

Adabas Cluster Services

Operator Commands

Version 7.4.2

September 2009

This document applies to Adabas Cluster Services Version 7.4.2 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 2009. 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 Operator Commands	1
2 Adabas Cluster Operator Commands	3
ADACOM Operator Commands	4
Adabas Cluster Nucleus Operator Commands	8
3 Entire Net-Work Operator Commands	39
Overview of Commands for OS/390 Environments	41
Command Descriptions	42
XCF Line Driver Commands	55
Index	59

1 Operator Commands

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.

The Adabas Cluster Services Operator Commands documentation is organized in the following parts:

	Adabas Cluster Operator Commands
	Entire Net-Work Operator Commands

2 Adabas Cluster Operator Commands

- ADACOM Operator Commands 4
- Adabas Cluster Nucleus Operator Commands 8

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.

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 dataset with the DD name COMPRINT. It writes messages pertinent to an individual SVC/DBID combination into the output dataset with the DD name 'P*ss**dddd*', 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

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.

Option	Action
nucleus-id {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.
nucleus-id {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

In addition to the console operator commands documented in the *Adabas Operations* documentation , which can be issued against any local nucleus, you can issue the following commands against a local Adabas cluster nucleus:

Command	To display...
DMEMTB	information about active Adabas cluster nuclei
DNFV	information about current file use
DPPT	information about the nucleus's own PPT block
DXCACHE	the primary cache-related statistics
DXFILE	the cache-related statistics for 1 to 5 files
DXLOCK	the lock-related statistics
DXSTAT	all cache- and lock-related statistics



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

DMEMTB - Display Member State Table

Use DMEMTB to display information about active nuclei in an Adabas cluster.

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

Sample Output

```
ADAI29 OPER CMD: DMEMTB
ADAX61 00006 2001-01-18 00:13:09 Member Status Table
ADAX61 00006 2001-01-18 00:13:09 Other members:      1
ADAX61 00006 2001-01-18 00:13:09   This system:      1
ADAX61 00006 2001-01-18 00:13:09
ADAX61 00006 2001-01-18 00:13:09           NUCID:      132
ADAX61 00006 2001-01-18 00:13:09           Flags 1: 11
ADAX61 00006 2001-01-18 00:13:09           Flags 2: 00
ADAX61 00006 2001-01-18 00:13:09           System: IMAGE1
ADAX61 00006 2001-01-18 00:13:09
ADAX61 00006 2001-01-18 00:13:09           NUCID:       3
ADAX61 00006 2001-01-18 00:13:09           Flags 1: 97
ADAX61 00006 2001-01-18 00:13:09           Flags 2: 00
ADAX61 00006 2001-01-18 00:13:09           System: IMAGE1
ADAN41 00006 2001-01-18 00:13:09 Function completed
```

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=nnnnn	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=nnnnn	is the number of users on this nucleus who are currently accessing this file.
CU=nnnnn	is the number of users on this nucleus who are currently updating this file.

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
```

Adabas Cluster Operator Commands

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 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 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 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 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 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 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	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 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 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	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	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		

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 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
```

Adabas Cluster Operator Commands

```

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

```

Adabas Cluster Operator Commands

```

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
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
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
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 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

```

Adbas Cluster Operator Commands

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	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	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			
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 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			
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	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

Adabas Cluster Operator Commands

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		

3 Entire Net-Work Operator Commands

- Overview of Commands for OS/390 Environments 41
- Command Descriptions 42
- XCF Line Driver Commands 55

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:

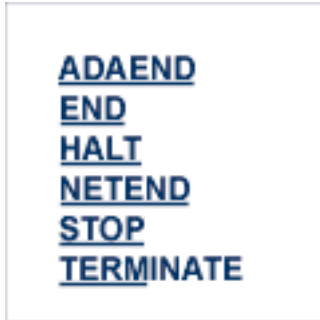
Overview of Commands for OS/390 Environments

The following table summarizes the Entire Net-Work operator commands for OS/390:

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

Command Descriptions

Commands to Terminate an Entire Net-Work Session



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



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* in the *Parameter and Parameter Statements* documentation 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 dataset 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 section *XCF LINK Statement* in the *Parameter and Parameter Statements* documentation. 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



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



ENABLE

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 drivername
```

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`). See section *XCF DRIVER Statement* in the *Parameter and Parameter Statements* documentation for more information.

Note that this command is currently a synonym for the `START` command. See the explanation in section *XCF DRIVER Statement* in the *Parameter and Parameter Statements* documentation 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 in the *Parameters and Parameter Statements* documentation.

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 dataset. For more information, refer to the *LOG* parameter description in the *Parameters and Parameter Statements* documentation.

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 in the *Parameters and Parameter Statements* documentation.

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 in the *Parameters and Parameter Statements* documentation.

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 in the *Parameters and Parameter Statements* documentation.

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 in the *Parameters and Parameter Statements* documentation.

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 in the *Parameters and Parameter Statements* documentation.

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 in the *Parameters and Parameter Statements* documentation.

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`). See section *XCF DRIVER Statement* in the *Parameter and Parameter Statements* documentation 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. See section *XCF DRIVER Statement* in the *Parameter and Parameter Statements* documentation 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

A rectangular box with a thin border containing the text `ALTER configuration-parameters`. The word `ALTER` is in a bold, blue font, and `configuration-parameters` is in a regular, blue font.

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 (see the section *XCF DRIVER Statement* in the *Parameter and Parameter Statements* documentation); the link configuration parameters are the same as those specified on the `LINK` statement (see the section *XCF LINK Statement* in the *Parameter and Parameter Statements* documentation).

For example:

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

HELP - List Available Operator Commands



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

RESET - Reset All Statistics



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:	XCFD # RESET
link target:	XCFD linkname STATS RESET

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.

Index

A

- ADAEND operator command
 - terminate an Entire Net-Work session, 42, 48
- ALTER operator command
 - change line driver configuration, 55
 - XCF
 - link, 56
 - XCF line driver, 56

C

- CLOSE operator command
 - terminate line driver activity, 42, 49, 54
- CONNECT operator command
 - attempt to connect a named link, 42, 43

D

- DEFINE LINK operator command
 - define an Entire Net-Work link, 43
- DISABLE operator command
 - disconnect/disable link, 44
- DISCONNECT operator command
 - disconnect a link, 44
- DISPLAY ALINKS operator command
 - display active link information, 45
- DISPLAY LINKS operator command
 - display link information, 44, 45
- DISPLAY NODES operator command
 - display node information, 44, 45
- DISPLAY operator command, 44
- DISPLAY PATHS operator command
 - display path information, 46
- DISPLAY STATS operator command
 - display usage statistics, 46
- DISPLAY TARGETS operator command
 - display target information, 47
- DISPLAY ZAPS operator command
 - display ZAP level, 47
- DUMP operator command
 - terminate an Entire Net-Work session with dump, 42
 - terminate an Entire Net-Work session with snap dump, 48

E

- ENABLE operator command
 - enable a previously disabled link, 48

- END operator command
 - terminate an Entire Net-Work session, 42
- Entire Net-Work parameters
 - using SET operator command to change settings, 50

F

- Format
 - ADACOM operator commands, 4
 - Entire Net-Work operator commands, 40
 - XCF line driver operator commands, 55

H

- HALT operator command
 - terminate an Entire Net-Work session, 42
- HELP operator command
 - XCF line driver, 57

L

- Logging checkpoint and data areas
 - LOG= subcommand, 52

N

- NETEND command
 - terminate Entire Net-Work, 40, 42
- Node
 - operation, verifying, 49

O

- OPEN operator command
 - reopen a stopped/closed line driver, 42, 49, 54
- Operator commands, 1, 40
 - ADACOM, 4
 - ADAEND
 - terminate Entire Net-Work session, 42, 48
 - ALTER
 - change line driver configuration, 55
 - CLOSE
 - terminate all line driver activity, 42, 49, 54
 - CONNECT
 - attempt to connect a named link, 42, 43
 - DEFINE LINK
 - define an Entire Net-Work link, 43
 - DISABLE
 - disconnect/disable link, 44

DISCONNECT
 disconnect a link, 44

DISPLAY ALINKS
 display active link information, 45

DISPLAY LINKS
 display link information, 44, 45

DISPLAY NODES
 display node information, 44, 45

DISPLAY PATHS
 display path information, 46

DISPLAY STATS
 display usage statistics, 46

DISPLAY TARGETS
 display target information, 47

DISPLAY ZAPS
 display ZAP level, 47

DUMP
 terminate Entire Net-Work session with dump, 42
 terminate Entire Net-Work session with snap dump, 48

ENABLE
 enable a previously disabled link, 48

END
 terminate Entire Net-Work session, 42

Entire Net-Work, 40

HALT
 terminate Entire Net-Work session, 42

NETEND
 terminate Entire Net-Work session, 42

OPEN
 reopen a stopped/closed line driver, 42, 49, 54

PROBE
 verify node availability, 49

RESET
 resets all statistics, 57, 58

RESUME
 allows link to resume sending queued messages, 50

SET CQTIMER
 time allowed to retrieve command results, 51

SET DUMP
 storage areas to include, 51

SET LOG
 determine extent of logging activity, 52

SET MAXPATH
 maximum number of links in message path, 52

SET MSGFORM
 message format for display and print, 52

SET REPLYTIM
 time allowed for reply to user request, 52

SET TRACE, SET TROFF, SET TRON
 determine extent of tracing activity, 53

SHOW
 display parameter settings, 57

SNAP
 snap dump selected data areas, 48, 53, 57, 58

START
 restart a stopped/closed line driver, 42, 49, 54

STATS
 displays/prints statistics, 57, 58

STOP
 terminate Entire Net-Work session, 42

SUSPEND
 stops link from sending queued messages, 50, 54

TERMINATE
 terminate Entire Net-Work session, 42

TRACE
 formats/prints trace table, 58

XCF line driver, 55

P

Path length
 defining the maximum
 SET MAXPATH command, 52

PROBE operator command
 verify node availability, 49

R

RESET operator command
 resets all statistics, 57, 58

RESUME operator command
 allows link to resume sending queued messages, 50

S

SET CQTIMER= command
 time allowed to retrieve command results, 51

SET DUMP= command
 storage areas to include, 51

SET LOG= command
 determine extent of logging activity, 52

SET MAXPATH= command
 maximum number of links in message path, 52

SET MSGFORM= command
 format for message display and print, 52

SET operator command
 change Entire Net-Work parameter settings, 50
 change NODE statement parameter settings, 50

SET REPLYTIM= command
 time allowed for reply to user request, 52

SET TRACE=, TROFF=, TRON= command
 determine extent of tracing activity, 53

SHOW operator command
 display parameter settings, 57

SNAP operator command
 snap dump selected data areas, 48, 53, 57, 58

START operator command
 restart a stopped/closed line driver, 42, 49, 54

STATS operator command
 displays/prints statistics, 57, 58

STOP operator command
 terminate an Entire Net-Work session, 42

SUSPEND operator command
 stops link from sending queued messages, 50, 54

T

TERMINATE operator command
 terminate an Entire Net-Work session, 42

TRACE operator command
 formats/prints trace table, 58

V

Verify remote node operation, 49