas Cluster Services only)				X	
CLULOCKNAME	Cluster lock structure name (Adabas Cluster Services only)				X	
CLULOCKSIZE	Cluster lock area size (Parallel Services only)				X	
CLUSTER	Adabas cluster session control				X	
CMAXCSPS	Storage area count for ADACSH (Adabas Caching Facility)			X		
CRETRY	Cache space retry interval (Adabas Caching Facility)			X		
CSTORAGE	RABN caching activation (Adabas Caching Facility)			X		
CT	Command time limit (seconds)					X

Parameter	Usage	No	LF	LM	GF	GM
CWORKSTORAGE	Cache space type (Adabas Caching Facility)			X		
CWORK2FAC	Percentage of Work part 2 cached (Adabas Caching Facility)			X		
CWORK3FAC	Percentage of Work part 3 cached (Adabas Caching Facility)			X		
CXFILE	Excluded files (Adabas Caching Facility)			X		
DATACACHE	Controller caching control for the Data Storage component		X			
DIRRATIO/ELEMENTRATIO	Ratio of directory entries to data elements in a cluster cache structure/area				X	
DBID	Database ID (physical)				X	
DEVICE	Device type of the first ASSO extent				X	
DSF	Delta Save Facility control				X	
DSFEX1	Delta Save Facility user exit				X	
DTP *	Distributed transaction processing control	*			X	
DUALCLD	Dual command log device		X			
DUALCLS	Dual command log size (blocks)		X			
DUALPLD	Dual protection log device		X			
DUALPLS	Dual protection log size (blocks)		X			
FASTPATH	Adabas Fastpath control				X	
FMXIO	Limit parallel I/O operations by LFIOP flush processing			X		
FORCE	Overwrite IDTE		X			
HEXnn	Hyperdescriptor exit(s)				X	
IGNDIB	Ignore DIB entry		X			
IGNDTP	Ignore distributed transaction processing area (Work part 4)		X			
INTNAS	Interval between nucleus statistic checkpoints (SYNS 60)				X	
LBP	Length of buffer pool		X			
LCP	Length of security pool		X			
LDEUQP	Length of unique (UQ) descriptor pool		X			
LDTP	Length of distributed transaction processing area (Work part 4)		X			
LFIOP	Length of asynchronous flush pool		X			
LFP	Length of internal format buffer pool		X			
LI	Length of ISN list table (TBI)		X			
LOCAL	Nucleus (cluster) unknown to the network				X	
LOGCB	Log control block			X		

Parameter	Usage	No	LF	LM	GF	GM
<u>LOGCLEX</u>	Log command log extension (CLEX)			X		
<u>LOGFB</u>	Log format buffer			X		
<u>LOGGING</u>	Logging of Adabas commands			X		
<u>LOGIB</u>	Log ISN buffer			X		
<u>LOGIO</u>	Log I/O activity			X		
<u>LOGRB</u>	Log record buffer			X		
<u>LOGSB</u>	Log search buffer			X		
<u>LOGSIZE</u>	Maximum command log size		X			
<u>LOGUX</u>	Log user exit B data			X		
<u>LOGVB</u>	Log value buffer			X		
<u>LP</u>	Length of data protection area (Work part 1)		X			
<u>LQ</u>	Length of sequential command table		X			
<u>LRDP</u>	Length of the redo pool in cluster environments.		X			
<u>LS</u>	Length of sort area		X			
<u>LU</u>	Length of intermediate user buffer				X	
<u>LWKP2</u>	Length of ISN list processing area (Work part 2)		X			
<u>LWP</u>	Length of Adabas work pool		X			
<u>MODE</u>	Mode of operation	X				
<u>MSGBUF</u>	Size of the message buffer		X			
<u>MXCANCEL</u>	Timeout threshold for a cancel request between cluster nuclei(Adabas Cluster Services and Adabas Parallel Services)			X		
<u>MXCANCELWARN</u>	Timeout threshold for a cancel request warning between cluster nuclei (Adabas Cluster Services and Adabas Parallel Services)			X		
<u>MXMSG</u>	Maximum message reply time between cluster nuclei (Adabas Cluster Services and Adabas Parallel Services)			X		
<u>MXMSGWARN</u>	Timeout threshold for a message reply warning between cluster nuclei (Adabas Cluster Services and Adabas Parallel Services)			X		
<u>MXSTATUS</u>	XCF status monitoring heartbeat interval (Adabas Cluster Services)			X		
<u>MXTNA</u>	Maximum inactivity time limit override for a user				X	
<u>MXTSX</u>	Maximum Sx execution time limit override for a user				X	
<u>MXTT</u>	Maximum transaction time limit override for a user				X	
<u>MXWTOR</u>	Self-termination query operator response interval (Adabas Cluster Services and Adabas Parallel Services)			X		
<u>NAB</u>	Number of attached buffers		X			

Parameter	Usage	No	LF	LM	GF	GM
NC	Number of command queue elements		X			
NCLOG	Number of command logs		X			
NH	Number of hold queue elements		X			
NISNHQ	Number of ISNs in hold queue for user					X
NPLOG	Number of protection logs		X			
NONDES	Non-descriptor searches					X
NQCID	Number of active command IDs per user					X
NSISN	Number of ISNs per ISN table element		X			
NT	Number of threads		X			
NU	Number of user queue elements		X			
NUCID	Custer nucleus ID		X			
OPENRQ	Open command required				X	
PLOGDEV	Multiple protection log device		X			
PLOGRQ	Protection log required				X	
PLOGSIZE	Multiple protection log size (blocks)		X			
PREFETCH**	Prefetch/multifetch feature control (see note below)	X				
PREFICMD**	Include command from prefetch/multifetch (see note below)	X				
PREFIFIL**	Include file from prefetch/multifetch (see note below)	X				
PREFNREC**	Multifetch record count (see note below)	X				
PREFSBL**	Prefetch single buffer length (see note below)	X				
PREFTBL**	Prefetch total buffer length (see note below)	X				
PREFXCMD**	Exclude command from prefetch/multifetch (see note below)	X				
PREFXFIL**	Exclude file from prefetch/multifetch (see note below)	X				
QBLKSIZE	Sequential data set block size (optimized by ADAIOR)		X			
READONLY	Read-only session control	X				
REVIEW	Adabas Review control			X		
SMGT	Error handling (PIN) facility control		X			
SORTCACHE	Controller caching control for the Adabas sort area component		X			
SPT	Adabas triggers and stored procedures control				X	
SVC	SVC number		X			
TCPIP	TCP/IP access control			X		
TCPURL	TCP/IP universal resource locator (URL)			X		
TEMPCACHE	Controller caching for the Adabas temp area component		X			
TFLUSH	Synchronous buffer flush time	X				

Parameter	Usage	No	LF	LM	GF	GM
TLSCMD	Time limit for S1, S2, and S4 complex searches (seconds)				X	
TNAA	Non-activity time limit (access-only users)					X
TNAE	Non-activity time limit (ET logic users)					X
TNAX	Non-activity time limit (exclusive update users)					X
TT	Transaction time limit					X
UEXn	User exits: 1, 3, 4, 5, 8		X			
UEXn	User exits: 2, 12				X	
UEXn	User exits: 6, 9 (for utilities)	X				
UTIONLY	Utilities-only session				X	
VISTA	Adabas Vista control				X	
WORKCACHE	Controller caching for the Adabas work area component		X			

**Notes:**

1. * Adabas Cluster Services 8.1 does not support DTP=TM, but it does support DTP=RM.
2. ** The PREFxxx parameters are used with application programs (PROGRAM=USER) making Adabas calls. They have no effect when specified for an Adabas nucleus.

5

Entire Net-Work (NETWRK) Initialization Parameters

▪ Guidelines	21
▪ CT - Command Timeout Limit	21
▪ FORCE - Overwrite ID Table Entry	22
▪ LU - Length of Intermediate User Buffer	23
▪ NAB - Number of Attached Buffers	23
▪ NC - Number of Request Queue Elements	24
▪ SVC - Interregion Communication Number	24
▪ TARGETID - Entire Net-Work Target Identification	25

The ADARUN parameters for Entire Net-Work specify all interregion communication information for the node. They are specified with ADARUN PROGRAM=NETWRK.

Software AG recommends that you use your existing values (or the default settings) of the ADARUN parameters for each Entire Net-Work node, and then tune the values after analyzing the performance of the node. See the *Performance and Tuning* documentation for information about expected differences.

The following table summarizes the Entire Net-Work ADARUN parameters:

Parameter	Use	Minimum	Maximum	Default
CT	Command timeout limit (sec.)	1	16777215	60
FORCE	Overwrite target ID entry	-	-	NO
LU	User buffer size	4000	65535	65535
NAB	Number of attached buffers	3	49150	16
NC	Number of request queue elements	20	32767	200
PROGRAM	Program to be executed	NETWRK	NETWRK	(none)
SVC	SVC number	200	255	249
TARGETID	Entire Net-Work target ID	1*	65535	1

* Software AG strongly recommends that you restrict Entire Net-Work target IDs in a sysplex cluster environment to values in the range 65482 through 65535 to avoid interference with the target IDs used for other cluster requirements.

Following is an example of an ADARUN statement for Entire Net-Work:

```
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.

Guidelines

Use the following guidelines when specifying ADARUN session parameters:

- Ensure that the correct program to be executed is specified (`PROG=NETWRK`).
- Ensure that the correct target ID is specified (`TARGETID` parameter).
- Determine which settings for the following parameters are applicable for the session:

FORCE	(override active target ID)
SVC	(Adabas SVC number)

CT - Command Timeout Limit

Parameter	Specify...	Minimum	Maximum	Default
CT	the time allowed for a user program to obtain results from Entire Net-Work.	1	16777215	60

The maximum number of seconds (more precisely, units of 1.048576 seconds) that can elapse from the time an Entire Net-Work request is completed until the results are retrieved by the user through the interregion communication ("16-call"/operating system-dependent).

This parameter is used to prevent a request queue element (RQE) and the attached buffer from being held indefinitely if a user with an outstanding Entire Net-Work request terminates abnormally.

Possible causes for a command timeout are

- non-dispatchable or swapped-out user address space;
- user canceled;
- low user priority in a high-activity system.

If the CT limit is exceeded, the RQE and the attached buffer are released and message NETM93 is printed. If the user attempts to access the RQE after the timeout has occurred, response code 254 is returned.

Example:

Execute an Entire Net-Work task; set the maximum permitted time to obtain the result of a command using interregion communication to approximately 30 seconds.

```
ADARUN  PROG=NETWRK , CT=30
```

FORCE - Overwrite ID Table Entry

Parameter	Specify...	Possible Values	Default
FORCE	whether to overwrite the target ID table entry.	YES NO	NO

By default (FORCE=NO), error NETM98 occurs if the TARGETID parameter (see [TARGETID](#)) specifies a target ID that is already in the target ID table:

```
NETM98 targetid TARGET INITIALIZATION ERROR: DUP ID (LOCAL)
```

FORCE=NO prevents the table entry from being overwritten. FORCE=YES allows the existing target ID table entry to be overwritten by the value specified for the TARGETID= parameter.

When a target terminates, its ID table entry is normally removed from the ID table. However, if an Entire Net-Work task has terminated abnormally without the recovery mechanism receiving control, its entry can remain in the ID table. FORCE=YES permits an Entire Net-Work target to be restarted even though its ID is already present in the ID table.

In addition to Entire Net-Work, other Software AG products such as Adabas or Entire System Server (formerly Natural Process) may have entries in the ID table.

Because FORCE=YES replaces *any* entry that matches the specified TARGETID value, you must ensure that the ID table does not contain that same value for any target that is active.



Caution: If you were to mistakenly override the ID table entry of an active target, it could severely disrupt the operation of the affected service system.

For this reason, all target and database IDs must be unique, and FORCE should be issued *only* after you determine that no other target with the ID specified by TARGETID is active.

Example:

Execute an Entire Net-Work task.

If the ID table contains an entry for target ID 3333, overwrite it; that is, allow the new task to use ID table entry 3333. Disconnect any existing task with an ID of 3333 from all user access.

```
ADARUN PROG=NETWRK, FORCE=YES, TARGETID=3333
```

If FORCE=NO (the default), Entire Net-Work does not permit the use of target ID 3333 if the ID table already contains an entry for that ID. The message NETM98 is issued, indicating that a target using the ID 3333 is already active.

LU - Length of Intermediate User Buffer

Parameter	Specify...	Minimum	Maximum	Default
LU	the size of the intermediate user buffer.	4000	65535	65535

The size specified must be large enough to accommodate all Adabas control information (108 bytes), all user buffers (format, record, search, value, ISN) that may be required for any given Adabas command, and any user information passed from Adabas link routines to nucleus user exits.

If the 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.

An error occurs if the LU parameter specifies a value greater than the byte count implied by the NAB (number of attached buffers) parameter. The NAB default of 16 allocates more than 64 kilobytes.

Example:

Start an Entire Net-Work session using an Adabas intermediate user buffer area of 20,000 bytes.

```
ADARUN  PROG=NETWRK, LU=20000
```

NAB - Number of Attached Buffers

Parameter	Specify...	Minimum	Maximum	Default
NAB	the number of attached buffers to be used.	3	49150	16

NAB is an optional parameter that defines the number of attached buffers to be used during the Entire Net-Work session. Entire Net-Work allocates an attached buffer pool whose size equals the NAB value multiplied by 4112.

An attached buffer is an internal buffer used for interregion communication. Entire Net-Work requires the use of attached buffers in all environments.



Note: A value of less than 16 prevents some Adabas utility functions from being performed remotely.

Example:

Execute an Entire Net-Work session; use 50 attached buffers.

```
ADARUN  PROG=NETWRK, NAB=50
```

NC - Number of Request Queue Elements

Parameter	Specify...	Minimum	Maximum	Default
NC	the number of elements allowed in the Entire Net-Work request queue.	20	32767	200

NC is an optional parameter that specifies the maximum number of user requests that can be processed simultaneously by this Entire Net-Work communicator. Any user requests originating on this node that cause the NC value to be exceeded receive response code 151.

Each user request originating from this node is assigned a request queue element (RQE), which exists for the duration of the request. The RQE is released when the user has either received the results of this request from Entire Net-Work, or has been timed out. Each RQE requires 192 bytes.

Example:

Execute an Entire Net-Work session; set the request queue to contain as many as 500 elements at one time.

```
ADARUN  PROG=NETWRK , NC=500
```

SVC - Interregion Communication Number

Parameter	Specify...	Possible Values	Default
SVC	the number of the SVC to be used.	200-255	249

The number of the SVC to be used for the Entire Net-Work session. The SVC number is specified as an integer. The SVC is used to perform various internal functions, especially the processing of interregion communication.

svc is an optional parameter that defines the interregion communications supervisor call (SVC) to be used for this session.

The svc number specified must correspond to that which has been installed as the Software AG communications svc at the user installation. The same svc may be used for Adabas, Adabas Cluster Services, Entire Net-Work, and all other Software AG products using interregion communication.

Example:

Execute an Entire Net-Work session using SVC 205.

```
ADARUN  PROG=NETWRK , SVC=205
```

TARGETID - Entire Net-Work Target Identification

Parameter	Specify...	Minimum	Maximum	Default
TARGETID	the ID for an Entire Net-Work node.	1*	65535	1

* Software AG strongly recommends that you restrict Entire Net-Work target IDs in a sysplex cluster environment to values in the range 65482 through 65535 to avoid interference with the target IDs used for other cluster requirements.

The TARGETID parameter value must uniquely identify a particular Entire Net-Work node. Each target (node) ID must be unique, and can be any valid decimal value ranging from 1 (the default) to 65535. If the default is used for one node, it cannot be used for any other node and thus, this parameter must be specified for all other nodes.

In a sysplex cluster environment, all NUCIDs, Adabas database IDs (DBIDs), internally assigned operating system image IDs, target IDs used by Entire Net-Work, Natural global buffer pool IDs, 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. An exception to this rule allows databases to be specified as "local" targets as explained in [Special Exception \(Local\)](#).

Example:

Execute an Entire Net-Work session using target ID 65485.

```
ADARUN PROG=NETWRK , TARGETID=65485
```

This section covers the following topics:

- [Uniqueness Required \(Global\)](#)
- [Special Exception \(Local\)](#)

Uniqueness Required (Global)

Entire Net-Work normally enforces unique target IDs throughout the network, that is, across all connected nodes. A target is not permitted to start if another target with the same ID is active anywhere in the network. The ADAM98 message (target initialization error) will result.

When a link is established between two previously unconnected partial networks, it is possible that the same target ID is used in both parts simultaneously. Because target IDs must be unique within the network, this is considered an error situation.

Entire Net-Work cannot determine which of the targets should be deactivated. Therefore the link between the two parts of the network is not established and explanatory messages (NET0108 and NET0106) are written to the operator consoles of both nodes trying to connect.

It is the responsibility of the installation to terminate one of the conflicting targets. The link can then be connected from either side by using the CONNECT link operator command.

Special Exception (Local)

The Adabas ADARUN parameter `LOCAL` permits the use of duplicate target IDs under special circumstances. A target ID can be used on all nodes in the network simultaneously if *all* targets with that ID are declared `LOCAL=YES` (see the *Adabas Operations* documentation for the appropriate ADARUN parameter). In this case, any one of the targets with that ID can be accessed *only* by users running on the same node.

Entire Net-Work does not transport requests to `LOCAL` targets. Response code 148 is generated if a user program calls a target declared `LOCAL` from a remote node.

A target ID cannot be used simultaneously for both a local target and a global target. The reason for this restriction is illustrated in the following example.

Imagine that a (global) database, DBID 35, is active on node PROD, and user programs from all nodes in the network are issuing calls to it. If a local database, also with DBID 35, starts on node TEST, it immediately draws all calls that originate on node TEST to the local database DBID 35. Users on other nodes are not affected, but all users on node TEST are switched in mid-transaction from the global database DBID 35 to the local database DBID 35, a different and probably totally unrelated database. This error situation cannot be permitted.

Entire Net-Work enforces this rule and thus prevents the confusion that would result if a local target ID coincided with a global target ID by broadcasting information about local targets to all nodes the same way it does for global targets.

6 Useful ADARUN Parameters

■ CLOGMRG : Cluster Command Log Merge	28
■ CLUCACHENAME : Cluster Cache Structure/Area Name	29
■ CLUGROUPNAME : Cluster Group Name	30
■ CLULOCKNAME : Cluster Lock Structure/Area Name	30
■ CLUSTER : Adabas Nucleus Cluster Session Control	31
■ DIRRATIO / ELEMENTRATIO : Cluster Cache Directory-to-Data Elements Ratio	32
■ LRDP : Length of the Redo Pool	33
■ MXCANCEL: Timeout Threshold for Canceled Peer Nucleus	34
■ MXCANCELWARN : Timeout Threshold for Canceled Peer Nucleus Warning	35
■ MXMSG: Timeout Threshold for Internucleus Command Processing	36
■ MXMSGWARN : Timeout Threshold for Internucleus Command Processing Warning	37
■ MXSTATUS : Member-Level XCF Status Monitoring Heartbeat Interval	38
■ MXWTOR : Self-Termination Operator Query Interval	39
■ NUCID : Adabas Cluster Nucleus ID	40
■ PROGRAM : Program to Run	41

This chapter provides a reference for the ADARUN parameters useful to Adabas Cluster Services processing. For information on how to specify ADARUN parameters, read the *Adabas Operations* documentation provided with Adabas.

CLOGMRG : Cluster Command Log Merge

Parameter	Specify . . .	Possible Values	Default
CLOGMRG	whether the Adabas cluster is to run with automatic CLOG merge.	YES NO	NO

An automatic CLOG merge process provides a single cluster-wide command log containing, in chronological order, all Adabas commands executed by any of the cluster nuclei in the time period covered by the log.

CLOGMRG specifies whether the Adabas cluster is to run with the automatic CLOG merge:

Value	Action
YES	enables an automated process that copies dual or multiple command logs to sequential data sets and merges those data sets across a cluster into a single cluster-wide command log. This process automatically merges command logs when an ADARES CLCOPY is submitted from user exit 2 and executed. The cluster nuclei and the ADARES utility perform coordinated CLOG switches and CLCOPY functions with merging of CLOG records, much the same way the PLOG merge process works.
NO	(the default), each nucleus produces its own CLOG, independent of any other nucleus in the cluster; dual or multiple CLOGs can be manually copied using ADARES CLCOPY to a sequential data set; and the resulting sequential CLOGs can be manually merged across a cluster using the ADARES MERGE CLOG function.

CLOGMRG is a global (that is, a cluster-wide) parameter, which means that the setting of this parameter in the first cluster nucleus to become active is propagated to all nuclei that subsequently become active.

Parameter Dependencies

It is possible to specify both LOGGING=NO and CLOGMRG=YES. In a cluster environment, it is also possible that a CLOG will be written to even if LOGGING=NO. As long as CLOGMRG=NO, nothing will be written to the CLOG if LOGGING=NO. However, if LOGGING=NO and CLOGMRG=YES, control records necessary for ADARES CLCOPY will be written to the CLOG.

CLOGMRG is effective only in an Adabas cluster environment; that is, when CLUSTER is specified with a value other than NO and NUCID is specified with a nonzero value.

Dynamic Modification

The setting of the CLOGMRG parameter can be changed dynamically using the CLOGMRG command from the operator console, the ADADBS OPERCOM CLOGMRG function, or the Modify Parameter function of Adabas Online System.

Because CLOGMRG is a global parameter, the change request to one nucleus is automatically propagated to all nuclei in the cluster.

CLUCACHENAME : Cluster Cache Structure/Area Name

Parameter	Specify . . .	Possible Values	Default
CLUCACHENAME	the name of the cache structure/area of the Adabas cluster in which the current nucleus participates.	text	none

For Adabus Cluster Services, the CLUCACHENAME identifies the particular sysplex cache structure (defined in the CFRM policy) associated with a particular Adabas sysplex nucleus cluster. The same name must be set in each nucleus in the cluster. CLUCACHENAME must be unique for each cluster; that is, each database.

Adabas Parallel Services automatically creates a default name for each global cache area allocated by ADACOM. However, if you wish to control the name of the global cache area, you can set this parameter.



Note: Software AG recommends that you allow the system to build the default name.

For Adabas Parallel Services, the CLUCACHENAME identifies a particular global cache area associated with all cluster nuclei that have the same SVC/DBID set as the current nucleus. The same name must be set in each cluster nucleus that has the same SVC/DBID set. CLUCACHENAME must be unique for each SVC/DBID set.

In all cases, the name can be 1-16 characters long. Valid characters are numeric, uppercase alphabetic, the special characters \$, @, and #. The name must begin with an alphabetic character (A-Z). IBM names begin with SYS, UNDESIG, or the characters A through I (these characters are not reserved).



Note: Additionally, the special character underscore (_) is allowed in Adabus Cluster Services; that is, sysplex environments.

Example

In the following example, the sysplex cache structure name is defined as "SAG1SMP#CACHE1".

```
ADARUN PROG=ADANUC , CLUCACHENAME=SAG1SMP#CACHE1
```

CLUGROUPNAME : Cluster Group Name

Parameter	Specify . . .	Possible Values	Default
<u>CLUGROUPNAME</u>	the name of the XCF messaging group in Adabus Cluster Services.	text	none

The CLUGROUPNAME parameter provides:

- the XCF group name to be used by the sysplex cluster nuclei for their communication using XCF.
- the name to be used by the cluster nuclei for their communication.

All nuclei in the same cluster must have the same CLUGROUPNAME. Nuclei belonging to different clusters must have different CLUGROUPNAMEs.

In addition, the CLUGROUPNAME must be different from the XCF group names used by other subsystems that communicate using XCF.

The name can be 1-8 characters long. Valid characters are numeric, uppercase alphabetic, and the special characters \$, @, and #. The name must begin with an alphabetic character (A-Z). IBM names begin with SYS, UNDESIG, or the characters A through I (these characters are not reserved).

Example

The following examples specifies the name of the XCF messaging group as "SAG1CLU".

```
ADARUN  PROG=ADANUC , CLUGROUPNAME=SAG1CLU
```

CLULOCKNAME : Cluster Lock Structure/Area Name

Parameter	Specify . . .	Possible Values	Default
<u>CLULOCKNAME</u>	the name of the lock structure defined in the coupling facility's CFRM policy or the global lock area in which the current nucleus participates	text	none

For Adabus Cluster Services, CLULOCKNAM identifies a particular sysplex lock structure (defined in the CFRM policy) associated with a particular Adabas nucleus cluster. The same name must be set in each nucleus in the cluster. CLULOCKNAME must be unique for each cluster; that is, each database.

Adabas Parallel Services automatically creates a default name for each global lock area allocated by ADACOM. However, if you wish to control the name of the global lock area, you can set this parameter.



Note: Software AG recommends that you allow the system to build the default name.

For Adabas Parallel Services, the CLULOCKNAME identifies a particular global lock area associated with all Adabas Parallel Services cluster nuclei that have the same SVC/DBID set as the current nucleus. The same name must be set in each cluster nucleus that has the same SVC/DBID set. CLULOCKNAME must be unique for each SVC/DBID set.

In all cases, the name can be 1-16 characters long. Valid characters are numeric, uppercase alphabetic, the special characters \$, @, and #. The name must begin with an alphabetic character (A-Z). IBM names begin with SYS, UNDESIG, or the characters A through I (these characters are not reserved).



Note: Additionally, the special character underscore (_) is allowed in Adabus Cluster Services; that is, sysplex environments.

Example

The following example set the name of the lock structure or global lock area as "SAG1SMP#LOCK1".

```
ADARUN  PROG=ADANUC , CLULOCKNAME=SAG1SMP#LOCK1
```

CLUSTER : Adabas Nucleus Cluster Session Control

Parameter	Specify . . .	Possible Values	Default
CLUSTER	whether the Adabas nucleus will participate in a cluster of nuclei working on the same Adabas database, and, if so, the type of cluster (Adabas Parallel Services or Adabas Cluster Services) in which it will participate.	NO SYSPLEX LOCAL	NO

The CLUSTER parameter specifies whether the Adabas nucleus will participate in a cluster of nuclei working on the same Adabas database, and if so, which type of cluster:

- If CLUSTER=NO (the default), the nucleus will not participate in an Adabas cluster. If CLUSTER=NO, the NUCID parameter must be set (or default) to 0.
- If CLUSTER=LOCAL, the nucleus will participate in a cluster where all nuclei run on the same operating system image. The add-on product Adabas Parallel Services is required to run in this mode.
- If CLUSTER=SYSPLEX, the nucleus will participate in a cluster where the nuclei may run on different systems within a parallel sysplex. This is supported only for z/OS systems. The add-on product Adabas Cluster Services is required to run in this mode.

If CLUSTER=LOCAL or SYSPLEX, the NUCID parameter must be set to a nonzero value that identifies the individual cluster nucleus. Also, the parameter settings MODE=SINGLE,

READONLY=YES, LFIOP=0 and DTP=TM, and the use of sequential protection log datasets (DD-name/link name DDSIBA) are not allowed for cluster nuclei.

Example

The following example includes the nucleus in a cluster run with Adabas Cluster Services:

```
ADARUN DBID=53 , CLUSTER=SYSPLEX , NUCID=5301
```

DIRRATIO / ELEMENTRATIO : Cluster Cache Directory-to-Data Elements Ratio

The DIRRATIO and ELEMENTRATIO parameters together define the ratio of directory elements to data elements in the Adabas Cluster Services coupling facility cache structure or the Adabas Parallel Services global cache area.

This ratio influences the allocation of the cache structure/area when the first cluster nucleus starts. It has no influence on the cache structure/area when a subsequent nucleus starts.

Parameter	Specify . . .	Minimum	Maximum	Default
DIRRATIO	the directory portion of the directory-to-data element ratio	1	32767	4
ELEMENTRATIO	the element portion of the directory-to-data element ratio	1	32767	1

The DIRRATIO / ELEMENTRATIO ratio is optionally used in Adabas cluster environments to tune the cache structure/area for better performance.

Directory elements are used to keep track of the cluster members that have individual blocks in their local caches (buffer pool and Adabas Caching Facility caches), so that the blocks can be invalidated should other members modify them.

Data elements are used to hold the contents of Associator and Data blocks from the database.

If the number of directory elements is insufficient, existing directory elements are reused and the blocks associated with those directory elements are invalidated and can then no longer be reused. When blocks become invalid even though they have not been modified, they must be reread and registered again the next time they are referenced and validated.

It is generally better to reassign storage for data elements to keep more Associator and Data blocks in the cache structure/area than to define too many directory elements in the cache. More data elements than necessary can be used to keep additional blocks to improve the cache efficiency.

The number of directory elements need not be greater than the sum of the sizes of all buffer pools plus the amount of space represented by data elements (number of data elements times data element size), divided by the smallest Associator block size.

You can use the Adabas Online System cache structure size calculator described in *Estimating Sizes for the Cache Structure in a Cluster Environment* to determine appropriate global cache area sizes and DIRRATIO/ELEMENRATIO parameter values.

When connecting to the cache structure/area during startup, the ADAX57 message reports the number of directory and data elements allocated and the size of a data element.

LRDP : Length of the Redo Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LRDP	the size (in bytes) of the redo pool.	0; or 80000	2147483647	LFIOP value

The LRDP parameter specifies the size of the redo pool in an Adabas Cluster Services or Adabas Parallel Services nucleus. It has an effect only when CLUSTER=SYSPLEX or LOCAL has also been specified. Its default is the size of the flush I/O pool (LFIOP parameter) of the cluster nucleus.

- If LRDP is set to zero, the cluster nucleus performs immediate publishing: Updated database blocks are always written to the global cache at the time each update takes place.
- If LRDP is set to (or defaults to) a nonzero value, the cluster nucleus performs deferred publishing: The writing of updated database blocks to the global cache is deferred until later. However, updates are always published in the global cache before the transactions they belong to are committed (by ET commands). The redo pool is used to maintain descriptions of all unpublished updates, in case they need to be redone due to concurrent updates to the same blocks by other nuclei in the cluster.

By combining multiple updates to the same block into a single write-to-cache operation, deferred publishing tends to result in fewer read and write operations from and to the global cache than immediate publishing. Depending on the type of application workload, this may have a large positive effect on performance. This is especially the case for Adabas Cluster Services, where frequent communication with the cache in the coupling facility may cause a high overhead.

Recommendation: For Adabas Cluster Services, specify a nonzero LRDP parameter (or use the default).

Different nuclei in the same cluster can have different settings of LRDP. It is also possible, although not recommended, to run one nucleus with LRDP=0 and another one with LRDP>0.

If the redo pool specification is too small and the pool runs full, the nucleus will write additional updated blocks to the global cache before the surrounding transactions end. Such additional cache writes may hurt system performance.

The nucleus reports on the usage (high watermark) of the redo pool in a shutdown statistic and in the response to the DRES operator command or the corresponding AOS or ADADBS OPERCOM functions.

MXCANCEL: Timeout Threshold for Canceled Peer Nucleus

Parameter	Specify . . .	Minimum	Maximum	Default
<u>MXCANCEL</u>	the timeout threshold, in seconds, in which the processing of an intracluster cancellation request must occur.	15	2147483647	MXMSG parameter value

The optional `MXCANCEL` parameter specifies the maximum number of seconds for which one cluster nucleus (the sending nucleus) waits for the termination of another nucleus (the peer nucleus) in the cluster that it has canceled for not responding to an internal, intracluster command. If a value for `MXCANCEL` is not specified, the value specified for the ADARUN `MXMSG` parameter is used. Values from zero through 14 are not valid `MXCANCEL` values.

If the sending nucleus does not receive notification that the canceled nucleus has terminated within the `MXCANCEL` timeout period, it does one of the following things:

- It returns response 124, subcode 28 (no reply) to the user on whose behalf the intracluster command was issued.
- It terminates itself abnormally if the communication occurred on behalf of an internal process that must not fail.

When a cluster nucleus fails, the other nuclei in the cluster can recover from the failure only once they know for sure that the first nucleus has disconnected from the shared structures and is no longer writing to the database. When an unresponsive cluster nucleus has been canceled, it is important that it terminate quickly, so that it does not hold up the recovery process performed by the other nuclei in the cluster.



Note: Do not specify an `MPMDUMP` JCL statement for a cluster nucleus. An `MPMDUMP` is written before the failing nucleus disconnects from the shared structures and may significantly delay the start of the online recovery process performed by the other nuclei in the cluster. Instead, on z/OS systems, consider specifying a `//SVCDUMP DD DUMMY` statement, which usually produces a very quick dump prior to disconnecting. On other systems, and as a backup method on z/OS, specify a regular system dump statement (e.g., `SYSUDUMP` on z/OS) for a dump that is produced after disconnecting.

The value for `MXCANCEL` can be changed dynamically (for the time the Adabas nucleus is running) using the `MXCANCEL` operator command.

MXCANCELWARN : Timeout Threshold for Canceled Peer Nucleus Warning

Parameter	Specify . . .	Minimum	Maximum	Default
<u>MXCANCELWARN</u>	the timeout threshold, in seconds, for an intracluster cancellation request to occur before an early warning about the failure of the cancellation request is issued.	0 (no warning) or 4	MXCANCEL – 1	MXCANCEL / 4

The optional `MXCANCELWARN` parameter specifies the number of seconds after an intracluster cancellation request is made when the cluster nucleus requesting the cancellation should issue warnings about the inability of the canceled peer nucleus to terminate. If no value is specified for the `MXCANCELWARN` parameter, the value used is calculated as the value of the `ADARUN MXCANCEL` parameter divided by four. When dividing by four to determine the default value, the quotient is rounded up to the next integer value.

The value of `MXCANCELWARN` must be less than the value of `MXCANCEL`. If `MXCANCELWARN` is set to a value greater than or equal to the value specified for `MXCANCEL`, the value is rejected. If `MXCANCELWARN` is set to a value of zero (0), no warnings are issued about canceled cluster nuclei that are slow to terminate.

If the target nucleus does not terminate within the time period set by `MXCANCELWARN`, the nucleus that issued the cancellation request writes message `ADAX9G` to the operator console, identifying the target nucleus by its nucleus ID, job name, and system name. This message indicates that the nucleus writing the message may be in danger of terminating itself if the canceled peer nucleus does not terminate within the timeout period defined by the `MXCANCEL` parameter. You can use the `ADAX9G` message to trigger an alert or take other appropriate action before the `MXCANCEL` timeout period expires.

If the peer nucleus targeted for cancellation terminates after the `ADAX9G` message is written, the nucleus that issued the cancellation request writes message `ADAX9K` to the operator console, indicating that the cancellation occurred and the requesting nucleus is no longer in danger of terminating itself as a result of this particular incident.

The value for `MXCANCELWARN` can be changed dynamically (for the time the Adabas nucleus is running) using the `MXCANCELWARN` operator command.



Note: If the value of the `MXCANCEL` parameter is dynamically changed using the `MXCANCEL` operator command while the nucleus is running, but its new value is less than or equal to the current `MXCANCELWARN` setting, the value of `MXCANCELWARN` is automatically set to zero. You can then use the `MXCANCELWARN` operator command to dynamically set the `MXCANCELWARN` parameter to a new value that is less than the new `MXCANCEL` value.

MXMSG: Timeout Threshold for Internucleus Command Processing

Parameter	Specify . . .	Minimum	Maximum	Default
MXMSG	the timeout threshold, in seconds, in which the processing of an intraccluster command must occur.	15	32767	300

The optional `MXMSG` parameter specifies the maximum number of seconds in which one cluster nucleus waits for the response of another nucleus in the cluster to an internal, intraccluster command. If a value for `MXMSG` is not specified, a value of 300 seconds (5 minutes) is used.

If the receiving nucleus does not respond to the intraccluster command within the time period set by the `MXMSG` parameter, the requesting cluster nucleus issues a cancellation request for the unresponsive nucleus. The cancellation is requested to preserve the ability of all of the other active nuclei in the cluster to process user commands.

Adabas nuclei working on the same cluster database must collaborate to keep the database physically and logically consistent while processing user commands. To do this, they intermittently submit internal, intraccluster commands to one another. If a nucleus in the cluster does not respond to these intraccluster requests in a timely manner, the other nuclei in the cluster may quickly or gradually become unable to continue processing user commands.

Problems that may prevent a cluster nucleus from responding to intraccluster commands fall into two broad categories:

- Problems have occurred in the cluster nucleus, such as a CPU loop or a resource deadlock. In such cases, the nucleus experiencing the problem must be terminated to resolve the problem and the sooner this happens, the sooner the other cluster nuclei can recover from the failure and get back to normal processing.
- Problems have occurred outside the cluster nucleus, such as CPU starvation or a problem in the host system. In such cases, the cause of the delayed response might possibly be removed by system measures unrelated to the cluster database, and letting the other cluster nuclei wait some time for a resolution might prevent an unnecessary abnormal termination of a cluster member.

The value of `MXMSG` should strike a balance between these two categories, so that faulty cluster nuclei that cannot possibly stay active are canceled reasonably quickly while cluster nuclei that are victims of temporary system problems are not unnecessarily terminated. For assistance in setting the value of the `MXMSG` parameter, review the statistics provided on the messaging performance reports produced at termination or by the `DXMSG` operator command.

The value for `MXMSG` can be changed dynamically (for the time the Adabas nucleus is running) using the `MXMSG` operator command.

MXMSGWARN : Timeout Threshold for Internucleus Command Processing Warning

Parameter	Specify . . .	Minimum	Maximum	Default
MXMSGWARN	the timeout threshold, in seconds, for a response to an intracenter command to occur before an early warning about the failure of the intracenter request is issued	0 (no warning) or 4	MXMSG - 1	MXMSG / 4

The optional MXMSGWARN parameter specifies the number of seconds after an intracenter request is made when the cluster nucleus issuing the intracenter request should issue warnings about the failure of the target cluster nucleus to respond. If no value is specified for the MXMSGWARN parameter, the value used is calculated as the value of MXMSG divided by four. When dividing by four to determine the default value, the quotient is rounded up to the next integer value.

The value of MXMSGWARN must be less than the value of MXMSG. If MXMSGWARN is set to a value greater than or equal to the value specified for MXMSG, the value is rejected. If MXMSGWARN is set to zero, no warnings are issued about cluster nuclei that are slow to respond.

If the target nucleus does not respond within the time period set by MXMSGWARN, the nucleus that issued the request writes message ADAX9C to the operator console, identifying the target nucleus by its nucleus ID, job name, and system name. This message indicates that the target nucleus may be in danger of being canceled by the nucleus that issued the request if it does not respond within the timeout period defined by the MXMSG parameter. You can use the ADAX9C message to trigger an alert or take other appropriate action before the MXMSG timeout period expires.

If the target nucleus responds to the intracenter command after the ADAX9C message is written, the nucleus that issued the intracenter-command writes message ADAX9D to the operator console, indicating that a response was given and the target nucleus is no longer in danger of being canceled as a result of this particular incident.

The value for MXMSGWARN can be changed dynamically (for the time the Adabas nucleus is running) using the MXMSGWARN operator command.



Note: If the value of the MXMSG parameter is dynamically changed using the MXMSG operator command while the nucleus is running, but its new value is less than or equal to the current MXMSGWARN setting, the value of MXMSGWARN is automatically set to zero. You can then use the MXMSGWARN operator command to dynamically set the MXMSGWARN parameter to a new value that is less than the new MXMSG value.

MXSTATUS : Member-Level XCF Status Monitoring Heartbeat Interval

Parameter	Specify . . .	Minimum	Maximum	Default
<u>MXSTATUS</u>	the interval, in seconds, at which the member-level XCF status monitoring heartbeat should occur.	0 (no member-level status monitoring) or 15	21474836	15

The optional `MXSTATUS` parameter specifies the number of seconds between member-level XCF status monitoring heartbeats. If no value is specified for the `MXSTATUS` parameter, a value of "15" is used.



Note: Note: The `MXSTATUS` parameter is only used by Adabas Cluster Services and not by Adabas Parallel Services. Adabas Parallel Services does not use XCF and ignores this parameter when it is set.

If `MXSTATUS` is set to zero, member-level status monitoring will not occur. However, system-level status monitoring always occurs.

The value for `MXSTATUS` can be changed dynamically (for the time the Adabas nucleus is running) using the `MXSTATUS` operator command.

XCF status monitoring provides a second method by which Adabas Cluster Services can warn you that a cluster nucleus might be unable to respond in a timely way to intracluster commands. The first method is, of course, via the normal intracluster communication that occurs between cluster members. If a nucleus has heartbeat exceptions (as determined by XCF status monitoring), it most likely will be unable to process and respond to an intracluster command; if a nucleus is slow to respond to an intracluster command, it might or might not have a heartbeat monitor exception (a nucleus may appear to be active to XCF but be unable to respond to an intracluster command). If the ADARUN `MXMSGWARN` parameter for a cluster nucleus is nonzero, it produces warning messages (ADAX9B or ADAX9C) when intracluster communication with other nuclei in the cluster is too slow; likewise, when XCF status monitoring determines that a nucleus is missing its heartbeat updates, it produces warning messages (ADAX22 and ADAX04). You can use an automated mechanism set up at installation to raise an alert or take other appropriate action based on the existence of these messages, as they identify existing or potential problems in the cluster.

MXWTOR : Self-Termination Operator Query Interval

Parameter	Specify . . .	Minimum	Maximum	Default
<u>MXWTOR</u>	the timeout interval, in seconds, in which an operator response to a self-termination operator query must occur.	0 (no warning) or 15	64800	0

The optional `MXWTOR` parameter specifies the maximum number of seconds that a cluster nucleus should wait for an operator response to a self-terminating operator query. If no value is specified for the `MXWTOR` parameter, a value of zero is used. Values from 1 through 14 are not valid `MXWTOR` values.

A cluster nucleus issues a self-terminating operator query to the console via message `ADAX9J` if a canceled peer nucleus remains active after the time period defined by the `MXCANCEL` parameter. The `MXWTOR` parameter specifies how long the nucleus should wait for a response to this query.

One way that the operator may reply to the `ADAX9J` message is to make sure that the canceled cluster nucleus mentioned in the message actually terminates; once this happens, the `ADAX9J` message is retracted automatically, an online recovery process is performed, and the nucleus that issued the `ADAX9J` message remains active.

If the operator does not respond to the `ADAX9J` message within the time period set by the `MXWTOR` parameter and the canceled peer nucleus has not terminated in the meantime, the nucleus that requested the operator response and printed message `ADAX9J` terminates itself with message `ADAX99`.

If `MXWTOR` is set to zero (the default), the operator query does not occur (no `ADAX9J` messages are written) and no operator intervention is expected. Instead, the nucleus terminates itself immediately with the `ADAX99` message and user abend code 79.

Some installations may want to specify `MXWTOR` value long enough to allow the operator a good opportunity to bring down the canceled peer nucleus. Other installations may prefer not to get such operator queries at all.

The value for `MXWTOR` can be changed dynamically (for the time the Adabas nucleus is running) using the `MXWTOR` operator command.

NUCID : Adabas Cluster Nucleus ID



Note: The NUCID parameter replaces the earlier parameters PLXID and SMPID.

Parameter	Specify . . .	Possible Values	Default
NUCID	the ID for a nucleus in an Adabas cluster.	0-65000	0

NUCID identifies an Adabas cluster nucleus. The cluster may span operating system images under Adabas Cluster Services, or it may be confined to a single operating system image under Adabas Parallel Services. A value greater than 0 (the default) identifies this nucleus as an Adabas cluster nucleus.

Value	Meaning
0	The default setting. This is not an Adabas cluster nucleus.
1-65000	This is an Adabas cluster nucleus. This value must be unique among all target IDs declared to the Adabas router (ADASVC): Adabas databases (DBID), cluster nuclei (NUCID), Natural buffer pools and other MPM servers, Entire Net-Work, and internal targets used by Adabas Cluster Services and Adabas Parallel Services (restricted range 65,001-65,479). Nucleus IDs for the same cluster need not be specified contiguously.

Guidelines

- Each nucleus being run must have its own Work data set.
- All nuclei in a cluster must be run against the same database.
- Software AG strongly recommends that you run utilities or user applications against a nucleus in the local z/OS image. Although a remote open nucleus may be used, performance degradation due to network overhead is likely.

Example

The following example starts an Adabas cluster session with the specified cluster nuclei:

```
ADARUN PROG=ADANUC,NUCID=1, . . .
```

PROGRAM : Program to Run

Parameter	Specify:	Possible Values	Default
<code>PROGRAM</code>	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 <i>Adabas Session Execution</i> , in <i>Adabas Operations</i> documentation.
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 your Adabas TP monitor installation documentation.
USER	a user program to be run using a non-reentrant Adabas batch/TSO link routine. For more information, read <i>Linking Applications to Adabas</i> , in <i>Adabas Operations</i> documentation.
<i>utility-name</i>	an Adabas utility Specify an Adabas utility for <i>utility-name</i> . For more information, refer to <i>Adabas Utilities</i> documentation.

Examples

The following example starts an Adabas nucleus.

```
ADARUN PROGRAM=ADANUC
```

The following example starts an Adabas Review hub.

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


7

Entire Net-Work Parameter Statements

■ Conventions	44
■ XCF NODE Statement	47
■ XCF DRIVER Statement	57
■ XCF LINK Statement	61

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: `NODE`, `DRIVER`, `LINK`.

The `NODE` statement specifies the global operating parameters for the Entire Net-Work node (note that all interregion communication information is specified by the `ADARUN` control statement described in the previous section).

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.

A `DRIVER` statement must be specified for each line driver. The `DRIVER` statement indicates the operational parameters for the line driver's access method and default values for the related links.

A `LINK` statement may be specified for each link to a remote node. The `LINK` statement specifies the driver-specific parameters to be used for the connection. Each `LINK` statement must be associated with a previously specified `DRIVER` statement.

This chapter covers the following topics:

Conventions

This section covers the following topics:

- [Syntax Conventions](#)
- [Syntax Rules](#)

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.	<div><code>DRIVER TCPI [DRVCHAR = driver-char #]</code></div> <p>The syntax elements <code>DRIVER</code>, <code>TCPI</code>, and <code>DRVCHAR</code> are Entire Net-Work keywords.</p>

Convention	Description	Example
lowercase, italic, normal font	Syntax elements appearing in lowercase and normal, italic font identify items that you must supply.	<div>DRIVER TCPI [DRVCHAR = <i>driver-char</i> #]</div> <p>The syntax element <i>driver-char</i> 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.</p>
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.	<div>DRIVER TCPI [DRVCHAR = <i>driver-char</i> #]</div> <p>In the example above, # is the default that will be used for the DRVCHAR parameter if no other record buffer length is specified.</p> <p>Also in the example above, the short version of the DRVCHAR parameter is D.</p>
vertical bars ()	Vertical bars are used to separate mutually exclusive choices. Note: In more complex syntax involving the use of large brackets or braces, mutually exclusive choices are stacked instead.	<div>DRIVER TCPI API = { IBM HPS BS2 CMS CNS FUJ ILK OES }</div> <p>In the example above, you must select IBM, HPS, BS2, CMS, CNS, FUJ, ILK, or OES for the API parameter. There are no defaults.</p>
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.	<div>DRIVER TCPI [DRVCHAR = <i>driver-char</i> #]</div> <p>In this example, the DRVCHAR parameter is optional.</p>
braces ({ })	Braces are used to identify required elements. When multiple elements are stacked or separated by vertical bars within brackets, one and only one of the elements must be supplied.	<div>DRIVER TCPI API = { IBM HPS BS2 CMS CNS FUJ ILK OES }</div> <p>In this example, one of the following values is required for the API parameter: IBM, HPS, BS2, CMS, CNS, FUJ, ILK, or OES.</p>

Convention	Description	Example
other punctuation and symbols	All other punctuation and symbols must be entered exactly as shown.	<pre>LINK linkname TCPI [INETADDR = n1.n2.n3.n4] [,][-]</pre> <p>In this example, the periods must be specified in the IP address.</p> <p>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.</p>

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, or DRIVER) must be specified as the first nonblank 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, . . .
```

XCF NODE Statement



Note: This section describes the `NODE` statement and parameters for the XCF line driver. For detailed information see the sections [XCF DRIVER Statement](#) and [XCF LINK Statement](#).

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

Format

The following is the format of the `NODE` statement:

```
NODE nodename BUFFERS = (abufsize,ltbufsize,stbufsize,pfxbufsize),-  
CQTIMER = seconds,-  
DEFINE = {N | Y },-  
DOMAIN = domain-name,-  
DUMP = storage-area,-  
LOG = {N | Y },-  
MAXPATH = link-count,-  
MSGFORM = message-form,-  
NID0 = {N | Y },-  
NTRACE = n,-  
REPLYTIM = seconds,-  
START={Y | N },-  
TIMER = seconds,-  
TRACE = argument-list,-  
UCMSG={ N | Y },-  
ULINK={ N | Y }
```

Parameters

The **NODE** statement parameters along with a description of their function and their accepted values and defaults are described in the following sections.



Note: The underlined portion of the parameter is the minimum abbreviation.

- **nodename**
- **BUFFERS**
- **CQTIMER**
- **DEFINE**
- **DOMAIN**
- **DUMP**
- **LOG**
- **MAXPATH**
- **MSGFORM**
- **NID0**
- **NTRACE**
- **REPLYTIM**
- **START**
- **TIMER**
- **TRACE** | **TROFF** | **TRON**
- **UCMSG**

- **ULINK**

nodename

A 1-8 character name for this node. The node name 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	Specify...	Minimum	Maximum	Default
BUFFERS	the amount of working storage (in bytes) reserved for internal buffer management.	0	2147483647 (see text)	see text

The values required depend on the number and types of line drivers and links, and on the message volume; these values can be adjusted according to usage statistics printed at the end of each Entire Net-Work session, or from issuing the operator command "D S" (display statistics).

Specify decimal values for the operands ranging from 0 to 2147483647 bytes; or, optionally, a value followed by either the multiplier "M" (multiply by 1048576) or "K" (multiply by 1024). If followed by "M", the specified value must range from 0 to 2048; if followed by "K", the value must range from 0 to 2097151. If a subparameter is omitted or zero is specified, no corresponding buffer pool will exist for this node. The requested storage space must be available; if the space is not available, Entire Net-Work ends with error message NET0013.

For OS/390 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 (**pxbufsize**) is used to set the size of the page-fixed buffer pool.

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

where

abufsize	is the asynchronous buffer pool size. This required subparameter specifies the storage space to reserve for the asynchronous buffer pool; used whenever asynchronously scheduled routines (exit routines, interrupt handlers) require storage. All requests to this pool are serialized.
ltbufsize	is the long-term buffer pool size. This required subparameter specifies the storage space to reserve for the long-term buffer pool, from which comparatively small buffers are allocated for relatively long-term use (for example, dynamically created control blocks).
stbufsize	is the short-term buffer pool size. This required subparameter specifies the storage space to reserve for the short-term buffer pool, from which comparatively large individual buffers are allocated for short time periods (for example, message and transmission buffers).
pfxbufsize	is the page-fixed buffer pool size. This optional subparameter specifies the bytes to reserve for the page-fixed buffer pool, from which all requests for page-fixed buffers are filled. The use of this pool is similar to that of the short-term buffer pool; that is, relatively large buffers are allocated for short time periods. The default '0' (zero) means that no page-fixed buffer pool is allocated. Because the segment size of the page-fixed buffer pool is identical to the page size of the operating system, more page-fixed storage than short-term storage may be required.

CQTIMER

Parameter	Specify...	Minimum	Maximum	Default
<u>CQTIMER</u>	the time allowed (in seconds) for a user to retrieve command results with a router 16-call.	1	2147483647	60

Specify a practical decimal value, depending on the node system's environment. The maximum value is approximately 68 years: effectively, no timeout will occur. The default is approximately one minute.

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



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

DEFINE

Parameter	Specify...	Possible Values	Default
<u>DEFINE</u>	whether the <code>DEFINE</code> operator command can be used to define links during Entire Net-Work operation.	N Y	N

If "Y" is specified, the `DEFINE` operator command is accepted and executed; "N" (the default) rejects the `DEFINE` operator command.

DOMAIN

Parameter	Specify...	Possible Values	Default
<u>DOMAIN</u>	a domain name	1-6 character name	blank (no domain name)

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

DUMP

Parameter	Specify...	Possible Values	Default
<u>DUMP</u>	the areas of storage to be printed after an abnormal termination of Entire Net-Work.	see text	ALL

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.

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

where

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.


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:

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

If values conflict, the last value specified is used. 'NONE' is used in the following example:

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

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

LOG

Parameter	Specify...	Possible Values	Default
LOG	whether control flow is recorded and/or selected data areas are logged to the printer data set.	see text	NO


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

LOG 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. It can have the following values:

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

where

OFF, NO	No logging occurs during this Entire Net-Work node's session.
ON, YES, and FULL	Log both the node's checkpoint records and data areas.
SHORT	Log the checkpoint records only.

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

MAXPATH

Parameter	Specify...	Minimum	Maximum	Default
<u>MAX</u> PATH	the maximum path length (in links) that a message can travel in the network.	1	32767	4

The default (4) results in a stack large enough for four enroute node IDs.

The Entire Net-Work 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 on the way to its target, each node's ID is added to the stack.

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



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

MSGFORM

Parameter	Specify...	Possible Values	Default
<u>MSG</u> FORM	the format of console messages and DDPRINT output.	1 2 3 4	2

where the valid values are

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 for information, W for warning, or E for error) 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: <code>NET0090I NODE2 : BUFFER USAGE STATISTICS</code>
4	provides the message number followed by the node name, not padded, of the issuing Entire Net-Work node. For example: <code>NET0090I NODE2: BUFFER USAGE STATISTICS</code>

**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	Specify...	Possible Values	Default
<u>NID0</u>	whether to force a node ID of 0 (zero) for all unsolicited connections.	N Y	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.

NTRACE

Parameter	Specify...	Minimum	Maximum	Default
<u>NTRACE</u>	the number of 32-byte entries to be allocated for the internal trace table.	0	2147483647	0

Specify a practical decimal value, depending on the node's system configuration. The maximum may be limited by the system. If the default (0) is used, no trace table is allocated for this node and no tracing takes place, whether `TRACE=` or `TRON=` is specified or not.

REPLYTIM

Parameter	Specify...	Minimum	Maximum	Default
<u>REPLYTIM</u>	the time allowed (in seconds) for a user request to complete.	1	2147483647	60

A request is considered complete when the originating node receives a reply.

Specify a practical decimal value, depending on the node system's operation. The maximum value is approximately 68 years: effectively, no timeout will occur. The default is approximately one minute.

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



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

START

Parameter	Specify...	Possible Values	Default
<u>START</u>	whether Entire Net-Work starts normal operations automatically.	N Y	Y

The values for `START` are interpreted as follows:

Y	Entire Net-Work automatically starts all line drivers and initiates connections for all links that have the parameter <code>ACQUIRE=Y</code> specified.
N	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 <code>START</code> operator command.

TIMER

Parameter	Specify...	Minimum	Maximum	Default
<u>TIMER</u>	the interval (in seconds) between handling of time-dependent requests.	1	2147483647	30

Each time the specified interval expires, 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. The maximum value is approximately 68 years: 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.

TRACE | TROFF | TRON

Parameter	Specify...	Possible Values	Default
<code>TRACE</code> <code>TROFF</code> <code>TRON</code>	trace control parameters for performing program traces.	trace (see text)	none

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.

`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 that the `NTRACE` parameter must specify a nonzero value in order to permit tracing.

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

One or more trace values can be specified. If more than one value is specified, the values must be enclosed in parentheses and separated by commas:

`(trace,trace,...)`

The value 'trace' can be one or more of the following:

MAIN	mainline trace
BPM	buffer pool manager trace
TQM	transmission queue manager trace
RQM	receive queue manager trace
XCF	XCF line driver trace

An XCF line driver trace can be requested for installed line drivers on the local node only.



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

UCMSG

Parameter	Specify...	Possible Values	Default
<u>UCMSG</u>	whether messages are issued in uppercase (Y) or mixed case (N).	N Y	N



Notes:

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

ULINK

Parameter	Specify...	Possible Values	Default
<u>ULINK</u>	whether multiple links are allowed between two Entire Net-Work nodes.	N Y	N

The values for `ULINK` are interpreted as follows:

N	(the default) multiple links between two Entire Net-Work nodes are allowed.
Y	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 are rejected.

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.



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

XCF DRIVER Statement

The Entire Net-Work DRIVER control statement defines the line driver type (that is, XCF) to be loaded. Only one DRIVER statement may be specified for a given line driver type.

DRIVER statement parameters are read from a sequential file during system startup, and can be modified after startup using the `ALTER` operator command. Some parameters can be modified when the line driver is open or closed; others can be modified only when the line driver is closed.

See the operator commands `ALTER` and `CLOSE` in [XCF Line Driver Commands](#). The open/closed requirement for each parameter is included in its description.

The XCF DRIVER statement and its parameters are used to activate and define the characteristics of the local sysplex node. The access method name 'XCF' or 'XCFD' instructs Entire Net-Work to load the line driver module NETXCF, which must exist in the libraries defined for the related job step.



Note: The software accepts both XCF and XCFD as the access method name. However, the name used in the DRIVER statement must also be used for the LINK statements.

This section covers the following topics:

- [Format](#)
- [Parameters](#)

Format

The XCF DRIVER statement has the following format:

```
DRIVER XCFD ACCEPTUI = { Y | N },-  
            GROUP = group-name,-  
            LARGMSG = { message-size | 8192 },-  
            PSTATS = { Y | N },-  
            RSTATS = { Y | N },-  
            SMALLMSG = { message-size | 1024 },-  
            STATINT = { statistics-interval | 3600 },-  
            TRACESIZ = { internal-trace-buffer-size | 8192 }
```

where 'XCFD' is the four-character access method name for the XCF driver type:

Driver	Module Name	Access Method
XCFD	NETXCF	parallel sysplex cross-system coupling facility

Parameters



Note: The underlined portion of the parameter is the minimum abbreviation.

- ACCEPTUI
- GROUP
- LARGMSG
- PSTATS
- RSTATS
- SMALLMSG
- STATINT
- TRACESIZ

ACCEPTUI

Parameter	Specify...	Possible Values	Default
<u>ACCEPTUI</u>	whether this node will connect to other nodes in the XCF group that are not defined by a LINK statement.	N Y	Y

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

GROUP

Parameter	Specify...	Possible Values	Default
<u>GROUP</u>	the Entire Net-Work group name.	group-name	WCPXCFV5

All Entire Net-Work nodes in the SYSPLEX must use the same unique XCF group name. The group name must be the same on all Entire Net-Work nodes; it must also be different from the name of any other product or system using XCF. Results are unpredictable if Entire Net-Work nodes are assigned to the same group as other types of systems such as DB2 or CICS. The GROUP parameter specifies the Entire Net-Work group name, which may be defined in the COUPLExx member in SYS1.PARMLIB (see step *Modify COUPLExx and the CFRM Policy* in the *Installation* documentation). The GROUP parameter can be modified only when the driver is closed.

LARGMSG

Parameter	Specify...	Default
<u>LARGMSG</u>	the minimum size (in bytes) of a large message for statistics reporting.	8192 (8K)

Any data messages larger than this size are counted as large messages. Any data messages smaller than this size are counted as medium or small messages, depending on the value specified by the SMALLMSG parameter described below.

For example, using the default values for SMALLMSG and LARGMSG:

- Data messages between 1 and 1024 bytes long are reported as small messages.
- Data messages between 1025 and 8192 bytes long are reported as medium messages.
- Data messages larger than 8192 bytes long are reported as large messages.

PSTATS

Parameter	Specify...	Possible Values	Default
<u>P</u> STATS	whether statistics are printed automatically at regular intervals.	N Y	N

This parameter is only used to set the default for all links.

RSTATS

Parameter	Specify...	Possible Values	Default
<u>R</u> STATS	whether statistics are reset automatically at regular intervals.	N Y	Y

This parameter is only used to set the default for all links.

SMALLMSG

Parameter	Specify...	Default
<u>S</u> MALLMSG	the maximum size (in bytes) of a small message for statistics reporting.	1024

Any data messages smaller than or equal this size are counted as small messages. Any data messages larger than this size are counted as medium or large messages, depending on the value specified by the `LARGMSG` parameter described above.

STATINT

Parameter	Specify...	Minimum	Maximum	Default
<u>S</u> TATINT	the amount of time, in seconds, before statistics are automatically reported or reset.	0	2147483647	3600

This parameter is only used to set the default for all links.

TRACESIZ

Parameter	Specify...	Default
TRACESIZ	the size (in bytes) of the internal trace buffer.	8192 (8K)

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

XCF LINK Statement

The LINK statement is currently not required for the XCF DRIVER statement. During the initialization process, the XCF line driver requests a list of all other members in its XCF group and then, if the DRIVER statement parameter ACCEPTUI=Y is set, logically connects to each member.

Although the LINK statement is optional, it can be added for documentation or security purposes.



Note: If two or more Entire Net-Work nodes are active in the sysplex within the same group, the Entire Net-Work operator command `D L` (display links) can be used to display the list of dynamically added links and parameter-defined links.

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



Note: The software accepts both XCF and XCFL as driver names. However, the name used in the DRIVER statement must also be used for the LINK statements.

The LINK statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. Some parameters can be modified when the link is open or closed; others can be modified only when the link is closed. See the operator commands ALTER and CLOSE in *XCF Line Driver Commands*. The open/closed requirement for each parameter is included in its description.

- [Format](#)

- Parameters

Format

The XCF LINK statement has the following format:

```
LINK linkname XCFD PSTATS = { Y | N },-  
                   RSTATS = { Y | N },-  
                   SAF = { Y | L | N },-  
                   STATINT = statistics-interval,  
                   WEIGHT = { n | 256 }
```

where 'XCFD' is the four-character name of the XCF line driver. The XCF DRIVER statement must precede the LINK statement.

Parameters



Note: The underlined portion of the parameter is the minimum abbreviation.

- *linkname*
- PSTATS
- RSTATS
- SAF
- STATINT
- WEIGHT

linkname

This required parameter specifies the name by which this link is to be known; it is a one- to eight-character name identifying the link and must be unique on the node. It is positional and must be specified immediately after the LINK keyword and immediately before the driver name (XCFD).



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.

PSTATS

Parameter	Specify...	Possible Values	Default
<u>P</u> STATS	whether statistics are printed automatically at regular intervals.t	N Y	see text

The default value is the value of the `PSTATS` parameter on the `DRIVER` statement (see *XCF DRIVER Statement* parameter [PSTATS](#)).

RSTATS

Parameter	Specify...	Possible Values	Default
<u>R</u> STATS	whether statistics are reset automatically at regular intervals.	N Y	see text

The default value is the value of the `PSTATS` parameter on the `DRIVER` statement (see *XCF DRIVER Statement* parameter [RSTATS](#)).

SAF

Parameter	Specify...	Possible Values	Default
<u>S</u> AF	whether Entire Net-Work will call the SAF Interface for all incoming requests on this link.	N L Y	N

Failure to load the Interface is considered a security violation and Entire Net-Work will shut down. If `SAF=L`, the calls are traced and the output directed to DDPRINT. An error code is transmitted to the user if access to `SAF` is denied. The `SAF` parameter can be modified when the link is open or closed.

STATINT

Parameter	Specify...	Minimum	Maximum	Default
<u>S</u> TATINT	the elapsed time (in seconds) before statistics are automatically reported or reset.	0	2147483647	see text

The default value is the value of the `STATINT` parameter on the `DRIVER` statement (*XCF DRIVER Statement* parameter [STATINT](#)).

WEIGHT

Parameter	Specify...	Minimum	Maximum	Default
<u>WEIGHT</u>	the weight of this link with respect to other links going to the same node.	1	999999	256

If a given target can be reached by more than one path (chain of connected links), the path with the lowest weight is used. Slow or expensive links should be given a higher value than fast or inexpensive links. The `WEIGHT` parameter can be modified only when the link is closed.

8

Adabas Cluster Operator Commands

■ ADACOM Operator Commands	66
■ Adabas Cluster Nucleus Operator Commands	71

This part of the documentation provides information on the operator commands for ADACOM and for an Adabas cluster nucleus.

This chapter covers the following topics:

ADACOM Operator Commands

Special ADACOM operator commands exist to display and control the multiprocessing environment. These commands, which are similar to regular Adabas operator commands, are issued to the local ADACOM initialization job.

This section covers the following topics:

- [Commands Issued during ADACOM Initialization](#)
- [Commands Issued after ADACOM Initialization](#)
- [DIM - Display Images](#)
- [DN - Display Active Nuclei](#)
- [DUMP - Dump Storage Areas](#)
- [SN - Set Nucleus Status](#)

Commands Issued during ADACOM Initialization

When running the Adabas Cluster Services initialization routine ADACOM, the operator commands described in this section can be specified in one of the following input formats:

```
MODIFY  jobname,command  
F      jobname,command
```

where

jobname	name of the ADACOM job or started task
command	one of the operator commands described in this section

Commands Issued after ADACOM Initialization

After initialization, any command issued is directed to the last SVC/DBID pair encountered in the input.

```
MODIFY  jobname,SVC=svc,DBID=dbid,command
F      jobname,SVC=svc,DBID=dbid,command
```

A command can be given only if the SVC/DBID pair is already active (has been specified before, and not terminated). Otherwise, specify the SVC/DBID pair without a command to activate the SVC/DBID (as shown above), and then issue the command separately (as shown below).

```
MODIFY  jobname,SVC=svc,DBID=dbid[,NU= maxusers] [,FORCE=YES],command
F      jobname,SVC=svc,DBID=dbid[,NU= maxusers] [,FORCE=YES],command
```

When you change or add an SVC/DBID set in this manner, the new set becomes the default for all commands issued until the set is changed.

```
MODIFY  jobname,SVC=svc,DBID=dbid,command,ADAEND
F      jobname,SVC=svc,DBID=dbid,command,ADAEND
```



Caution: You must specify the SVC and DBID when terminating an SVC/DBID combination. Otherwise, the ADAEND command terminates the entire ADACOM job. The current set default does not apply when using ADAEND.

Every operator command is directed to the ADACOM job and is echoed with the message

```
PLI060 SVC=svc DBID=dbid OPERATOR COMMAND:xxx
```



Note: ADACOM writes global ADACOM messages into the output data set with the DD name COMPRINT. It writes messages pertinent to an individual SVC/DBID combination into the output data set with the DD name 'Pssdddd', where *ss* is the last two digits of the SVC number and *dddd* is the database ID.

DIM - Display Images

DIM [*image-name*]

Use DIM to display the number of commands processed and the number of currently active users for each active nucleus on every image known to the local image.

If the image name is specified, DIM displays the information for the specified image only.

Sample Output

```
PLI060 SVC=svc DBID=dbid OPERATOR COMMAND: DIM
PLI004 image-name NUCID UP LO RO -#USERS- -#CMNDS- LURA= RULA=
      jobname      00001 Y  Y  N  00000152 00000001
PLI007 image-name NUCID UP LO RO -#USERS- -#CMNDS- LURA= RULA=
      jobname      00002 Y  ** NN 00000089 00000000
```

Nuclei on the local image start open; nuclei on remote images start closed to the local image. The display uses the following indicators:

image-name	the name of the image: in PLI004 displays, this is the local image; in PLI007 displays, this is a remote image
jobname	the job name of a cluster nucleus active on that image
NUCID	the unique cluster nucleus identifier between 1 and 65000
UP	whether (Y or N) the specified nucleus is available for normal processing
LO	whether the specified nucleus is on the local image and open for new users (Y); on the local image and closed for new users (N); or not on the local image (**)
RO	whether the specified local nucleus (PLI0004 message) is closed for new users from remote images (N) or open for new users from all images (G); or whether the specified remote nucleus (PLI007 message) is closed for new users from remote (as seen from that nucleus) images (NN), open for new users from this image (where DIM was issued) but not necessarily other images (LN), or open for new users from all images (NG and LG)
#USERS	the number of users currently assigned to the specified nucleus
#CMNDS	the number of commands currently in progress in the specified nucleus
LURA=	the number of local users currently assigned to a remote image
RULA=	the number of remote users currently assigned to the local image

DN - Display Active Nuclei

Use **DN** to display the number of commands processed and the number of currently active users for each active nucleus on the local image only.

Sample Output

```
PLI060 SVC=svc DBID=dbid OPERATOR COMMAND:DN
PLI004 image-name NUCID UP LO RO -#USERS- -#CMNDS- LURA= RULA=
      jobname      00001 Y  Y  N  00000152 00000001
PLI004 image-name NUCID UP LO RO -#USERS- -#CMNDS- LURA= RULA=
      jobname      00002 Y  Y  N  00000000 00000000
```

Nuclei on the local image start open. The display uses the following indicators:

image-name	the name of the image: in PLI004 displays, this is the local image name
jobname	the job name of the cluster nucleus active on the local image
NUCID	the unique cluster nucleus identifier between 1 and 65000
UP	whether (Y or N) the specified nucleus is available for normal processing
LO	whether the specified local nucleus is open (Y) or closed for new users (N)
RO	whether the specified nucleus is closed for new users from remote images (N) or open for new users from all images (G)
#USERS	the number of users currently assigned to the specified nucleus
#CMNDS	the number of commands currently in progress in the specified nucleus
LURA=	the number of local users currently assigned to a remote image
RULA=	the number of remote users currently assigned to the local image

DUMP - Dump Storage Areas

The **DUMP** command displays internal information about various storage areas that Adabas Cluster Services and Adabas Parallel Services use for controlling the communication with the cluster nuclei. Use this command as directed by Software AG Support. Its output might be helpful in the diagnosis of communication-related problems with cluster databases.

DUMP *stg-area*

where *stg-area* can be any of the following storage areas: CLUDSP, IDTE, IDTH, IDTHPRFX, PLXCB, PLXMAP, PLXNUC, PLXUSER, or SVC.

SN - Set Nucleus Status

```
SN {RMTALL | image-name | nucleus-id} {OP | CL}
SN {LCLALL | nucleus-id} {OP | CL} {LCL | GBL}
```

where

RMTALL	is REMOTE ALL: all nuclei on all remote images
OP	is OPEN
CL	is CLOSE
LCLALL	is LOCAL ALL: all nuclei on the local image
LCL	is LOCAL: local users
GBL	is GLOBAL: all sysplex cluster users

The possible options settings for the SN command are described as follows:

Option	Action
RMTALL {OP CL}	open or close all nuclei on all remote images to the image issuing the command.
<i>image-name</i> {OP CL}	open or close all nuclei on the specified remote image to the image issuing the command.
<i>nucleus-id</i> {OP CL}	open or close the specified remote nucleus to the image issuing the command.
LCLALL {OP CL} LCL	open or close all nuclei on the local image to local users. Information is not broadcast to other images.
<i>nucleus-id</i> {OP CL} LCL	open or close the specified local nucleus to local users. Information is not broadcast to other images.
LCLALL {OP CL} GBL	open or close all nuclei on the local image to all sysplex cluster users.
<i>nucleus-id</i> {OP CL} GBL	open or close the specified local nucleus to all sysplex cluster users.

By default, nuclei start open to local users and closed to remote users.

After the nuclei start, the SN operator commands may be used to exercise some control over the assignment of users to nuclei. For example, in a node with no active local nuclei, it is necessary to open at least one remote nucleus to users from the local node in order to run user programs in that node.

Adabas Cluster Nucleus Operator Commands

The following console operator commands can be issued against a local Adabas cluster nucleus. In addition, the console operator commands documented in *Adabas Operations* documentation can be issued against any local nucleus.

Command	To ...
DMENTB	Display information about active Adabas cluster nuclei
DNFV	Display information about current file use
DPARM	Display Adabas session parameters currently in effect
DPPT	Display information about the nucleus's own PPT block
DXCACHE	Display the primary cache-related statistics
DXFILE	Display the cache-related statistics for 1 to 5 files
DXLOCK	Display the lock-related statistics
DXMSG	Display the Adabas Cluster Services messaging performance statistics
DXSTAT	Display all cache- and lock-related statistics
MXCANCEL	Dynamically change the ADARUN MXCANCEL setting
MXCANCELWARN	Dynamically change the ADARUN MXCANCELWARN setting
MXMSG	Dynamically change the ADARUN MXMSG setting
MXMSGWARN	Dynamically change the ADARUN MXMSGWARN setting
MXSTATUS	Dynamically change the ADARUN MXSTATUS setting
MXWTOR	Dynamically change the ADARUN MXWTOR setting



Note: See the *Adabas Cluster Services Operations* documentation for information about issuing Adabas utility ADADBS OPERCOM commands against the local cluster nucleus, a specified cluster nucleus, or all cluster nuclei.

DMENTB - Display Member State Table

The DMENTB command can be used to display the member state table, which provides information about all of the active nuclei in an Adabas cluster. The information produced by this command may be used by Software AG technical support.

The output produced by the DMENTB command includes a third flag that indicates whether a system- or message-level XCF status monitoring exception has been encountered and whether a message was issued for the exception.

Sample Output

```

17:28:02 ADAI29 Oper cmd: DMENTB
17:28:02 ADAX61 00226 2007-06-01 17:28:01 Member Status Table
17:28:02 ADAX61 00226 2007-06-01 17:28:01 Other members:      2
17:28:02 ADAX61 00226 2007-06-01 17:28:01 This system:          1
17:28:02 ADAX61 00226 2007-06-01 17:28:01
17:28:02 ADAX61 00226 2007-06-01 17:28:01          NUCID:      2,261
17:28:02 ADAX61 00226 2007-06-01 17:28:01          Internal ID:    1
17:28:02 ADAX61 00226 2007-06-01 17:28:01          System: DA3F
17:28:02 ADAX61 00226 2007-06-01 17:28:01          Flags 1/2/3: 11/00/00
17:28:02 ADAX61 00226 2007-06-01 17:28:01
17:28:02 ADAX61 00226 2007-06-01 17:28:01          NUCID:      2,262
17:28:02 ADAX61 00226 2007-06-01 17:28:01          Internal ID:    2
17:28:02 ADAX61 00226 2007-06-01 17:28:01          System: DAEF
17:28:02 ADAX61 00226 2007-06-01 17:28:01          Flags 1/2/3: 97/00/00
17:28:02 ADAX61 00226 2007-06-01 17:28:01
17:28:02 ADAX61 00226 2007-06-01 17:28:01          NUCID:      2,263
17:28:02 ADAX61 00226 2007-06-01 17:28:01          Internal ID:    3
17:28:02 ADAX61 00226 2007-06-01 17:28:01          System: DA3F
17:28:02 ADAX61 00226 2007-06-01 17:28:01          Flags 1/2/3: 97/00/00
17:28:02 ADAN41 00226 2007-06-01 17:28:01 Function completed

```

Flags 1, 2, and 3 in this report are hexadecimal values that provide the following information about each nucleus:

Flag	Provides
1	<p>The cluster member service summary, as a combination of one or more of the following values:</p> <ul style="list-style-type: none"> ■ X'80' – All services are active ■ X'40' – Work data set is needed for recovery ■ X'10' – Accepting and processing intracluster commands ■ X'04' – Cache services are available ■ X'02' – Lock services are available ■ X'01' – Messaging services are available <p>For example, a value of 97 for Flag 1 indicates that all services are active (X'80'), the nucleus is accepting and processing intracluster commands (X'10'), and cache (X'04'), lock (X'02') and messaging (X'01') services are all available (80+10+4+2+1=97).</p>
2	<p>The cluster member failure status, as a combination of one or more of the following values:</p> <ul style="list-style-type: none"> ■ X'80' – All services are down ■ X'40' – Canceled by this nucleus ■ X'10' – Abnormal termination of intracluster command processing occurred ■ X'04' – Abnormal exit from cache occurred

Flag	Provides
	<ul style="list-style-type: none"> ■ X'02' – Abnormal exit from lock occurred ■ X'01' – Abnormal exit from the XCF group (messaging) occurred
3	<p>The cluster XCF status monitoring summary, as a combination of one or more of the following values:</p> <ul style="list-style-type: none"> ■ X'80' – A system-level XCF status monitoring exception occurred. ■ X'40' – A member-level XCF status monitoring exception occurred. ■ X'02' – A warning message was issued for the system-level XCF status monitoring exception. ■ X'01' – A warning message was issued for the member-level XCF status monitoring exception. <p>For example, a value of 41 for Flag 3 indicates that a member-level XCF status monitoring exception occurred (X'40') and that a warning message (X'01') was issued for it (40+1=41).</p>

DNFV - Display Nucleus File Variables

Use `DNFV` to display information about current nucleus file use.

This command provides information about the files in use at a particular point in time. It also indicates which other nucleus has exclusive file control if, for example, a user program receives a response 148, subcode 15.

Sample Output

```
ADAI29 OPER CMD: DNFV
FNR=00008 A=Y U=Y ID=      CA=00000 CU=00001
```

where

FNR=nnnnnn	is the file number
A={Y N}	(yes or no) indicates whether the file is used for access (read and/or search)
U={Y N}	(yes or no) indicates whether the file is used for update. Use for update includes use for access.
ID=nucid	is the ID of the nucleus that owns the file lock, if the file is locked.
CA=nnnnnn	is the number of users on this nucleus who are currently accessing this file.
CU=nnnnnn	is the number of users on this nucleus who are currently updating this file.

DPARM - Display Adabas Session Parameters

The DPARM command can be used to display the Adabas session parameters currently in effect. The output produced by the DPARM command includes settings for the ADARUN parameters related to Adabas Cluster Services alert and timeout enhancements.

Sample Output

```
17:28:14 ADAI29 Oper cmd: DPARM
17:28:14 ADAN16 00226 2007-06-01 17:28:13 READONLY=NO,UTIONLY=NO
17:28:14 ADAN16 00226 2007-06-01 17:28:13 ASYTVS=YES,AOSLOG=NO
17:28:14 ADAN16 00226 2007-06-01 17:28:13 NC=200,NH=500,NT=20,NU=200
17:28:14 ADAN16 00226 2007-06-01 17:28:13 LBP=375808,LFP=12000,LWP=500000
17:28:14 ADAN16 00226 2007-06-01 17:28:13 LI=10000,LP=200,LQ=10000,LS=49920
17:28:14 ADAN16 00226 2007-06-01 17:28:13 LFIOP=100000,FMXIO=1,LU=65535
17:28:14 ADAN16 00226 2007-06-01 17:28:13 TT=900,TNAA=900,TNAE=900,CT=60
17:28:14 ADAN16 00226 2007-06-01 17:28:13 TNAX=900,MXTT=3600,MXTNA=3600
17:28:14 ADAN16 00226 2007-06-01 17:28:13 TLSCMD=300,MXTSX=3600
17:28:14 ADAN16 00226 2007-06-01 17:28:13 NOLOG
17:28:14 ADAN16 00226 2007-06-01 17:28:13 NUCID=2261,MXMSG=300,MXMSGWARN=60
17:28:14 ADAN16 00226 2007-06-01 17:28:13 MXCANCEL=600,MXCANCELWARN=120
17:28:14 ADAN16 00226 2007-06-01 17:28:13 MXSTATUS=15,MXWTOR=0
17:28:14 ADAN16 00226 2007-06-01 17:28:13 CLUSTER=SYSPLEX,CLUGROUPNAME=PTGCJP
17:28:14 ADAN16 00226 2007-06-01 17:28:13 CLULOCKNAME=ADA_LOCK7
17:28:14 ADAN16 00226 2007-06-01 17:28:13 CLUCACHENAME=ADA_CACHE7
17:28:14 ADAN16 00226 2007-06-01 17:28:13 DIRRATIO=4,ELEMRATIO=1
17:28:14 ADAN41 00226 2007-06-01 17:28:13 Function completed
```

DPPT - Display Parallel Participant Table (PPT)

Use DPPT to display all used PPT blocks.

This command produces internal information for use by Software AG technical support.

Sample Output

```
ADAI29 OPER CMD: DPPT
ADAN24 00006 2001-01-18 00:15:49 Display PPT RABNs 000005FB to 0000061A
ADAN24 00006 2001-01-18 00:15:49
ADAN24 00006 2001-01-18 00:15:49 PPT RABN: 000005FB
ADAN24 00006 2001-01-18 00:15:49 Number of entries: 03
ADAN24 00006 2001-01-18 00:15:49 Nucleus indicator: C0
ADAN24 00006 2001-01-18 00:15:49 NUCID: 0084
ADAN24 00006 2001-01-18 00:15:49 PPT Entry length: 0025
ADAN24 00006 2001-01-18 00:15:49 Entry ID: W
ADAN24 00006 2001-01-18 00:15:49 Dataset=/SAGUID/DB006/Vvr/WORKR1/
ADAN24 00006 2001-01-18 00:15:49 PPT Entry length: 0026
ADAN24 00006 2001-01-18 00:15:49 Entry ID: 1
ADAN24 00006 2001-01-18 00:15:49 Dataset=/SAGUID/DB006/Vvr/PLOGR11/
```

```

ADAN24 00006 2001-01-18 00:15:49 PPT Entry length: 0026
ADAN24 00006 2001-01-18 00:15:49 Entry ID: 2
ADAN24 00006 2001-01-18 00:15:49 Dataset=/SAGUID/DB006/Vvr/PLOGR12/
ADAN24 00006 2001-01-18 00:15:49

```

```

ADAN24 00006 2001-01-18 00:15:49 PPT RABN: 000005FC
ADAN24 00006 2001-01-18 00:15:49 Number of entries: 03
ADAN24 00006 2001-01-18 00:15:49 Nucleus indicator: C0
ADAN24 00006 2001-01-18 00:15:49 NUCID: 0003
ADAN24 00006 2001-01-18 00:15:49 PPT Entry length: 0025
ADAN24 00006 2001-01-18 00:15:49 Entry ID: W
ADAN24 00006 2001-01-18 00:15:49 Dataset=/SAGUID/DB006/Vvr/WORKR2/
ADAN24 00006 2001-01-18 00:15:49 PPT Entry length: 0026
ADAN24 00006 2001-01-18 00:15:49 Entry ID: 1
ADAN24 00006 2001-01-18 00:15:49 Dataset=/SAGUID/DB006/Vvr/PLOGR21/
ADAN24 00006 2001-01-18 00:15:49 PPT Entry length: 0026
ADAN24 00006 2001-01-18 00:15:49 Entry ID: 2
ADAN24 00006 2001-01-18 00:15:49 Dataset=/SAGUID/DB006/Vvr/PLOGR22/
ADAN41 00006 2001-01-18 00:15:49 Function completed

```

DXCACHE - Display Cache Statistics

Use `DXCACHE` to display the primary cache-related statistics.

The full set of statistics shown in this output is displayed only for users who have the selectable unit Adabas Online System (AOS) installed.

If you have installed only the demo version of AOS delivered with Adabas, only the statistics in the sections Totals, Data Storage, and Normal Index are displayed using this command.

Sample Output

```

ADAX61 00006 2000-09-06 19:29:23 External cache
statistics:
ADAX61 00006 2000-09-06 19:29:23

```

Cast-out Directory

```

ADAX61 00006 2000-09-06 19:29:23 Cast-out dir      :      35
ADAX61 00006 2000-09-06 19:29:23 Synchronous   :       0
ADAX61 00006 2000-09-06 19:29:23 Asynchronous  :      35
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Unlock cast-out:      35
ADAX61 00006 2000-09-06 19:29:23 Synchronous   :       1
ADAX61 00006 2000-09-06 19:29:23 Asynchronous  :      34
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Directory reads:       1
ADAX61 00006 2000-09-06 19:29:23 Synchronous   :       0
ADAX61 00006 2000-09-06 19:29:23 Asynchronous  :       1

```

```
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
```

Totals

```
ADAX61 00006 2000-09-06 19:29:23 Totals:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads          :          1,681
ADAX61 00006 2000-09-06 19:29:23 Synchronous    :             71
ADAX61 00006 2000-09-06 19:29:23 Asynchronous    :          1,610
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 In cache        :             888
ADAX61 00006 2000-09-06 19:29:23 Not in cache    :             793
ADAX61 00006 2000-09-06 19:29:23 Structure full:              0
```

```
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Writes          :          25,467
ADAX61 00006 2000-09-06 19:29:23 Synchronous     :          22,724
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :           2,743
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Written          :          25,467
ADAX61 00006 2000-09-06 19:29:23 Not written      :              0
ADAX61 00006 2000-09-06 19:29:23 Structure full:              0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates        :          65,552
ADAX61 00006 2000-09-06 19:29:23 Block invalid    :              0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads   :           1,727
ADAX61 00006 2000-09-06 19:29:23 Synchronous      :             265
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :           1,462
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes          :              0
ADAX61 00006 2000-09-06 19:29:23 Timeouts         :              0
ADAX61 00006 2000-09-06 19:29:23
```

Address Converter

```
ADAX61 00006 2000-09-06 19:29:23 AC:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads          :             11
ADAX61 00006 2000-09-06 19:29:23 Synchronous     :             11
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :              0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 In cache        :              6
ADAX61 00006 2000-09-06 19:29:23 Not in cache    :              5
ADAX61 00006 2000-09-06 19:29:23 Structure full:              0
ADAX61 00006 2000-09-06 19:29:23
```


ADAX61	00006	2000-09-06	19:29:23	Writes	:	2,644
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	2,608
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	36
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Written	:	2,644
ADAX61	00006	2000-09-06	19:29:23	Not written	:	0
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Validates	:	8,772
ADAX61	00006	2000-09-06	19:29:23	Block invalid	:	0
ADAX61	00006	2000-09-06	19:29:23			

ADAX61	00006	2000-09-06	19:29:23	Cast-out reads	:	38
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	38
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Deletes	:	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts	:	0
ADAX61	00006	2000-09-06	19:29:23			

Data Storage

ADAX61	00006	2000-09-06	19:29:23	DS:		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Reads	:	1,609
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	1,609
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	In cache	:	855
ADAX61	00006	2000-09-06	19:29:23	Not in cache	:	754
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Writes	:	2,645
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	2,645
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Written	:	2,645
ADAX61	00006	2000-09-06	19:29:23	Not written	:	0
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Validates	:	6,603
ADAX61	00006	2000-09-06	19:29:23	Block invalid	:	0
ADAX61	00006	2000-09-06	19:29:23			

ADAX61	00006	2000-09-06	19:29:23	Cast-out reads :	1,461
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	1,461
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Deletes :	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts :	0
ADAX61	00006	2000-09-06	19:29:23		

Data Storage Space Table

ADAX61	00006	2000-09-06	19:29:23	DSST:	
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Reads :	1
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	1
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	In cache :	0
ADAX61	00006	2000-09-06	19:29:23	Not in cache :	1
ADAX61	00006	2000-09-06	19:29:23	Structure full:	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Writes :	2,644
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	2,622
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	22
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Written :	2,644
ADAX61	00006	2000-09-06	19:29:23	Not written :	0
ADAX61	00006	2000-09-06	19:29:23	Structure full:	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Validates :	3,969
ADAX61	00006	2000-09-06	19:29:23	Block invalid :	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Cast-out reads :	34
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	33
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	1
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Deletes :	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts :	0
ADAX61	00006	2000-09-06	19:29:23		

File Control Block

ADAX61	00006	2000-09-06	19:29:23	FCB:
ADAX61	00006	2000-09-06	19:29:23	
ADAX61	00006	2000-09-06	19:29:23	

ADAX61	00006	2000-09-06	19:29:23	Reads	:	2
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	2
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	In cache	:	0
ADAX61	00006	2000-09-06	19:29:23	Not in cache	:	2
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Writes	:	2,132
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	2,123
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	9
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Written	:	2,132
ADAX61	00006	2000-09-06	19:29:23	Not written	:	0
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Validates	:	17,000
ADAX61	00006	2000-09-06	19:29:23	Block invalid	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Cast-out reads	:	33
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	33
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Deletes	:	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts	:	0
ADAX61	00006	2000-09-06	19:29:23			

Normal Index

ADAX61	00006	2000-09-06	19:29:23	NI:		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Reads	:	50
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	49
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	1
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	In cache	:	25
ADAX61	00006	2000-09-06	19:29:23	Not in cache	:	25
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Writes	:	7,767
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	7,747
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	20
ADAX61	00006	2000-09-06	19:29:23			

ADAX61	00006	2000-09-06	19:29:23	Written	:	7,767
ADAX61	00006	2000-09-06	19:29:23	Not written	:	0
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Validates	:	7,273
ADAX61	00006	2000-09-06	19:29:23	Block invalid	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Cast-out reads	:	101
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	101
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Deletes	:	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts	:	0
ADAX61	00006	2000-09-06	19:29:23			

Upper Index

ADAX61	00006	2000-09-06	19:29:23	UI:		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Reads	:	8
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	8
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	In cache	:	2
ADAX61	00006	2000-09-06	19:29:23	Not in cache	:	6
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Writes	:	7,635
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	7,624
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	11
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Written	:	7,635
ADAX61	00006	2000-09-06	19:29:23	Not written	:	0
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Validates	:	21,935
ADAX61	00006	2000-09-06	19:29:23	Block invalid	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Cast-out reads	:	60
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	60
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Deletes	:	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts	:	0
ADAX61	00006	2000-09-06	19:29:23			

File Statistics

```

ADAX61 00006 2000-09-06 19:29:23 File statistics forfiles with over 25
ADAX61 00006 2000-09-06 19:29:23 percent of the total cache statistics:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 File      1:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads           :           1,672
ADAX61 00006 2000-09-06 19:29:23 Writes          :           22,798
ADAX61 00006 2000-09-06 19:29:23 Validates       :           61,531
ADAX61 00006 2000-09-06 19:29:23

```

DXFILE - Display Cache Statistics for Files

Use `DXFILE` to display cache-related statistics for 1 to 5 files.

The command is specified using the following format:

```
DXFILE= fnr [,fnr1 ]...
```

Sample Output

```

ADAI29 OPER CMD: DXFILE=0,1,2,3,9
ADAX61 00006 2000-09-06 19:30:38

```

File 0

```

ADAX61 00006 2000-09-06 19:30:38 File      0:
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Reads           :           1
ADAX61 00006 2000-09-06 19:30:38 Synchronous     :           1
ADAX61 00006 2000-09-06 19:30:38 Asynchronous    :           0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 In cache        :           0
ADAX61 00006 2000-09-06 19:30:38 Not in cache    :           1
ADAX61 00006 2000-09-06 19:30:38 Structure full:           0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Writes          :           2,644
ADAX61 00006 2000-09-06 19:30:38 Synchronous     :           2,622
ADAX61 00006 2000-09-06 19:30:38 Asynchronous    :           22
ADAX61 00006 2000-09-06 19:30:38

```

```

ADAX61 00006 2000-09-06 19:30:38 Written      :          2,644
ADAX61 00006 2000-09-06 19:30:38 Not written :           0
ADAX61 00006 2000-09-06 19:30:38 Structure full:          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Validates      :          3,969
ADAX61 00006 2000-09-06 19:30:38 Block invalid :           0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Cast-out reads :           34
ADAX61 00006 2000-09-06 19:30:38 Synchronous   :           33
ADAX61 00006 2000-09-06 19:30:38 Asynchronous   :           1
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Deletes        :           0
ADAX61 00006 2000-09-06 19:30:38 Timeouts       :           0
ADAX61 00006 2000-09-06 19:30:38

```

File 1

```

ADAX61 00006 2000-09-06 19:30:38 File      1:
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Reads       :          1,672
ADAX61 00006 2000-09-06 19:30:38 Synchronous :           64
ADAX61 00006 2000-09-06 19:30:38 Asynchronous :          1,608
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 In cache    :           888
ADAX61 00006 2000-09-06 19:30:38 Not in cache :          784
ADAX61 00006 2000-09-06 19:30:38 Structure full:          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Writes      :          22,798
ADAX61 00006 2000-09-06 19:30:38 Synchronous :          20,082
ADAX61 00006 2000-09-06 19:30:38 Asynchronous :           2,716
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Written     :          22,798
ADAX61 00006 2000-09-06 19:30:38 Not written :           0
ADAX61 00006 2000-09-06 19:30:38 Structure full:          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Validates    :          61,531
ADAX61 00006 2000-09-06 19:30:38 Block invalid :           0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Cast-out reads :          1,677
ADAX61 00006 2000-09-06 19:30:38 Synchronous   :           221
ADAX61 00006 2000-09-06 19:30:38 Asynchronous   :          1,456
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Deletes       :           0
ADAX61 00006 2000-09-06 19:30:38 Timeouts      :           0
ADAX61 00006 2000-09-06 19:30:38

```

File 2

```

ADAX61 00006 2000-09-06 19:30:38 File      2:
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Reads          :          0
ADAX61 00006 2000-09-06 19:30:38 Synchronous    :          0
ADAX61 00006 2000-09-06 19:30:38 Asynchronous   :          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 In cache       :          0
ADAX61 00006 2000-09-06 19:30:38 Not in cache   :          0
ADAX61 00006 2000-09-06 19:30:38 Structure full:          0
ADAX61 00006 2000-09-06 19:30:38

ADAX61 00006 2000-09-06 19:30:38 Writes          :          0
ADAX61 00006 2000-09-06 19:30:38 Synchronous     :          0
ADAX61 00006 2000-09-06 19:30:38 Asynchronous    :          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Written         :          0
ADAX61 00006 2000-09-06 19:30:38 Not written     :          0
ADAX61 00006 2000-09-06 19:30:38 Structure full:          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Validates       :          0
ADAX61 00006 2000-09-06 19:30:38 Block invalid  :          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Cast-out reads :          0
ADAX61 00006 2000-09-06 19:30:38 Synchronous     :          0
ADAX61 00006 2000-09-06 19:30:38 Asynchronous    :          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Deletes         :          0
ADAX61 00006 2000-09-06 19:30:38 Timeouts        :          0
ADAX61 00006 2000-09-06 19:30:38

```

File 3

```

ADAX61 00006 2000-09-06 19:30:38 File      3:
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Reads          :          0
ADAX61 00006 2000-09-06 19:30:38 Synchronous    :          0
ADAX61 00006 2000-09-06 19:30:38 Asynchronous   :          0
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 In cache       :          0
ADAX61 00006 2000-09-06 19:30:38 Not in cache   :          0
ADAX61 00006 2000-09-06 19:30:38 Structure full:          0
ADAX61 00006 2000-09-06 19:30:38

```

ADAX61	00006	2000-09-06	19:30:38	Writes	:	0
ADAX61	00006	2000-09-06	19:30:38	Synchronous	:	0
ADAX61	00006	2000-09-06	19:30:38	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	Written	:	0
ADAX61	00006	2000-09-06	19:30:38	Not written	:	0
ADAX61	00006	2000-09-06	19:30:38	Structure full:		0
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	Validates	:	0
ADAX61	00006	2000-09-06	19:30:38	Block invalid	:	0
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	Cast-out reads	:	0
ADAX61	00006	2000-09-06	19:30:38	Synchronous	:	0
ADAX61	00006	2000-09-06	19:30:38	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	Deletes	:	0
ADAX61	00006	2000-09-06	19:30:38	Timeouts	:	0
ADAX61	00006	2000-09-06	19:30:38			

File 9

ADAX61	00006	2000-09-06	19:30:38	File	9:	
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	Reads	:	8
ADAX61	00006	2000-09-06	19:30:38	Synchronous	:	6
ADAX61	00006	2000-09-06	19:30:38	Asynchronous	:	2
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	In cache	:	0
ADAX61	00006	2000-09-06	19:30:38	Not in cache	:	8
ADAX61	00006	2000-09-06	19:30:38	Structure full:		0
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	Writes	:	25
ADAX61	00006	2000-09-06	19:30:38	Synchronous	:	20
ADAX61	00006	2000-09-06	19:30:38	Asynchronous	:	5
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	Written	:	25
ADAX61	00006	2000-09-06	19:30:38	Not written	:	0
ADAX61	00006	2000-09-06	19:30:38	Structure full:		0
ADAX61	00006	2000-09-06	19:30:38			
ADAX61	00006	2000-09-06	19:30:38	Validates	:	52
ADAX61	00006	2000-09-06	19:30:38	Block invalid	:	0
ADAX61	00006	2000-09-06	19:30:38			


```

ADAX61 00006 2000-09-06 19:30:38 Cast-out reads : 16
ADAX61 00006 2000-09-06 19:30:38 Synchronous : 11
ADAX61 00006 2000-09-06 19:30:38 Asynchronous : 5
ADAX61 00006 2000-09-06 19:30:38
ADAX61 00006 2000-09-06 19:30:38 Deletes : 0
ADAX61 00006 2000-09-06 19:30:38 Timeouts : 0
ADAN41 00006 2000-09-06 19:30:38 Function completed

```

DXLOCK - Display Lock Statistics

Use DXLOCK to display lock-related statistics.

Sample Output

```

ADAX61 00006 2000-09-06 19:29:23 External lock statistics:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23

```

General Control Block Lock

```

ADAX61 00006 2000-09-06 19:29:23 1. GCB lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional : 0
ADAX61 00006 2000-09-06 19:29:23           Granted : 0
ADAX61 00006 2000-09-06 19:29:23           Rejected : 0
ADAX61 00006 2000-09-06 19:29:23           Unconditional : 0
ADAX61 00006 2000-09-06 19:29:23           Synchronous : 0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued : 0
ADAX61 00006 2000-09-06 19:29:23           Synchronous : 0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23

```

Security Lock

```

ADAX61 00006 2000-09-06 19:29:23 2. Security lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional : 0
ADAX61 00006 2000-09-06 19:29:23           Granted : 0
ADAX61 00006 2000-09-06 19:29:23           Rejected : 0
ADAX61 00006 2000-09-06 19:29:23           Unconditional : 0
ADAX61 00006 2000-09-06 19:29:23           Synchronous : 0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23

```

```

ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23

```

FST Lock

```

ADAX61 00006 2000-09-06 19:29:23 3. FST lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional :      0
ADAX61 00006 2000-09-06 19:29:23      Granted          :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected         :      0
ADAX61 00006 2000-09-06 19:29:23      Unconditional    :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous     :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous     :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23

```

File Lock Table Lock

```

ADAX61 00006 2000-09-06 19:29:23 4. File-lock-table lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional :      0
ADAX61 00006 2000-09-06 19:29:23      Granted          :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected         :      0
ADAX61 00006 2000-09-06 19:29:23      Unconditional    :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous     :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous     :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23

```

Online Save Lock

```

ADAX61 00006 2000-09-06 19:29:23 5. Online save lock
ADAX61 00006 2000-09-06 19:29:23

```

ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			

ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

Buffer Flush Lock

ADAX61	00006	2000-09-06	19:29:23	6. Buffer flush lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	38
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	38
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	38
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	38
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

Global ET Sync Lock

ADAX61	00006	2000-09-06	19:29:23	7. Global ET sync lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

Recovery Lock

ADAX61	00006	2000-09-06	19:29:23	8. Recovery lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

Hold ISN Locks

ADAX61	00006	2000-09-06	19:29:23	9. Hold ISN locks		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	3100
ADAX61	00006	2000-09-06	19:29:23	Granted	:	3100
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	3100
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	3100
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	3100
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

Unique Descriptor Locks

ADAX61	00006	2000-09-06	19:29:23	10. Unique descriptor locks		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	1
ADAX61	00006	2000-09-06	19:29:23	Granted	:	1
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	1
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			

ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	1
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	1
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

ETID Locks

ADAX61	00006	2000-09-06	19:29:23	11. ETID locks		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	1
ADAX61	00006	2000-09-06	19:29:23	Granted	:	1
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	1
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

New Data RABN Locks

ADAX61	00006	2000-09-06	19:29:23	12. New-Data-RABN locks		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

Checkpoint Lock

ADAX61	00006	2000-09-06	19:29:23	13. Checkpoint lock		
ADAX61	00006	2000-09-06	19:29:23			

ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	6
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	6
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0

ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	6
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	6
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

ET Data Lock

ADAX61	00006	2000-09-06	19:29:23	14. ET data lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

Global Update Command Sync Lock

ADAX61	00006	2000-09-06	19:29:23	15. Global update command sync lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	33
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	33
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	33
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	33
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			

Parameter Lock

```

ADAX61 00006 2000-09-06 19:29:23 16. Parameter lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :      0
ADAX61 00006 2000-09-06 19:29:23           Granted      :      0
ADAX61 00006 2000-09-06 19:29:23           Rejected     :      0
ADAX61 00006 2000-09-06 19:29:23           Unconditional :      0
ADAX61 00006 2000-09-06 19:29:23           Synchronous  :      0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23           Synchronous  :      0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous :      0
ADAX61 00006 2000-09-06 19:29:23
ADAN41 00006 2000-09-06 19:29:23 Function completed

```

DBID Target Assignment Lock

```

ADAX61 00006 2000-09-06 19:29:23 17.DBID target assignment lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :      0
ADAX61 00006 2000-09-06 19:29:23           Granted      :      0
ADAX61 00006 2000-09-06 19:29:23           Rejected     :      0
ADAX61 00006 2000-09-06 19:29:23           Unconditional :      2
ADAX61 00006 2000-09-06 19:29:23           Synchronous  :      2
ADAX61 00006 2000-09-06 19:29:23           Asynchronous :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      2
ADAX61 00006 2000-09-06 19:29:23           Synchronous  :      2
ADAX61 00006 2000-09-06 19:29:23           Asynchronous :      0
ADAX61 00006 2000-09-06 19:29:23
ADAN41 00006 2000-09-06 19:29:23 Function completed

```

DXMSG - Display Messaging Performance Statistics

The DXMSG command can be used to display the messaging performance statistics available for Adabas Cluster Services.

The statistics produced by the DXMSG command can be used to determine the impact of messaging on system performance and to determine how to set the ADARUN MXMSG parameter related to the other alert and timeout enhancements in Adabas Cluster Services.

Sample Output

```

17:28:45 ADAI29 Oper cmd: DXMSG
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messaging statistics
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message Control Block statistics
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Allocated 224
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Used 2
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Total requests 1,567
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Statistics for ACMD-type messages
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages sent 771
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages arrived 796
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages accepted 796
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Replies sent 794
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Statistics for XCF transport service
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages subject to MXMSG
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message duration in seconds
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Minimum 0.000021
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Maximum 0.050979
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Mean 0.004110
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Std Dev 0.004507
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message count 686
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1000 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 10 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 ms 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 10 ms 8% 56
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1 ms 81% 553
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 us 11% 74
17:28:45 ADAX16 00226 2007-06-01 17:28:44 <= 100 us 0% 3
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages not subject to MXMSG
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message duration in seconds
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Minimum 0.000938
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Maximum 0.047989
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Mean 0.010455
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Std Dev 0.009466
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message count 83
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1000 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 10 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 ms 0% 0

```



```

17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 10 ms 37% 31
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1 ms 61% 51
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 us 1% 1
17:28:45 ADAX16 00226 2007-06-01 17:28:44 <= 100 us 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Total for all messages
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message duration in seconds
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Minimum 0.000021
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Maximum 0.050979
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Mean 0.004795
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Std Dev 0.003194
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message count 769
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1000 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 10 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1 s 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 ms 0% 0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 10 ms 11% 87
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1 ms 79% 604
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 100 us 10% 75
17:28:45 ADAX16 00226 2007-06-01 17:28:44 <= 100 us 0% 3
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAN41 00226 2007-06-01 17:28:44 Function completed

```

The performance statistics are split into those that are subject to the ADARUN `MXMSG` parameter setting and those that are not; after each is reported separately in the output, a combined report is provided containing the summarization of the two for all messages.

In the DXMSG output, "Mean" refers to the average message duration (arithmetic mean of all the messages measured) and "Std Dev" refers to the related standard deviation (average deviation from the mean). The unit symbols used in the statistics are "s" for seconds, "ms" for milliseconds, and "us" for microseconds.

DXSTAT - Display Cache and Lock Statistics

Use `DXSTAT` to display all cache- and lock-related statistics.

The full set of external cache statistics shown in this output is displayed only for users who have the selectable unit Adabas Online System (AOS) installed.

If you have installed only the demo version of AOS delivered with Adabas, only the external cache statistics in the sections Totals, Data Storage, and Normal Index are displayed using this command. All file cache statistics for files and all external lock statistics are displayed.

Sample Output

ADAI29 OPER CMD: DXSTAT

External Cache Statistics

```

ADAX61 00006 2000-09-06 19:29:23 External cache statistics:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out dir      :          35
ADAX61 00006 2000-09-06 19:29:23 Synchronous      :           0
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :          35
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Unlock cast-out: 35
ADAX61 00006 2000-09-06 19:29:23 Synchronous      :           1
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :          34
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Directory reads: 1
ADAX61 00006 2000-09-06 19:29:23 Synchronous      :           0
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :           1
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Totals:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads            :        1,681
ADAX61 00006 2000-09-06 19:29:23 Synchronous      :           71
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :        1,610
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 In cache         :          888
ADAX61 00006 2000-09-06 19:29:23 Not in cache     :          793
ADAX61 00006 2000-09-06 19:29:23 Structure full: 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Writes           :       25,467
ADAX61 00006 2000-09-06 19:29:23 Synchronous      :       22,724
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :        2,743
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Written          :       25,467
ADAX61 00006 2000-09-06 19:29:23 Not written      :           0
ADAX61 00006 2000-09-06 19:29:23 Structure full: 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates        :      65,552
ADAX61 00006 2000-09-06 19:29:23 Block invalid   :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads   :        1,727
ADAX61 00006 2000-09-06 19:29:23 Synchronous      :          265
ADAX61 00006 2000-09-06 19:29:23 Asynchronous     :        1,462
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes          :           0
ADAX61 00006 2000-09-06 19:29:23 Timeouts         :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23

```

```

ADAX61 00006 2000-09-06 19:29:23 AC:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads           :           11
ADAX61 00006 2000-09-06 19:29:23 Synchronous    :           11
ADAX61 00006 2000-09-06 19:29:23 Asynchronous   :            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 In cache       :            6
ADAX61 00006 2000-09-06 19:29:23 Not in cache   :            5
ADAX61 00006 2000-09-06 19:29:23 Structure full:            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Writes         :          2,644
ADAX61 00006 2000-09-06 19:29:23 Synchronous    :          2,608
ADAX61 00006 2000-09-06 19:29:23 Asynchronous   :            36
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Written        :          2,644
ADAX61 00006 2000-09-06 19:29:23 Not written    :            0
ADAX61 00006 2000-09-06 19:29:23 Structure full:            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates      :          8,772
ADAX61 00006 2000-09-06 19:29:23 Block invalid :            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads :            38
ADAX61 00006 2000-09-06 19:29:23 Synchronous    :            38
ADAX61 00006 2000-09-06 19:29:23 Asynchronous   :            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes        :            0
ADAX61 00006 2000-09-06 19:29:23 Timeouts       :            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 DS:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads           :          1,609
ADAX61 00006 2000-09-06 19:29:23 Synchronous    :            0
ADAX61 00006 2000-09-06 19:29:23 Asynchronous   :          1,609
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 In cache       :            855
ADAX61 00006 2000-09-06 19:29:23 Not in cache   :            754
ADAX61 00006 2000-09-06 19:29:23 Structure full:            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Writes         :          2,645
ADAX61 00006 2000-09-06 19:29:23 Synchronous    :            0
ADAX61 00006 2000-09-06 19:29:23 Asynchronous   :          2,645
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Written        :          2,645
ADAX61 00006 2000-09-06 19:29:23 Not written    :            0
ADAX61 00006 2000-09-06 19:29:23 Structure full:            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates      :          6,603
ADAX61 00006 2000-09-06 19:29:23 Block invalid :            0
ADAX61 00006 2000-09-06 19:29:23

```

ADAX61	00006	2000-09-06	19:29:23	Cast-out reads :	1,461
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	1,461
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Deletes :	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts :	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	DSST:	
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Reads :	1
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	1
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	In cache :	0
ADAX61	00006	2000-09-06	19:29:23	Not in cache :	1
ADAX61	00006	2000-09-06	19:29:23	Structure full:	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Writes :	2,644
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	2,622
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	22
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Written :	2,644
ADAX61	00006	2000-09-06	19:29:23	Not written :	0
ADAX61	00006	2000-09-06	19:29:23	Structure full:	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Validates :	3,969
ADAX61	00006	2000-09-06	19:29:23	Block invalid :	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Cast-out reads :	34
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	33
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	1
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Deletes :	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts :	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	FCB:	
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Reads :	2
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	2
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	In cache :	0
ADAX61	00006	2000-09-06	19:29:23	Not in cache :	2
ADAX61	00006	2000-09-06	19:29:23	Structure full:	0
ADAX61	00006	2000-09-06	19:29:23		
ADAX61	00006	2000-09-06	19:29:23	Writes :	2,132
ADAX61	00006	2000-09-06	19:29:23	Synchronous :	2,123
ADAX61	00006	2000-09-06	19:29:23	Asynchronous :	9

```

ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Written      :           2,132
ADAX61 00006 2000-09-06 19:29:23   Not written  :           0
ADAX61 00006 2000-09-06 19:29:23   Structure full:           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Validates    :          17,000
ADAX61 00006 2000-09-06 19:29:23   Block invalid :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Cast-out reads :           33
ADAX61 00006 2000-09-06 19:29:23   Synchronous   :           33
ADAX61 00006 2000-09-06 19:29:23   Asynchronous  :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Deletes       :           0
ADAX61 00006 2000-09-06 19:29:23   Timeouts      :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   NI:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Reads         :           50
ADAX61 00006 2000-09-06 19:29:23   Synchronous   :           49
ADAX61 00006 2000-09-06 19:29:23   Asynchronous  :           1
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   In cache      :           25
ADAX61 00006 2000-09-06 19:29:23   Not in cache  :           25
ADAX61 00006 2000-09-06 19:29:23   Structure full:           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Writes        :          7,767
ADAX61 00006 2000-09-06 19:29:23   Synchronous   :          7,747
ADAX61 00006 2000-09-06 19:29:23   Asynchronous  :           20
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Written      :          7,767
ADAX61 00006 2000-09-06 19:29:23   Not written  :           0
ADAX61 00006 2000-09-06 19:29:23   Structure full:           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Validates    :          7,273
ADAX61 00006 2000-09-06 19:29:23   Block invalid :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Cast-out reads :          101
ADAX61 00006 2000-09-06 19:29:23   Synchronous   :          101
ADAX61 00006 2000-09-06 19:29:23   Asynchronous  :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Deletes       :           0
ADAX61 00006 2000-09-06 19:29:23   Timeouts      :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   UI:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23   Reads         :           8
ADAX61 00006 2000-09-06 19:29:23   Synchronous   :           8
ADAX61 00006 2000-09-06 19:29:23   Asynchronous  :           0

```

```

ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 In cache : 2
ADAX61 00006 2000-09-06 19:29:23 Not in cache : 6
ADAX61 00006 2000-09-06 19:29:23 Structure full: 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Writes : 7,635
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 7,624
ADAX61 00006 2000-09-06 19:29:23 Asynchronous : 11
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Written : 7,635
ADAX61 00006 2000-09-06 19:29:23 Not written : 0
ADAX61 00006 2000-09-06 19:29:23 Structure full: 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates : 21,935
ADAX61 00006 2000-09-06 19:29:23 Block invalid : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads : 60
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 60
ADAX61 00006 2000-09-06 19:29:23 Asynchronous : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes : 0
ADAX61 00006 2000-09-06 19:29:23 Timeouts : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 File statistics for files with over 25
ADAX61 00006 2000-09-06 19:29:23 percent of the total cache statistics:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 File 1:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads : 1,672
ADAX61 00006 2000-09-06 19:29:23 Writes : 22,798
ADAX61 00006 2000-09-06 19:29:23 Validates : 61,531
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23

```

File Cache Statistics for Files

```

ADAX61 00006 2000-09-06 19:29:23 File 0:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads : 1
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 1
ADAX61 00006 2000-09-06 19:29:23 Asynchronous : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 In cache : 0
ADAX61 00006 2000-09-06 19:29:23 Not in cache : 1
ADAX61 00006 2000-09-06 19:29:23 Structure full: 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Writes : 2,644
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 2,622

```

```

ADAX61 00006 2000-09-06 19:29:23 Asynchronous : 22
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Written : 2,644
ADAX61 00006 2000-09-06 19:29:23 Not written : 0
ADAX61 00006 2000-09-06 19:29:23 Structure full: 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates : 3,969
ADAX61 00006 2000-09-06 19:29:23 Block invalid : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads : 34
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 33
ADAX61 00006 2000-09-06 19:29:23 Asynchronous : 1
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes : 0
ADAX61 00006 2000-09-06 19:29:23 Timeouts : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 File 1:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads : 1,672
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 64
ADAX61 00006 2000-09-06 19:29:23 Asynchronous : 1,608
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 In cache : 888
ADAX61 00006 2000-09-06 19:29:23 Not in cache : 784
ADAX61 00006 2000-09-06 19:29:23 Structure full: 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Writes : 22,798
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 20,082
ADAX61 00006 2000-09-06 19:29:23 Asynchronous : 2,716
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Written : 22,798
ADAX61 00006 2000-09-06 19:29:23 Not written : 0
ADAX61 00006 2000-09-06 19:29:23 Structure full: 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates : 61,531
ADAX61 00006 2000-09-06 19:29:23 Block invalid : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads : 1,677
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 221
ADAX61 00006 2000-09-06 19:29:23 Asynchronous : 1,456
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes : 0
ADAX61 00006 2000-09-06 19:29:23 Timeouts : 0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 File 9:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads : 8
ADAX61 00006 2000-09-06 19:29:23 Synchronous : 6

```

ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	2
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	In cache	:	0
ADAX61	00006	2000-09-06	19:29:23	Not in cache	:	8
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Writes	:	25
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	20
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	5
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Written	:	25
ADAX61	00006	2000-09-06	19:29:23	Not written	:	0
ADAX61	00006	2000-09-06	19:29:23	Structure full:		0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Validates	:	52
ADAX61	00006	2000-09-06	19:29:23	Block invalid	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Cast-out reads	:	16
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	11
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	5
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Deletes	:	0
ADAX61	00006	2000-09-06	19:29:23	Timeouts	:	0

External Lock Statistics

ADAX61	00006	2000-09-06	19:29:23	External lock statistics:		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	1. GCB lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	2. Security lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			


```

ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 3. FST lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional :      0
ADAX61 00006 2000-09-06 19:29:23      Granted          :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected          :      0
ADAX61 00006 2000-09-06 19:29:23      Unconditional     :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous       :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 4. File-lock-table lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional :      0
ADAX61 00006 2000-09-06 19:29:23      Granted          :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected          :      0
ADAX61 00006 2000-09-06 19:29:23      Unconditional     :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous       :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 5. Online save lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional :      0
ADAX61 00006 2000-09-06 19:29:23      Granted          :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected          :      0
ADAX61 00006 2000-09-06 19:29:23      Unconditional     :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous       :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 6. Buffer flush lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional :      0
ADAX61 00006 2000-09-06 19:29:23      Granted          :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected          :      0

```

ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	38
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	38
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	38
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	38
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	7. Global ET sync lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	8. Recovery lock		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Granted	:	0
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	0
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	9. Hold ISN locks		
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Obtains - Conditional	:	3100
ADAX61	00006	2000-09-06	19:29:23	Granted	:	3100
ADAX61	00006	2000-09-06	19:29:23	Rejected	:	0
ADAX61	00006	2000-09-06	19:29:23	Unconditional	:	0
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	3100
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	Releases - Issued	:	3100
ADAX61	00006	2000-09-06	19:29:23	Synchronous	:	3100
ADAX61	00006	2000-09-06	19:29:23	Asynchronous	:	0
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23			
ADAX61	00006	2000-09-06	19:29:23	10. Unique descriptor locks		

```

ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :      1
ADAX61 00006 2000-09-06 19:29:23           Granted        :      1
ADAX61 00006 2000-09-06 19:29:23           Rejected       :      0
ADAX61 00006 2000-09-06 19:29:23           Unconditional   :      0
ADAX61 00006 2000-09-06 19:29:23           Synchronous    :      1
ADAX61 00006 2000-09-06 19:29:23           Asynchronous   :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued       :      1
ADAX61 00006 2000-09-06 19:29:23           Synchronous    :      1
ADAX61 00006 2000-09-06 19:29:23           Asynchronous   :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 11. ETID locks
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional   :      1
ADAX61 00006 2000-09-06 19:29:23           Granted        :      1
ADAX61 00006 2000-09-06 19:29:23           Rejected       :      0
ADAX61 00006 2000-09-06 19:29:23           Unconditional   :      0
ADAX61 00006 2000-09-06 19:29:23           Synchronous    :      1
ADAX61 00006 2000-09-06 19:29:23           Asynchronous   :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued       :      0
ADAX61 00006 2000-09-06 19:29:23           Synchronous    :      0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous   :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 12. New-Data-RABN locks
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional   :      0
ADAX61 00006 2000-09-06 19:29:23           Granted        :      0
ADAX61 00006 2000-09-06 19:29:23           Rejected       :      0
ADAX61 00006 2000-09-06 19:29:23           Unconditional   :      0
ADAX61 00006 2000-09-06 19:29:23           Synchronous    :      0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous   :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued       :      0
ADAX61 00006 2000-09-06 19:29:23           Synchronous    :      0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous   :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 13. Checkpoint lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional   :      0
ADAX61 00006 2000-09-06 19:29:23           Granted        :      0
ADAX61 00006 2000-09-06 19:29:23           Rejected       :      0
ADAX61 00006 2000-09-06 19:29:23           Unconditional   :      6
ADAX61 00006 2000-09-06 19:29:23           Synchronous    :      6
ADAX61 00006 2000-09-06 19:29:23           Asynchronous   :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued       :      6
ADAX61 00006 2000-09-06 19:29:23           Synchronous    :      6

```

```

ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 14. ET data lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :      0
ADAX61 00006 2000-09-06 19:29:23      Granted      :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected      :      0
ADAX61 00006 2000-09-06 19:29:23      Unconditional      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 15. Global update command sync loc
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :      0
ADAX61 00006 2000-09-06 19:29:23      Granted      :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected      :      0
ADAX61 00006 2000-09-06 19:29:23      Unconditional      :      33
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      33
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      33
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      33
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 16. Parameter lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :      0
ADAX61 00006 2000-09-06 19:29:23      Granted      :      0
ADAX61 00006 2000-09-06 19:29:23      Rejected      :      0
ADAX61 00006 2000-09-06 19:29:23      Unconditional      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued      :      0
ADAX61 00006 2000-09-06 19:29:23      Synchronous      :      0
ADAX61 00006 2000-09-06 19:29:23      Asynchronous      :      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAN41 00006 2000-09-06 19:29:23 Function completed

```

MXCANCEL - Dynamically Change MXCANCEL Parameter Setting

The `MXCANCEL` command can be used to dynamically change (while the Adabas nucleus is running) the setting of the `MXCANCEL` ADARUN parameter for a nucleus. This command can be used to specify the maximum number of seconds for which one cluster nucleus waits for the termination of another nucleus in the cluster that it has canceled for not responding to an internal, intracluster command. Valid values range from 15 through 2,147,483,647.

If the new `MXCANCEL` value is less than or equal to the current `MXCANCELWARN` setting, the `MXCANCELWARN` setting is automatically set to zero (0). (You can then use the `MXCANCELWARN` command to dynamically set the `MXCANCELWARN` setting appropriately.)

For more information about the `MXCANCEL` setting for a nucleus, read about the ADARUN `MXCANCEL` parameter in [*MXCANCEL: Timeout Threshold for Canceled Peer Nucleus*](#).

MXCANCELWARN - Dynamically Change MXCANCELWARN Parameter Setting

The `MXCANCELWARN` command can be used to dynamically change (while the Adabas nucleus is running) the setting of the `MXCANCELWARN` ADARUN parameter for a nucleus. This command can be used to specify the number of seconds after an intracluster cancellation request is made when the cluster nucleus requesting the cancellation should issue warnings about the inability of the canceled peer nucleus to terminate. Valid values range from 4 through 2,147,483,647 or zero (0).

The value of the `MXCANCELWARN` setting must be less than the value of the `MXCANCEL` setting. If `MXCANCELWARN` is set to a value greater than or equal to the value specified for `MXCANCEL`, the value is rejected. If `MXCANCELWARN` is set to zero, no warnings are issued about canceled cluster nuclei that are slow to terminate.

For more information about the `MXCANCELWARN` setting for a nucleus, read about the ADARUN `MXCANCELWARN` parameter in [*MXCANCELWARN: Timeout Threshold for Canceled Peer Nucleus Warning*](#).

MXMSG - Dynamically Change MXMSG Parameter Setting

The `MXMSG` command can be used to dynamically change (while the Adabas nucleus is running) the setting of the `MXMSG` ADARUN parameter for a nucleus. This command can be used to specify the maximum number of seconds in which one cluster nucleus waits for the response of another nucleus in the cluster to an internal, intracluster command. Valid values range from 15 through 32,767.

If the new `MXMSG` value is less than or equal to the current `MXMSGWARN` setting, the `MXMSGWARN` setting is automatically set to zero (0). (You can then use the `MXMSGWARN` command to dynamically set the `MXMSGWARN` setting appropriately.)

For more information about the MXMSG setting for a nucleus, read about the ADARUN MXMSG parameter in *[MXMSG: Timeout Threshold for Internucleus Command Processing](#)*.

MXMSGWARN - Dynamically Change MXMSGWARN Parameter Setting

The MXMSGWARN command can be used to dynamically change (while the Adabas nucleus is running) the setting of the MXMSGWARN ADARUN parameter for a nucleus. This command can be used to specify the number of seconds after an intracluster request is made when the cluster nucleus issuing the intracluster request should issue warnings about the failure of the target cluster nucleus to respond. Valid values range from 4 through 32,767 or zero (0).

The value of MXMSGWARN must be less than the value of MXMSG. If MXMSGWARN is set to a value greater than or equal to the value specified for MXMSG, the value is rejected. If MXMSGWARN is set to zero, no warnings are issued about cluster nuclei that are slow to respond.

For more information about the MXMSGWARN setting for a nucleus, read about the ADARUN MXMSGWARN parameter in *[MXMSGWARN: Timeout Threshold for Internucleus Command Processing Warning](#)*.

MXSTATUS - Dynamically Change MXSTATUS Parameter Setting

The MXSTATUS command can be used to dynamically change (while the Adabas nucleus is running) the setting of the MXSTATUS ADARUN parameter for a nucleus. This command can be used to specify the number of seconds between member-level XCF status monitoring heartbeats. Valid values range from 15 through 21,474,836.



Note: The MXSTATUS command is only used for Adabas Cluster Services and not for Adabas Parallel Services. Adabas Parallel Services does not use XCF and ignores the setting.

The MXSTATUS command can only be used to change the value of the MXSTATUS parameter if the MXSTATUS parameter is not zero (0). This command cannot be used to turn status monitoring on (by specifying a nonzero value if the ADARUN MXSTATUS setting is zero) or to turn it off (by specifying a value of zero when the ADARUN MXSTATUS setting is nonzero). However, setting MXSTATUS to a very high value is effectively the same as turning status monitoring off (monitoring will occur very infrequently at least).

For more information about the MXSTATUS setting for a nucleus, read about the ADARUN MXSTATUS parameter in *[MXSTATUS: Member-Level XCF Status Monitoring Heartbeat Interval](#)*.

MXWTOR - Dynamically Change MXWTOR Parameter Setting

The optional MXWTOR command can be used to dynamically change (while the Adabas nucleus is running) the setting of the MXWTOR ADARUN parameter for a nucleus. This command can be used to specify the maximum number of seconds that a cluster nucleus should wait for an operator response to a self-terminating operator query. Valid values range from 15 through 64,800 or zero (0).

If MXWTOR is set to zero, the operator query does not occur (no ADAX9J messages are written) and no operator intervention is expected.

For more information about the MXWTOR setting for a nucleus, read about the ADARUN MXWTOR parameter in *[MXWTOR : Self-Termination Operator Query Interval](#)*.

9

Entire Net-Work Operator Commands

■ Overview of Entire Net-WorkCommands for z/OS Environments	111
■ Command Descriptions	112
■ XCF Line Driver Commands	125

This section contains operator commands for Entire Net-Work, and for its XCF line driver.

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

In OS/390 environments, the following operator commands are used:

```
MODIFY  
F
```

```
STOP  
P
```

The STOP (or P) command serves as an alternative to the NETEND command and terminates Entire Net-Work. Its synonyms are described as part of the following example:

Entering the following long form MODIFY commands results in the following status displays (assuming that NETWK is the name of the started task running Entire Net-Work):

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

The following two commands are equivalent ways to terminate the Entire Net-Work session:

```
F NETWK,NETEND  
P NETWK
```

The Entire Net-Work commands described in this section are similar to Adabas operator commands. The underlined portion of the command is the minimum abbreviation.

Entire Net-Work 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 the XCF line driver are described in the section [*XCF Line Driver Commands*](#).

This chapter covers the following topics:

Overview of Entire Net-Work Commands for z/OS Environments

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

Command	Argument	Action
<u>ADAEND</u>		Terminate Entire Net-Work session.
<u>CLOSE</u>	driver	Disconnect all links of a driver, then close the driver.
<u>CLOSE NETPRNT</u>		Close the NETPRNT file and route all trace and snap output to DDPRINT.
<u>CONNECT</u>	link	Reconnect a link after a disconnect or handshake error.
<u>DEFINE</u>	link	Dynamically define a new link.
<u>DISABLE</u>	link	Disable a link (link cannot accept connects).
<u>DISCONNECT</u>	link	Disconnect a link.
<u>DISPLAY</u>	parameter	Display link, nodes, targets, paths, or statistics.
<u>DUMP</u>		Snap data areas, then terminate the Entire Net-Work session.
<u>ENABLE</u>	link	Enable a link (the link can accept connects).
<u>END</u>		Terminate Entire Net-Work session.
<u>FORCE</u>	node	Broadcast a "node down" message.
<u>HALT</u>		Terminate Entire Net-Work session.
<u>HELP</u>		List available operator commands.
<u>NETEND</u>		Terminate Entire Net-Work session.
<u>OPEN</u>	driver	Reopen a driver after a close or access method failure.
<u>OPEN NETPRNT</u>		Open the NETPRNT file and route all trace and snap output to the NETPRNT file.
<u>PROBE</u>	node	Send a probe message to a node.
<u>RESUME</u>	link	Resume sending messages using this link.
<u>SET</u>	parameter	Change the values of Entire Net-Work parameters. Note: The minimum abbreviation for SET is the null string (zero characters long).
<u>SNAP</u>		Snap data areas to DDPRINT.
<u>START</u>	driver	Restart driver (then connect all links of the driver).
<u>STOP</u>	taskid	Terminate Entire Net-Work session.
<u>SUSPEND</u>	link	Stop sending messages on this link.
<u>TERMINATE</u>		Terminate Entire Net-Work session.

Command Descriptions

This section covers the following topics:

- Commands to Terminate an Entire Net-Work Session
- CLOSE - Terminate Line Driver Activity
- CLOSE NETPRNT - Close the NETPRNT File
- CONNECT - Connect a Link
- DEFINE LINK - Dynamically Define a New Link
- DISABLE - Disallow Link Connection
- DISCONNECT - Disconnect a Link
- DISPLAY - Display Information About a Network Component
- DUMP - Terminate Entire Net-Work Session with Snap Dump
- ENABLE - Enable a Previously Disabled Link
- FORCE - Broadcast a "Node Down" Message
- HELP - List the Available Entire Net-Work Operator Commands
- OPEN - Reopen a Stopped or Closed Line Driver
- OPEN NETPRNT - Open the NETPRNT File
- PROBE - Verify Node Availability
- RESUME - Allow Link to Resume Sending Queued Messages
- SET - Dynamically Change NODE Statement Parameters
- SNAP - Snap Dump Selected Data Areas
- START - Restart a Stopped or Closed Line Driver
- SUSPEND - Stop Link from Sending Queued Messages

Commands to Terminate an Entire Net-Work Session



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 the OS/390 environment.

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 - Terminate Line Driver Activity

CLOSE *drivername*

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

The driver name must be the same as was specified on the XCF DRIVER statement (that is, either XCF or XCFD). See section [*XCF DRIVER Statement*](#) for more information.

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

CLOSE NETPRNT - Close the NETPRNT File

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 - Connect a Link

CONNECT *linkname*

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

DEFINE LINK - Dynamically Define a New Link

DEFINE {**LINK** *statement* | **LINK** *linkname* **LIKE** *linkname*}

Defines a link during Entire Net-Work operation. The link statement must adhere to the format described in [XCF LINK Statement](#). The following example applies to the Entire Net-Work XCF line driver:

```
DEFINE LINK SYS01    XCF MEMBER=NODE01,PSTATS=Y,-  
RSTATS=N,STATINT=3600,TRACESIZ=4096
```

The LIKE linkname clause 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 SYS02 LIKE SYS01
```



Note: DEFINE LINK is permitted only when DEFINE=Y is specified on the NODE statement.

DISABLE - Disallow Link Connection

DISABLE *linkname*

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

DISCONNECT - Disconnect a Link

DISCONNECT *linkname*

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

DISPLAY - Display Information About a Network Component

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

Displays current information about the specified network component. Only one component type (link, node, path, or target) can be specified in a single `DISPLAY` command. The information is displayed in the form of Entire Net-Work messages. Refer to the Entire Net-Work messages and codes documentation for information about specific messages.

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

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

DISPLAY ALINKS

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 LINKS

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
```

DISPLAY NODES

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

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

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

The following is an example of `DISPLAY STATS` output for an OS/390 system. It includes a `NETB001I` and a `NETB009I` for each active buffer pool, a set of `NETB008I`, `NETB010I`, and `NETB012I` for each subpool within the buffer pools, and a `NETB013I` for each operator command issued.

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

```
NETB000I:
-----
```

DISPLAY TARGETS

The following is an example of `DISPLAY TARGETS` output:

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

DISPLAY ZAPS

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

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

DUMP - Terminate Entire Net-Work Session with Snap 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 - Enable a Previously Disabled Link



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

FORCE - Broadcast a "Node Down" Message

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

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

HELP - List the Available Entire Net-Work Operator Commands

```
HELP
```

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

OPEN - Reopen a Stopped or Closed Line Driver

```
OPEN drivename
```

Reopen an installed and 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 was specified on the `XCF DRIVER` statement (that is, either `XCF` or `XCFD`). Read *XCF DRIVER Statement* for more information.

Note that this command is currently a synonym for the `START` command. See the explanation in [XCF DRIVER Statement](#) for more information.

OPEN NETPRNT - Open the NETPRNT File

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 - Verify Node Availability

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

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

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

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

where 'nodestack-size' is twice the number specified by the `NODE` statement parameter `MAXPATH`, rounded up to the next multiple of 4. For example, if `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 5.7.1 (1999/11/10)
```

RESUME - Allow Link to Resume Sending Queued Messages

```
RESUME linkname
```

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

SET - Dynamically Change NODE Statement Parameters

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

The `SET` command can be used to change Entire Net-Work parameter settings dynamically without interrupting network operations. Multiple parameters can be specified with one `SET` command.

The `SET` command keyword itself may be omitted. For example:

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

is equivalent to

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

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

SET CQTIMER - Set Time Allowed to Retrieve Command Results

```
SET CQTIMER = seconds
```

The approximate waiting time, in seconds, allowed for a user or application to retrieve command results with a router 16-call before timeout occurs. For more information, see the [CQTIMER](#) parameter description.

SET DUMP - Set Storage Areas for a Dump

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

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 - Set Extent of Logging Activity

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

Regulates control flow and logging of selected data areas to the printer data set. For more information, refer to the [LOG](#) parameter description.

SET MAXPATH - Set Maximum Number of Links in Message Path

```
SET MAXPATH = linkcount
```

The maximum path link, specified in number of links, that a message from users on this node is expected to travel. For more information, refer to the [MAXPATH](#) parameter description.

SET MSGFORM - Set Format of Messages for Console Display and Print

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

The message format of console messages and DDPRINT output. For more information, see the [MSGFORM](#) parameter.

SET REPLYTIM - Set Time Allowed for Reply to User Request

```
SET REPLYTIM = seconds
```

The time, in seconds, that this node is to wait for a reply to a user request before timing out. For more information, refer to the [REPLYTIM](#) parameter description.

SET TRACE, SET TROFF, SET TRON - Set Extent of Tracing Activity

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


```

Sets the trace control parameters for program traces. For more information, see the [TRACE](#) statement parameters.

SET UCMSG - Set Message Case

SET UCMSG= { N | Y }

Controls whether messages are issued in uppercase or mixed case. For more information, see the [UCMSG](#) parameter.

SET ULINK - Control Links to an Adjacent Node

SET ULINK= { N | Y }

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

SNAP - Snap Dump Selected Data Areas

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

Issue a snap dump of selected data areas to the DDPRINT file and continue processing.

 **Note:** Under certain circumstances, a snap dump is performed internally at either normal or abnormal session end.

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

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

START - Restart a Stopped or Closed Line Driver

```
START drivername
```

Restart the 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 was specified on the XCF DRIVER statement (that is, either XCF or XCFD). Read [XCF DRIVER Statement](#) for more information.

The `START` command is a synonym for the `OPEN` command.

SUSPEND - Stop Link from Sending Queued Messages

```
SUSPEND linkname
```

Instructs the specified link to stop sending messages. However, Entire Net-Work can still queue messages on this link. The `SUSPEND` command is valid only if the link is active.

XCF Line Driver Commands

The Entire Net-Work XCF line driver has the ability to process operator commands that are directed to a specific link, to all links, or directly to the driver.

Format

The XCF line driver operator commands have the following format:

```
XCFD target cmd1 cmd2 ... cmdx
```

where

- 'XCFD' tells Entire Net-Work that the command is destined for the XCF line driver. Read [XCF DRIVER Statement](#) for more information.

- 'target' identifies the target of the command to XCF and has the following format:

*target = { * | # | linkname }*

XCF line driver operator commands depend on whether the target is the driver of one or more links. Specify

- "*" if the target is all links.
 - "#" if the target is the driver itself.
 - the link name if the target is a specific link.
- 'cmd1 cmd2 cmdx' are the operator commands to be issued.

Multiple commands can be specified in a single command statement. When the ALTER command is specified, it must be the last command in the statement, because everything following the ALTER command is treated as a DRIVER or LINK statement parameter.

Command Overview

The Entire Net-Work XCF line driver supports the commands listed in the following table. All commands are available for both link and driver targets. The underlined portion of the command is the minimum abbreviation.

Command	Action
<u>ALTER</u>	Dynamically changes the driver or link configuration.
<u>HELP</u>	Lists available operator commands.
<u>RESET</u>	Resets all statistics for the driver or link.
<u>SHOW</u>	Displays parameter settings for the driver or link.
<u>SNAP</u>	For driver targets, snaps driver-specific data areas. For link targets, snaps all control blocks specific to the link.
<u>STATS</u>	For driver targets, displays/prints driver statistics immediately. For link targets, causes the immediate printing of statistics.
<u>TRACE</u>	For driver targets, formats and prints the driver-specific trace table. For link targets, prints the link-specific trace table.

Command Descriptions

ALTER - Dynamically Change Configuration

ALTER configuration-parameters

Dynamically changes the driver or link configuration. The `ALTER` command is followed by the driver or link configuration parameters to be altered, as appropriate. The driver configuration parameters are the same as those specified in the `DRIVER` statement (read [XCF DRIVER Statement](#)); the link configuration parameters are the same as those specified on the `LINK` statement (read [XCF LINK Statement](#)).

For example:

driver target:	XCFD # ALTER ACCEPTUI=Y,MAXRU=8A
link target:	XCFD linkname ALTER PSTAT=Y

HELP - List Available Operator Commands

HELP

Lists all the commands available to display or alter the XCF driver or an XCF link.

RESET - Reset All Statistics

RESET

Resets all statistics for the XCF driver or link target. Statistics are printed only if the `STATS` command precedes the `RESET` command.

SHOW - Display Parameter Settings



Causes the immediate printing of the current driver or link parameter settings.

SNAP - Snap Dump Selected Data Areas



Causes all driver- or link-specific control blocks and the driver- or link-specific trace table to be snapped (printed in hexadecimal). Entire Net-Work-specific control blocks are not snapped.

- For driver targets, link-specific control blocks are not snapped.
- For link targets, driver-specific control blocks are not snapped.

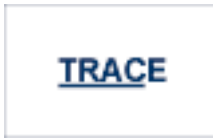
STATS - Display and Print Statistics



Causes the immediate printing of statistics and restarts the statistics interval. To print and reset statistics, specify `RESET` immediately after the `STATS` command.

For example:

driver target:	<code>XCFD # RESET</code>
link target:	<code>XCFD linkname STATS RESET</code>

TRACE - Format and Print Trace Table

Causes the driver- or link-specific trace table to be formatted and printed. The trace table is also formatted and printed automatically when the `SNAP` command is processed.

10

Entire Net-Work Diagnostic Utilities

■ NETPFIL1 Utility	132
■ NETPFIL2 Utility	134
■ Example JCL	136

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

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

NETPFIL1 Utility

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

NETPFIL1 can be used to select

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

The output from NETPFIL1 can be used as input to the [NETPFIL2 Utility](#) to perform additional filtering.

NETPFIL1 Parameters

This section describes the parameters for the NETPFIL1 utility.

DATE

```
DATE=yyyymmdd
```

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

STARTTIME

```
STARTTIME=hhmmsssth
```

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

ENDTIME

```
ENDTIME=hhmmsssth
```

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

LOG

```
LOG=log-title
```

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

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

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

TRACE

```
TRACE=trace-title
```

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

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

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

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

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

Examples

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

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

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

```
LOG=IDD
```

The following selects log records for IDDDATA only:

```
LOG=IDDDATA
```

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

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

NETPFIL2 Utility

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

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

To summarize, NETPFIL2 can be used to

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

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

Parameters

This section describes the parameters for the NETPFIL2 utility.

NONLOG

```
NONLOG={Y | N }
```

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

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

FIND

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

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

The log title must be specified exactly as it appears in the NETPRNT file; the value specified can be limited to the number of characters necessary to select the desired records. All characters following the last non blank character are considered wild cards. Spaces are valid in the log title, but the evaluation is performed only up to the last non blank character. To select all log and dump records, use the LOG= parameter.

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

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

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

The following example finds the driver block (D R I V B L K) for the TCP/IP (NETTCPI) driver.

```
FIND=D R I V B L K,OFFSET=10,VALUE=NETTCPI
FIND=D R I V B L K,OFFSET=10,VALUE=X`D5C5E3E3C3D7C9'
```

Example JCL

Running in z/OS Environments

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

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



Notes:

1. Specify the utility program you want to run.
2. This is the same load library you use to run Entire Net-Work.
3. This is the same Adabas or WAL load library you use to run Entire Net-Work.
4. This is the file that was created by Entire Net-Work with DDNAME NETPRNT.
5. This is the parameter file and may be any sequential file with LRECL=80.

Index

A

- Accepting unknown requests, 59
 - XCF DRIVER statement, 59
- ACCEPTUI parameter, 59
- ACCEPTUI= parameter
 - XCF DRIVER statement, 59
- Adabas Parallel Services
 - ADARUN parameter to enable, 31
- Adabas sysplex cluster
 - nucleus ID
 - parameter to specify, 40
- Adabus Cluster Services
 - ADARUN parameter to enable, 31
- ADACOM
 - initialization parameters, 5
- ADACOM operator commands
 - DIM
 - display active images, 68
 - DN
 - display active local nuclei, 69
 - DUMP, 69
 - SN
 - set nucleus status, 70
- ADACOM parameters
 - CMDMGR, 6
 - DBID, 7
 - FORCE, 7
 - NU
 - maximum number of users, 8
 - SVC
 - supervisory call number for interregion communication, 9
- ADAEND operator command
 - terminate an Entire Net-Work session, 112, 118
- ADARUN
 - directory of parameters, 13
- ADARUN parameters
 - CT
 - command timeout limit, 21, 50
 - determining correct settings, 12
 - example statement, 20
 - FORCE
 - overwrite ID table entry, 22
 - LU
 - length of intermediate user buffer, 23
 - NAB
 - number of attached buffers, 23
 - NC

- number of request queue elements, 24
 - setting values, 19
 - summary for Entire Net-Work, 20
- SVC
 - supervisory call number for interregion communication, 24
- TARGETID
 - Entire Net-Work node number, 25
- ALTER operator command, 57, 61
 - change line driver configuration, 126
- XCF
 - link, 127
 - XCF line driver, 127

B

- bold, 44
- braces ({}), 45
- brackets ([]), 45
- Buffers
 - length of intermediate user buffer, 23
 - number of attached buffers, 23
 - parameter for defining, 49
 - size of asynchronous pool, 50
 - size of long-term pool, 50
 - size of page-fixed pool, 50
 - size of short-term pool, 50
- BUFFERS parameter, 49

C

- Cache structure
 - statistics display, 75, 93
- cache structure
 - name, 29
 - ratio
 - directory entries to elements, 32
- choices in syntax, 45
- CLOGMRG
 - ADARUN parameter, 28
- CLOSE operator command, 57, 61
 - terminate line driver activity, 113, 119, 125
- CLUCACHENAME
 - ADARUN parameter, 29
- CLUGROUPNAME
 - ADARUN parameter, 30
- CLULOCKNAME
 - ADARUN parameter, 30
- CLUSTER
 - ADARUN parameter, 31

- cluster
 - group name, 30
 - redo updates
 - parameter to set size of pool, 33
- CMDMGR parameter, 6
- Command log
 - parameter to enable merge, 28
- communication
 - cluster nuclei
 - timeout threshold for canceled peer nucleus, 34
 - timeout threshold for command, 36
- CONNECT operator command
 - attempt to connect a named link, 113
- Control statements
 - Entire Net-Work, 44
 - order of, 44
- control statements
 - Entire Net-Work, 44
- CQTIMER parameter, 50, 55
- CT parameter, 21, 55

D

- D L command, 61
- D S command, 49
- DBID parameter, 7
- default parameter values, 45
- DEFINE LINK operator command, 51
 - define an Entire Net-Work link, 114
- DEFINE parameter, 51
- DIM command
 - display active images, 68
- DIRRATIO / ELEMENTRATIO
 - ADARUN parameter, 32
- DISABLE operator command
 - disconnect/disable link, 114
- DISCONNECT operator command
 - disconnect a link, 114
- display Adabas session parameters, 74
- DISPLAY ALINKS operator command
 - display active link information, 115
- DISPLAY LINKS operator command
 - display link information, 115
- display messaging performance statistics, 91
- DISPLAY NODES operator command
 - display node information, 115, 116
- DISPLAY operator command, 115
- DISPLAY PATHS operator command
 - display path information, 116
- DISPLAY STATS operator command
 - display usage statistics, 117
- DISPLAY TARGETS operator command
 - display target information, 118
- DISPLAY ZAPS operator command
 - display ZAP level, 118
- DMEMTB command
 - display member state table, 71
- DN command
 - display active local nuclei, 69
- DNFV command
 - display nucleus file variables, 73
- DOMAIN parameter, 51
- DPARM operator command, 74
- DPPT command

- display nucleus's own PPT block, 74
- DRIVER statement
 - defined, 44
 - defining for XCF, 57
 - overview, 57
 - parameters, 59
- DRIVER statement parameters
 - ACCEPTUI
 - whether to connect to nodes not defined in LINK statement, 59
 - GROUP
 - Entire Net-Work group name, 59
- DUMP command
 - dump storage areas, 69
- DUMP operator command
 - terminate an Entire Net-Work session with dump, 113
 - terminate an Entire Net-Work session with snap dump, 118
- DUMP parameter, 51
- DXCACHE command
 - display lock statistics, 75
- DXFILE command
 - display file statistics, 81
- DXLOCK command
 - display lock statistics, 85
- DXMSG operator command, 91
- DXSTAT command
 - display cache and lock statistics, 93
- dynamically change MXCANCEL setting, 105
- dynamically change MXCANCELWARN setting, 105
- dynamically change MXMSG setting, 105
- dynamically change MXMSGWARN setting, 106
- dynamically change MXSTATUS setting, 106
- dynamically change MXWTOR setting, 107

E

- ENABLE operator command
 - enable a previously disabled link, 118
- END operator command
 - terminate an Entire Net-Work session, 112
- Entire Net-Work
 - ADARUN parameters for, 20
 - request queue, 24
- Entire Net-Work parameters
 - using SET operator command to change settings, 121

F

- File
 - statistics display, 81
- FORCE parameter, 22
- Format
 - ADACOM operator commands, 66
 - NODE statement, 47
 - XCF DRIVER statement, 58
 - XCF line driver operator commands, 125
 - XCF LINK statement, 62
- format
 - Entire Net-Work operator commands, 110
 - Entire Net-Work statements, 44

G

- global cache area

- name, 29
- ratio
 - directory entries to elements, 32
- global lock area
 - name, 30
- GROUP parameter, 59
- GROUP= parameter
 - XCF DRIVER statement, 59

H

- HALT operator command
 - terminate an Entire Net-Work session, 112
- HELP operator command
 - XCF line driver, 127

I

- Initialization parameters
 - ADACOM, 5
- italic, 45

L

- LINK statement
 - defined, 44
 - defining for XCF, 61
 - overview, 61
 - parameters, 62
- LINK statement parameters
 - linkname, 62
 - PSTATS
 - whether statistics are printed automatically, 63
 - STATINT
 - time allowed before statistics are reported or reset, 63
 - WEIGHT
 - relative weight of this link, 64
- Lock structure
 - statistics display, 85, 93
- lock structure
 - name, 30
- LOG parameter, 52
- LOG= parameter (NODE statement), 52
- Logging checkpoint and data areas
 - LOG= parameter, 52
 - LOG= subcommand, 122
- lowercase, 45
- LRDP
 - ADARUN parameter, 33
- LU parameter, 23

M

- MAXPATH parameter, 53
- MAXPATH= parameter (NODE statement), 53
- Member state table
 - command to display, 71
- minimum keywords, 45
- MSGFORM parameter, 53
- multiprocessing
 - nucleus ID for
 - parameter to specify, 40
- MXCANCEL
 - ADARUN parameter, 34

- MXCANCEL operator command, 105
- MXCANCELWARN
 - ADARUN parameter, 35
- MXCANCELWARN operator command, 105
- MXMSG
 - ADARUN parameter, 36
- MXMSG operator command, 105
- MXMSGWARN
 - ADARUN parameter, 37
- MXMSGWARN operator command, 106
- MXSTATUS
 - ADARUN parameter, 38
- MXSTATUS operator command, 106
- MXWTOR
 - ADARUN parameter, 39
- MXWTOR operator command, 107

N

- NAB parameter, 23
- NC parameter, 24
- NETEND command
 - terminate Entire Net-Work, 110, 112
- NID0 parameter, 54
- Node
 - name, specifying the, 49
 - operation, verifying, 120
- NODE statement
 - defined, 44
 - defining for XCF, 47
 - format, 47
 - parameters, 48
- NODE statement parameters
 - BUFFERS
 - space reserved for internal buffer management, 49
 - CQTIMER
 - time allowed to retrieve command results, 50
 - DEFINE
 - accept or reject DEFINE operator command, 51
 - DOMAIN, 51
 - DUMP, 51
 - LOG
 - enable/disable logging of node statistics, 52
 - MAXPATH
 - max. path length for message in network, 53
 - MSGFORM
 - define format of console messages and DDPRINT output, 53
 - NID0, 54
 - nodename, 49
 - NTRACE
 - specify size of internal trace table, 54
 - REPLYTIM
 - time allowed for user request to complete, 55
 - START
 - enable/disable automatic start, 55
 - TIMER
 - interval between time-dependent requests, 55
 - TRACE, TROFF, TRON
 - enable/disable specific program traces, 56
 - UCMSG
 - set message for uppercase or mixed-case issue, 57
 - ULINK, 57
- normal font, 45

NTRACE parameter, 54, 56
NTRACE= parameter (NODE statement), 54
NU parameter, 8
NUCID
 ADARUN parameter, 40
nucleus
 ID for multiprocessing
 parameter to specify, 40
nucleus file variables
 display, 73

O

OPEN operator command
 reopen a stopped/closed line driver, 113, 119, 125
Operator commands, 1
 ADACOM, 66
 ADAEND
 terminate Entire Net-Work session, 112, 118
 ALTER, 57, 61
 change line driver configuration, 126
 CLOSE, 57, 61
 terminate all line driver activity, 113, 119, 125
 CONNECT
 attempt to connect a named link, 113
 D L
 display link, 61
 D S
 display statistics, 49
 DEFINE LINK, 51
 define an Entire Net-Work link, 114
 DISABLE
 disconnect/disable link, 114
 DISCONNECT
 disconnect a link, 114
 DISPLAY ALINKS
 display active link information, 115
 DISPLAY LINKS
 display link information, 115
 DISPLAY NODES
 display node information, 115, 116
 DISPLAY PATHS
 display path information, 116
 DISPLAY STATS
 display usage statistics, 117
 DISPLAY TARGETS
 display target information, 118
 DISPLAY ZAPS
 display ZAP level, 118
 DMEMTB
 display member state table, 71
 DUMP
 terminate Entire Net-Work session with dump, 113
 terminate Entire Net-Work session with snap dump, 118
 DXCACHE
 display lock statistics, 75
 DXFILE
 display file statistics, 81
 DXLOCK
 display lock statistics, 85
 DXSTAT
 display cache and lock statistics, 93
 ENABLE
 enable a previously disabled link, 118

END
 terminate Entire Net-Work session, 112
Entire Net-Work
 summary list for z/OS, 111
HALT
 terminate Entire Net-Work session, 112
NETEND
 terminate Entire Net-Work session, 112
OPEN
 reopen a stopped/closed line driver, 113, 119, 125
PROBE
 verify node availability, 120
RESET
 resets all statistics, 127, 128
RESUME
 allows link to resume sending queued messages, 121
SET CQTIMER
 time allowed to retrieve command results, 121
SET DUMP
 storage areas to include, 122
SET LOG
 determine extent of logging activity, 122
SET MAXPATH
 maximum number of links in message path, 123
SET MSGFORM
 message format for display and print, 123
SET REPLYTIM
 time allowed for reply to user request, 123
SET TRACE, SET TROFF, SET TRON
 determine extent of tracing activity, 123
SHOW
 display parameter settings, 128
SNAP
 snap dump selected data areas, 118, 124, 128, 129
START
 restart a stopped/closed line driver, 113, 119, 125
 start individual line driver, 55
STATS
 displays/prints statistics, 127, 128
STOP
 terminate Entire Net-Work session, 112
SUSPEND
 stops link from sending queued messages, 121, 125
TERMINATE
 terminate Entire Net-Work session, 112
to terminate an Entire Net-Work session, 112
TRACE
 formats/prints trace table, 129
XCF line driver, 125
operator commands, 110
 cluster nucleus, 71
 DNFV
 display nucleus file variables, 73
 DPARM, 74
 DPPT
 display nucleus's own PPT block, 74
 DXMSG, 91
 Entire Net-Work, 110
 MXCANCEL, 105
 MXCANCELWARN, 105
 MXMSG, 105
 MXMSGWARN, 106
 MXSTATUS, 106
 MXWTOR, 107

optional syntax elements, 45

P

parameter

- syntax conventions, 44
- syntax rules, 46

Path length

- defining the maximum, 53
- SET MAXPATH command, 123

Path length, defining the maximum

- MAXPATH= parameter (NODE statement), 53

PPT

- display nucleus's own block, 74

PROBE operator command

- verify node availability, 120

PROGRAM

- ADARUN parameter, 41

PSTATS parameter, 63

punctuation and symbols in syntax, 46

R

redo pool

- parameter to set size of, 33

REPLYTIM parameter, 55

required syntax elements, 45

RESET operator command

- resets all statistics, 127, 128

RESUME operator command

- allows link to resume sending queued messages, 121

S

SET CQTIMER= command, 50

- time allowed to retrieve command results, 121

SET DUMP= command

- storage areas to include, 122

SET LOG= command, 52

- determine extent of logging activity, 122

SET MAXPATH= command, 53

- maximum number of links in message path, 123

SET MSGFORM= command

- format for message display and print, 123

SET operator command

- change Entire Net-Work parameter settings, 121
- change NODE statement parameter settings, 121

SET REPLYTIM= command, 55

- time allowed for reply to user request, 123

SET TRACE=, TROFF=, TRON= command

- determine extent of tracing activity, 56, 123

SET UCMSG command, 57

SET ULINK command, 57

SHOW operator command

- display parameter settings, 128

SN command

- set nucleus status, 70

SNAP operator command

- snap dump selected data areas, 118, 124, 128, 129

START operator command, 55

- restart a stopped/closed line driver, 113, 119, 125

START parameter, 55

statement

- syntax conventions, 44

syntax rules, 46

STATINT parameter, 63

STATS operator command

- displays/prints statistics, 127, 128

STOP operator command

- terminate an Entire Net-Work session, 112

SUSPEND operator command

- stops link from sending queued messages, 121, 125

SVC parameter, 9, 24

Syntax

- DRIVER statement, 57

- LINK statement, 61

syntax

- conventions, 44

- rules, 46

syntax conventions

- bold, 44
- braces ({}), 45
- brackets ([]), 45
- defaults, 45
- italic, 45
- lowercase, 45
- minimum keywords, 45
- mutually exclusive choices, 45
- normal font, 45
- optional elements, 45
- punctuation and symbols, 46
- required elements, 45
- underlining, 45
- uppercase, 44
- vertical bars (|), 45

T

Target IDs

- allowing duplicates, 26
- overwriting, 22

TARGETID parameter, 25

TERMINATE operator command

- terminate an Entire Net-Work session, 112

Time

- waiting time for results (CQTIMER parameter), 49

TIMER parameter, 55

Timing parameters

- interaction between, 56

Trace control parameters, 56

TRACE operator command

- formats/prints trace table, 129

TRACE parameter, 56

Trace table size, 54

Trace table size, specifying the (NTRACE= parameter), 54

TRACESIZ= parameter

- XCF

- DRIVER statement, 61

Tracing line driver activity, 56

- XCF driver, 61

TROFF parameter, 56

TRON parameter, 56

U

UCMSG parameter, 57

ULINK parameter, 57

underlining, 45

- uppercase, 44
- Users
 - maximum number of, 8
- utility control statement
 - parameter values
 - default, 45
 - syntax conventions, 44
 - syntax rules, 46

V

- Verify remote node operation, 120
- vertical bars (|), 45

W

- WEIGHT parameter, 64

X

- XCF
 - DRIVER statement, 57
 - LINK statement, 61
 - NODE statement, 47
- XCF DRIVER statement
 - format, 58
 - parameters, 59
- XCF group
 - name, 30
- XCF line driver
 - operator commands
 - descriptions, 127
 - overview, 126
- XCF LINK statement
 - format, 62
 - parameters, 62