software AG

Adabas Parallel Services

Reference

Version 8.1.3

June 2014

Adabas Parallel Services

Table of Contents

Preface

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

This document is organized as follows:

Initialization Parameters   Describes the initialization parameters needed for an Adabas Parallel Services environment.

Cluster Operator Commands   Describes operator commands for ADACOM and for an Adabas Parallel Services cluster nucleus.

1 Initialization Parameters

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

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

- To run an Adabas nucleus, the name must be ADANUC.

- To set up an ADACOM task, the name must be ADACOM (no other ADARUN parameters are recognized for the ADACOM program run).

The chapter is organized in the following parts:

ADACOM Initialization Parameters	Describes the ADACOM initialization parameters required by Adabas Parallel Services.
About ADARUN Parameters for Cluster Nuclei	Describes the Adabas nucleus (ADANUC) initialization parameters required by Adabas Parallel Services.
Useful ADARUN Parameters	Provides a reference for all of the ADARUN parameters useful to Adabas Parallel Services processing.

2 ADACOM Initialization Parameters

An unlimited number of Adabas Parallel Services nucleus clusters, each with up to 31 members sharing a common database, can be defined for an operating system image.

The ADACOM initialization task (`ADARUN PROGRAM=ADACOM`) must be run in order to set up the environment, and it must be maintained in order to monitor and control the nuclei of one or more Adabas Parallel Services clusters.

ADACOM initialization parameters specify the Router ID / DBID combinations (sets) that the ADACOM is to manage.

- The Router ID identifies the SVC number on z/OS or z/VSE, or the IDTNAME on BS2000/OSD. The Router ID value must be the same within a cluster; for different clusters, the same or different Router IDs may be used.

- The DBID identifies the external physical database shared by a particular cluster of nuclei and known to the application.

Other ADACOM parameters are discussed in the following sections.

Although a single ADACOM job can run all Router ID / DBID sets in an Adabas Parallel Services environment, it is possible to run multiple ADACOM tasks simultaneously with the same, mixed, or completely different Router ID / DBID sets. An ADACOM subtask is attached to each Router ID / DBID set for each ADACOM task in which it occurs.

ADACOM prints global messages that apply to all Router ID / DBID sets to two output data sets or files. One is identified by the DD or link name P*ssddddd*, where *ss* is the last two digits of the SVC number and *ddddd* is the DBID. (On BS2000 systems, the *ss* characters are the fourth and last nonblank characters of the IDTNAME parameter. For example, if IDTNAME=ADAIDT1 and DBID=123, then the related link names are PI100123 and DI100123.) The P*ssddddd* data set or file is also used by Adabas Cluster Services. The second output data set or file is identified by the DD or link name D*ssddddd* and is used only by Adabas Parallel Services. ADACOM uses the D*ssddddd* data set or file to record which cache and lock spaces it allocates and releases for the associated cluster database, as the Adabas Parallel Services nuclei start and end.

On z/OS systems, ADACOM automatically allocates these data sets in the spool with SYSOUT=*, if they are not explicitly specified.

For a sample job to run ADACOM, read the z/OS section *Create a Startup Procedure for ADACOM* in the *Adabas Parallel Services Installation Guide*.

DBID - Database Identification

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

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

This number must be unique among all Adabas database IDs, NUCIDs, Natural buffer pool IDs, etc.

FORCE - Force Restart after Abnormal Termination

Parameter	Specify...	Possible Values	Default
FORCE	whether ADACOM forces a restart after an abnormal termination.	YES ǀ NO	NO

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

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

> **Notes:**

1. FORCE=YES works only when the NU parameter value is being changed and has the effect of resetting the environment.

2. The nucleus ADARUN parameter FORCE is neither seen nor used by the SVCCLU. The ADACOM parameter FORCE may only be used to change the NU that is valid for the cluster.

3. If you use the ADACOM FORCE parameter, remember to remove it from ADACOM after you have reset the environment.

IDTNAME - Define ID Table Name (BS2000 Only)

Parameter	Specify...	Possible Values	Default
`IDTNAME`	the IDT common memory pool name to be used for the Adabas Parallel Services cluster session.	ADAiiiii	ADABAS5B

The `IDTNAME` defines the name of the common memory pool used to find communication details for this nucleus. This common memory pool will be defined as Global.

Value is "ADA*iiiii*", maximum of eight characters.

For example, `IDTNAME=ADAPROD1`

NU - Number of User Table Elements on the OS Image

Parameter	Specify...	Minimum	Maximum	Default
`NU`	the number of users that can be active in parallel on cluster nuclei in the image.	20	16 777 215	200

In the extended CSA (ECSA), the `SVCCLU` maintains a user table with entries (UTEs) containing information about every active user in the cluster nuclei on the operating system image. A UTE is assigned when a user issues an OP command or (if the user did not issue an OP command) at the first Adabas command. The UTE is released when the user issues a CL command or when a timeout occurs.

The ADACOM `NU` parameter specifies the number of concurrent users (UTEs) allowed for all the nuclei of a cluster. The first ADACOM started governs the value for `NU`: different values set for subsequent nuclei or ADACOMs are ignored.

To manually change the `NU` value, you must stop all cluster nuclei and ADACOM tasks in the image, modify the `NU` value for all the nucleus and ADACOM jobs, and then restart.

When the `NU` parameter is set to zero, any space allocated for the user table is freed, the Adabas Parallel Services control block is freed, and the ADACOM task terminates. However, if ADACOM believes that there are active nuclei, ADACOMs, or users, and `FORCE=YES` is not used, no action is taken.

For example, the following parameter would run the Adabas Parallel Services cluster nuclei with 500 elements in the user table:

```
NU=500
```

SVC - Interregion Communication Number

Parameter	Specify...	Possible Values	Default
SVC	the supervisor call number to be used for the Adabas Parallel Services nucleus cluster session.	z/OS: 200-255 z/VSE: see text	z/OS: 249 z/VSE: 45

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

The SVC number is specified as an integer and must correspond to the number used for the Adabas SVC (ADASVC).

- For z/OS, valid SVC values are 200-255.

- For z/VSE, 45 is the recommended value but any free SVC value can be used. Read the *Adabas Installation Guide* for information about finding free z/VSE Adabas SVC values.

For example, the following parameter will execute an Adabas Parallel Services nucleus cluster session on a z/OS image using ADASVC 202.

```
SVC=202
```

3   About ADARUN Parameters for Cluster Nuclei

Software AG recommends that you use the default settings (or your existing values) of the Adabas ADARUN parameters for each Adabas nucleus in an Adabas Parallel Services cluster, and then tune the values after analyzing the performance of the node or cluster. Read *Performance and Tuning* in the *Adabas Parallel Services Operations* for information about expected differences.

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

For your convenience, ADARUN parameters that are most significant to Adabas Parallel Services usage are described in *Useful ADARUN Parameters*, elsewhere in this guide.

This chapter covers the following topics:

Global ADARUN Parameters

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

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

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

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

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

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

ADARUN Parameter Usage in Cluster Environments

A cluster nucleus makes use of:

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

- *local* parameters, which can be different for each nucleus. Some of these parameters are modifiable (locally modifiable) using an operator command or the Adabas Online System; others are fixed (locally fixed) and cannot be modified.

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

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

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

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

Parameter	Usage	No	LF	LM	GF	GM
CRETRY	Cache space retry interval (Adabas Caching Facility)			X		
CSTORAGE	RABN caching activation (Adabas Caching Facility)			X		
CT	Command time limit (seconds)					X
CWORKSTORAGE	Cache space type (Adabas Caching Facility)			X		
CWORK2FAC	Percentage of Work part 2 cached (Adabas Caching Facility)			X		
CWOFK3FAC	Percentage of Work part 3 cached (Adabas Caching Facility)			X		
CXFILE	Excluded files (Adabas Caching Facility)			X		
DATACACHE	Controller caching control for the Data Storage component		X			
DIRRATIO/ELEMENTRATIO	Ratio of directory entries to data elements in a cluster cache structure/area				X	
DBID	Database ID (physical)				X	
DEVICE	Device type of the first ASSO extent				X	
DSF	Delta Save Facility control				X	
DSFEX1	Delta Save Facility user exit				X	
DTP *	Distributed transaction processing control	*			X	
DUALCLD	Dual command log device		X			
DUALCLS	Dual command log size (blocks)		X			
DUALPLD	Dual protection log device		X			
DUALPLS	Dual protection log size (blocks)		X			
FASTPATH	Adabas Fastpath control				X	
FMXIO	Limit parallel I/O operations by LFIOP flush processing			X		
FORCE	Overwrite IDTE		X			
HEXnn	Hyperdescriptor exit(s)				X	
IGNDIB	Ignore DIB entry		X			
IGNDTP	Ignore distributed transaction processing area (Work part 4)		X			
INTNAS	Interval between nucleus statistic checkpoints (SYNS 60)				X	
LBP	Length of buffer pool		X			
LCP	Length of security pool		X			
LDEUQP	Length of unique (UQ) descriptor pool		X			
LDTP	Length of distributed transaction processing area (Work part 4)		X			
LFIOP	Length of asynchronous flush pool		X			
LFP	Length of internal format buffer pool		X			

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

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

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

Notes:

1. * Adabas Parallel Services 8.1 does not support DTP=TM, but it does support DTP=RM.

2. ** The `PREFxxx` parameters are used with application programs (`PROGRAM=USER`) making Adabas calls. They have no effect when specified for an Adabas nucleus.

Specifying ADARUN Parameters for Cluster Nuclei

When specifying ADARUN session parameters for Adabas Parallel Services cluster nuclei:

■ ensure that the correct program to be executed is specified (PROG=ADANUC); and

■ determine which setting is applicable for the `SVC` parameter for the session.

The `CLOGMRG`, `CLUSTER`, `CLUCACHESIZE`, `CLUCACHETYPE`, `CLUCACHEUNCHANGED`, `CLULOCKSIZE`, `DIRRATIO` / `ELEMENTRATIO`, `LRDP`, and `NUCID` parameters are used by the Adabas Parallel Services cluster nucleus and its environment.

If protection logs or command logs are used in a cluster environment, they must be dual or multiple logs and all nuclei must use them. All cluster nuclei must have the same PLOGRQ setting.

The remaining Adabas cluster nucleus parameters are the same as those of a standard Adabas nucleus. For more information, read the *Adabas Operations Manual*.

4   **Useful ADARUN Parameters**

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

CLOGMRG : Cluster Command Log Merge

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

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

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

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

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

Parameter Dependencies

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

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

Dynamic Modification

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

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

CLUCACHESIZE : Size of the Global Cache Area

Parameter	Specify . . .	Minimum	Maximum	Default
CLUCACHESIZE	the amount of storage to allocate for the global cache area that services the Adabas Parallel Services cluster in which the current nucleus participates.	128K	depends on the type of cache (CLUCACHETYPE parameter)	none

The CLUCACHESIZE parameter specifies the amount of storage to be allocated for the global cache area which all nuclei in the Adabas Parallel Services cluster use to share Associator and Data blocks from the database. It has an effect only if CLUSTER=LOCAL is also specified.

The location and maximum size of the global cache area depends on the CLUCACHETYPE parameter. If CLUCACHETYPE=DSP, the global cache is kept in a dataspace shared between the Adabas Parallel Services nuclei. Otherwise, the cache is kept in shared 64-bit virtual storage (z/OS systems only).

CLUCACHESIZE must be specified for Adabas Parallel Services nuclei; there is no default.

Each Adabas Parallel Services nucleus has its own buffer pool sized by the ADARUN parameter LBP . Optionally, it may also have one or more local cache spaces administered by the Adabas Caching Facility (refer to the description of the ADARUN parameter CACHE). In addition, Adabas Parallel Services uses a global cache area to ensure that current images of Associator and Data blocks from the database are available to all Adabas Parallel Services cluster nuclei.

The global cache area must be large enough to retain:

■ directory elements for all blocks that reside in all buffer pools and the global cache itself; and

■ enough data elements to keep changed blocks between buffer flushes (cast-outs).

The ADARUN parameters DIRRATIO **and** ELEMENTRATIO determine the ratio between the number of directory and data elements.

CLUCACHESIZE may be specified in bytes, in kilobytes followed by a "K", in megabytes followed by an "M", or in gigabytes followed by a "G":

- the minimum size is 131072 bytes (or 128K);

- for CLUCACHETYPE=DSP, the maximum size is 2147483147 bytes (or 2097152K; or 2048M; or 2G); and

- the maximum allowed numeral is 2147483147; for CLUCACHETYPE=V64, use the K, M or G suffix (for example, 10G or 10240M or 10485760K for a global cache of 10 gigabytes size).

The value is rounded up to the nearest 4-kilobyte boundary.

Appropriate sizes for the global cache area depend very much on the application load in the cluster. For more details about these calculations, read *Global Cache Area Size* in *Adabas Parallel Services Operations*. You can use the Adabas Online System cache structure size calculator described in *Estimating Sizes for the Cache Structure in a Cluster Environment* in *Adabas Parallel Services Operations*. to determine appropriate global cache area sizes.

Example

In the following example, the global cache area allocated is 500 M.

```
ADARUN PROG=ADANUC,CLUCACHESIZE=500M
```

CLUCACHETYPE: Global Cache Area Storage Type Option

Parameter	Specify . . .	Valid Values	Default
CLUCACHETYPE	the virtual storage type for the global cache area	DSP | V64 | L64	DSP

The CLUCACHETYPE parameter specifies the type of virtual storage to be used for the global cache of an Adabas Parallel Services cluster. It has an effect only if CLUSTER=LOCAL is also specified. Valid values are "DSP" (available on all supported operating systems) and "V64" or "L64" (available only on z/OS systems).

The size of the global cache is specified in the CLUCACHESIZE **parameter**.

The default value is "DSP", indicating that a shared dataspace will be used for the global cache. In this case, the maximum cache size is 2 GB.

If "V64" is specified (on z/OS systems), a shared storage area in 64-bit virtual storage will be used for the global cache. In this case, the cache can be made much larger than 2 GB, but is still subject to the MEMLIMIT value in effect for the ADACOM job for the Adabas Parallel Services database.

If "L64" is specified (on z/OS 1.9 or later systems running on IBM z10 machines), the global cache space shared by the cluster nuclei will reside in 64-bit virtual storage that is backed by large pages (provided the large page pool has been configured to a sufficient size and is available in the system).

Notes:

1. If one ADACOM job is used to hold the shared cache and lock spaces for multiple Adabas Parallel Services clusters, L64 cache spaces of different clusters are slightly less isolated from one another than DSP or V64 cache spaces, creating a marginally higher risk that a cache-related error in one cluster might induce an error in a different cluster. This very small risk can be avoided altogether by using different ADACOM jobs for different L64 cache spaces, but this is not mandatory.

2. Virtual 64-bit storage backed by large pages can only be used on IBM z10 mainframes running z/OS Version 1.9 or above and for which IBM large page support has been enabled via IBM APAR OA20902. You must also allocate the size of the large page pool (use the LFAREA parameter in the IEASYS*xx* member of SYS1.PARMLIB). The LFAREA parameter allows you to specify the amount of real storage to be used for large pages; this parameter cannot be changed dynamically and, if it is not set, page frames will remain allocated at a size of 4K.

Example

In the following example, the type of virtual storage used for the global cache area is specified as "V64".

```
ADARUN PROG=ADANUC,CLUCACHETYPE=V64
```

CLUCACHEUNCHANGED : Global Cache Unchanged Block Option

Parameter	Specify . . .	Possible Values	Default
CLUCACHEUNCHANGED	whether unchanged blocks are written to global cache during Adabas Parallel Services processing.	YES | NO	NO

The `CLUCACHEUNCHANGED` parameter specifies whether an Adabas Parallel Services nucleus should write unchanged blocks to the global cache. It has an effect only if CLUSTER=LOCAL is also specified. Valid values are "YES" and "NO".

■ The default value is "NO", which indicates that the nucleus will write only changed (updated) Associator and Data blocks from the database to the global cache.

■ If "YES" is specified, the nucleus will write changed and unchanged Associator and Data blocks to the global cache.

Recommendation

If a lot of virtual storage can be made available for local and global caching by the Adabas Parallel Services cluster (where local caching is controlled by the ADARUN `LBP` parameter and possibly the Adabas Caching Facility add-on product), assigning the bulk of the available storage to the global cache (ADARUN `CLUCACHESIZE` parameter), which is shared by all Adabas Parallel Services nuclei for the database, provides for more economic use of the storage than assigning it to the local caches of the individual cluster nuclei (ADARUN `LBP` parameter and the Adabas Caching

Facility ADARUN `CASSOMAXS` and `CDATAMAXS` parameters), which are private to each nucleus. However, each cluster nucleus requires sufficient local buffer pool space (`LBP` parameter) to hold the entire working set of Associator and Data blocks needed for parallel command processing.

In cases where the global cache size (`CLUCACHESIZE` parameter) is set to a large value, `CLUCACHEUNCHANGED` should be set to "YES", so that the global cache achieves a better hit rate (efficiency).

Example

In the following example, the `CLUCACHEUNCHANGED` parameter is set to that both changed and unchanged Associator and Data blocks are written to the global cache.

```
ADARUN PROG=ADANUC, CLUCACHEUNCHANGED=YES
```

CLULOCKSIZE : Size of the Global Lock Area

Parameter	Specify . . .	Minimum	Maximum	Default
`CLULOCKSIZE`	the amount of storage allocated for the global lock area	128K	2G	none

The `CLULOCKSIZE` parameter specifies the amount of storage to be allocated for the global lock area which all nuclei in the Adabas Parallel Services cluster use to synchronize their processing. It has an effect only if CLUSTER=LOCAL is also specified.

`CLULOCKSIZE` must be specified for Adabas Parallel Services nuclei; there is no default.

`CLULOCKSIZE` may be specified in bytes, in kilobytes followed by a "K", in megabytes followed by an "M", or in gigabytes followed by a "G".

- The minimum size is 131072 bytes (or 128K)

- The maximum size is 2147483147 bytes (or 2097152K, 2048M, or 2G).

The value is rounded up to the nearest 4-kilobyte boundary.

Appropriate sizes for the global lock area depend on the expected amount of parallel command processing in the cluster. For more details about these calculations, read *Global Cache Area Size* in *Adabas Parallel Services Operations*. You can use the Adabas Online System lock structure size calculator described in *Estimating Sizes for the Lock Structure in a Cluster Environment* in *Adabas Parallel Services Operations*. to determine appropriate global lock area sizes.

CLUSTER : Adabas Nucleus Cluster Session Control

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

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

■ If CLUSTER=NO (the default), the nucleus will not participate in an Adabas cluster. If CLUSTER=NO, the NUCID parameter must be set (or default) to 0.

■ If CLUSTER=LOCAL, the nucleus will participate in a cluster where all nuclei run on the same operating system image. The add-on product Adabas Parallel Services is required to run in this mode.

■ If CLUSTER=SYSPLEX, the nucleus will participate in a cluster where the nuclei may run on different systems within a parallel sysplex. This is supported only for z/OS systems. The add-on product Adabas Cluster Services is required to run in this mode.

If CLUSTER=LOCAL or SYSPLEX, the NUCID parameter must be set to a nonzero value that identifies the individual cluster nucleus. Also, the parameter settings MODE=SINGLE, READONLY=YES, LFIOP=0 and DTP=TM, and the use of sequential protection log data sets (DD-name/link name DDSIBA) are not allowed for cluster nuclei.

Example

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

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

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

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

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

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

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

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

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

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

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

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

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

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

LRDP : Length of the Redo Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LRDP	the size (in bytes) of the redo pool.	0; or 80000	2147483647	For Adabas Parallel Services 8.2 or later, the default is 0 (zero). For Adabas Cluster Services 8.2 or later, the default is either the value of the ADARUN LFIOP parameter or 10 megabytes, whichever is smaller.

Parameter	Specify . . .	Minimum	Maximum	Default
				For earlier releases of Adabas Parallel Services and Adabas Cluster Services, the default is the value of the ADARUN `LFIOP` parameter.

The LRDP parameter specifies the size of the redo pool in an Adabas Cluster Services or Adabas Parallel Services nucleus. It has an effect only when CLUSTER=SYSPLEX or LOCAL has also been specified. For Adabas Parallel Services 8.2 or later, its default is zero (0). For Adabas Cluster Services 8.2 or later, its default is either 10 megabytes (10M) or the value of the ADARUN LFIOP parameter, whichever is smaller. For earlier releases of Adabas Parallel Services and Adabas Cluster Services, the default is the value of the ADARUN `LFIOP` parameter.

- If LRDP is set to zero, the cluster nucleus performs immediate publishing: Updated database blocks are always written to the global cache at the time each update takes place.

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

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

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

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

If the redo pool specification is too small and the pool runs full, the nucleus will write additional updated blocks to the global cache before the surrounding transactions end. Such additional cache writes may hurt system performance. To obtain more information about the current usage of the redo pool, review the redo high-water mark in the output of the DRES operator command and in the cluster nucleus shutdown statistics. This information is also available in corresponding Adabas Online System (AOS) displays and in the output of the ADADBS OPERCOM function.

MXCANCEL: Timeout Threshold for Canceled Peer Nucleus

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

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

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

- It returns response 124, subcode 28 (no reply) to the user on whose behalf the intracluster command was issued.

- It terminates itself abnormally if the communication occurred on behalf of an internal process that must not fail.

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

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

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

MXCANCELWARN : Timeout Threshold for Canceled Peer Nucleus Warning

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

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

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

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

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

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

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

MXMSG: Timeout Threshold for Internucleus Command Processing

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

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

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

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

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

- Problems have occurred in the cluster nucleus, such as a CPU loop or a resource deadlock. In such cases, the nucleus experiencing the problem must be terminated to resolve the problem and the sooner this happens, the sooner the other cluster nuclei can recover from the failure and get back to normal processing.

- Problems have occurred outside the cluster nucleus, such as CPU starvation or a problem in the host system. In such cases, the cause of the delayed response might possibly be removed by system measures unrelated to the cluster database, and letting the other cluster nuclei wait some time for a resolution might prevent an unnecessary abnormal termination of a cluster member.

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

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

MXMSGWARN : Timeout Threshold for Internucleus Command Processing Warning

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

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

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

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

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

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

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

MXWTOR : Self-Termination Operator Query Interval

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

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

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

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

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

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

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

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

NUCID : Adabas Cluster Nucleus ID

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

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

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

Guidelines

- Each nucleus being run must have its own Work data set.

- All nuclei in a cluster must be run against the same database.

- Software AG strongly recommends that you run utilities or user applications against a nucleus in the local z/OS image. Although a remote open nucleus may be used, performance degradation due to network overhead is likely.

Example

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

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

PROGRAM : Program to Run

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

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

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

Examples

The following example specifies that an Adabas nucleus is running.

```
ADARUN PROGRAM=ADANUC
```

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

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

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

```
ADARUN PROGRAM=NETWRK
```

5 Cluster Operator Commands

This chapter describes operator commands for ADACOM and for an Adabas Parallel Services cluster nucleus.

This chapter covers the following topics:

ADACOM Operator Commands

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

This section is organized in the following topics:

- Entering z/OS System Commands
- Entering z/VSE System Commands
- Entering BS2000 System Commands
- DIM - Display Image
- DN - Display Active Nuclei
- DUMP - Dump Storage Areas
- SN - Set Nucleus Status

Entering z/OS System Commands

This section describes the format for entering ADACOM operator commands on z/OS systems. It contains the following topics:

- Commands Issued during ADACOM Initialization
- Commands Issued after ADACOM Initialization

Commands Issued during ADACOM Initialization

When running the Adabas Parallel Services initialization routine ADACOM, operator commands can be specified as follows:

```
{ MODIFY | 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.

To change the pair, enter the command preceded by `SVC=svc,DBID=dbid` (the order of the SVC and DBID is interchangeable) optionally followed by a comma. For example:

{ **MODIFY | 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, and then issue the command separately.

To dynamically add an IDTNAME/DBID combination, either one that was not specified in the startup JCL or one that was terminated, use the command format above and, optionally, an `NU` parameter setting or a nondefault `FORCE` parameter setting:

{ **MODIFY | F** } *jobname* ,**SVC=** *svc* ,**DBID=** *dbid* [,**NU=** *max-users*][,**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.

To dynamically terminate an IDTNAME/DBID combination, enter the ADAEND command as follows:

{ **MODIFY | F** } *jobname* ,**SVC=** *svc* ,**DBID=** *dbid* ,**ADAEND**

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

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

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

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

Entering z/VSE System Commands

▶ **To enter operator commands in z/VSE environments, use the following steps:**

1 Enter an MSG command for the z/VSE partition in which ADACOM is executing, using the following format:

```
MSG partition-id
```

When ready for communication, ADACOM will respond with message ADAI29. In cases where the ADAEND command causes ADACOM to end its session, no outstanding reply is presented so that an orderly shutdown can occur without the need for operator intervention.

2 Enter the desired operator command or commands. More than one command can be entered in a session.

You can also enter a command followed directly by a slash (/) to end any further operator communications until the next MSG command by z/VSE.

3 When all desired commands have been entered, close the operator communication session by entering a null command (EOB).

4 To enter any more commands after the session is closed, repeat this entire procedure.

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

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

Entering BS2000 System Commands

This section describes the format for entering ADACOM operator commands on BS2000 systems.

Commands Issued during ADACOM Initialization

When running the Adabas Parallel Services initialization routine ADACOM, the operator commands described in this section can be specified as follows:

INTR (*jobnumber* **,)** *command*

where

| *jobnumber* | Job Task Number, which is four alphanumeric bytes when given from the console |

Commands Issued after ADACOM Initialization

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

To change the pair, enter the command preceded by `IDTNAME=idtname,DBID=dbid' (the order of the IDTNAME and DBID is interchangeable) optionally followed by a comma. For example:

INTR (*jobnumber* **,)IDTNAME=** *idtname* **,DBID=** *dbid* **,** *command*

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

To dynamically add an IDTNAME/DBID combination, either one that was not specified in the startup JCL or one that was terminated, use the command format above and, optionally, an NU parameter setting and/or a nondefault FORCE parameter setting:

INTR (*jobnumber* **,)IDTNAME=** *idtname* **,DBID=** *dbid* **[,NU=** *max-users* **][,FORCE=YES],** *command*

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

To dynamically terminate an IDTNAME/DBID combination, enter the ADAEND command as follows:

INTR (*jobnumber* **,)IDTNAME=** *idtname* **,DBID=** *dbid* **,ADAEND**

⚠️ **Caution:** You must specify the IDTNAME and DBID when terminating an IDTNAME/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

```
PLIO60 IDTNAME=idtname DBID=dbid OPERATOR COMMAND:xxx
```

📄 **Note:** ADACOM writes global ADACOM messages into the output data set with the link name COMPRINT. It writes messages pertinent to an individual IDTNAME/DBID combination into the output data set with the DD name 'Pssddddd', where 'ss' is a unique number derived from the IDTNAME (counting backward from 55) and 'ddddd' is the database ID.

DIM - Display Image

DIM [*image-name*]

The DIM command displays information about each active cluster nucleus on the specified (or all) operating system image. Since with Adabas Parallel Services (in contrast to Adabas Cluster Services) all cluster nuclei run on the same system image, its output for DIM is equivalent to that for the DN command. See *DN - Display Nuclei* for sample output.

Specifying the image name is optional.

DN - Display Active Nuclei

Use DN to display the number of commands processed and the number of currently active users for each active nucleus.

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   00000089 00000000
```

The display uses the following indicators:

image-name	the name of the image
jobname	the job name of a cluster nucleus active on the local image
NUCID	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 open (Y); or closed (N) for new users
RO	not applicable to Parallel Services
#USERS	the number of users currently assigned to the specified nucleus
#CMNDS	the number of commands currently in progress in the specified nucleus
LURA=	not applicable to Parallel Services
RULA=	not applicable to Parallel Services

DUMP - Dump Storage Areas

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

DUMP *stg-area*

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

SN - Set Nucleus Status

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

where

RMTALL	is not applicable
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 Adabas Parallel Services cluster users

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

Option	Action
RMTALL {OP | CL}	not applicable
image-name {OP | CL}	not applicable
nucleus-id {OP | CL}	not applicable
LCLALL {OP | CL} LCL	open or close all nuclei on the *local* image to local users. Information is not broadcast to other images.
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 Adabas Parallel Services cluster users.
nucleus-id {OP | CL} GBL	open or close the specified *local* nucleus to all Adabas Parallel Services cluster users.

By default, nuclei start open to users.

After the nuclei start, the SN operator commands may be used to exercise some control over the assignment of users to nuclei.

Adabas Cluster Nucleus Operator Commands

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

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

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

DMEMTB - Display Member State Table

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

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

Sample Output

```
ADAX61 00226 2009-06-24 16:52:00 Member Status Table
ADAX61 00226 2009-06-24 16:52:00 Other members:        1
ADAX61 00226 2009-06-24 16:52:00   This system:        1
ADAX61 00226 2009-06-24 16:52:00
ADAX61 00226 2009-06-24 16:52:00        NUCID:   2,261
ADAX61 00226 2009-06-24 16:52:00    Internal ID:       2
ADAX61 00226 2009-06-24 16:52:00        System: z/OS
ADAX61 00226 2009-06-24 16:52:00      Job Name: USACHPS1
ADAX61 00226 2009-06-24 16:52:00   Flags 1/2/3: D7/00/00
ADAX61 00226 2009-06-24 16:52:00
ADAX61 00226 2009-06-24 16:52:00        NUCID:   2,262
ADAX61 00226 2009-06-24 16:52:00        System: z/OS
ADAX61 00226 2009-06-24 16:52:00      Job Name: USACHPS2
ADAX61 00226 2009-06-24 16:52:00   Flags 1/2/3: D7/00/00
ADAN41 00226 2009-06-24 16:52:00 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.

DPARM - Display Adabas Session Parameters

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

Sample Output

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

DPPT - Display Parallel Participant Table (PPT)

Use DPPT to display all occupied 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             UCID: 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 Global 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  Area 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  Area 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  Area 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   Area 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   Area 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   Area 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   Area 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   Area 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   Area 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  Area 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  Area 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  Area 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   Area 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   Area 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 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
```

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  Area 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  Area 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   Area 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   Area 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   Area 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  Area 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  Area 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  Area 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  Area 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  Area 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 Global lock statistics:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
```

General Control Block Lock

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

Security Lock

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

FST Lock

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

File Lock Table Lock

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

Online Save Lock

```
ADAX61 00006 2000-09-06 19:29:23 5. Online save lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :           0
ADAX61 00006 2000-09-06 19:29:23              Granted       :           0
ADAX61 00006 2000-09-06 19:29:23              Rejected      :           0
ADAX61 00006 2000-09-06 19:29:23           Unconditional    :           0
ADAX61 00006 2000-09-06 19:29:23           Synchronous      :           0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous     :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued          :           0
ADAX61 00006 2000-09-06 19:29:23           Synchronous      :           0
ADAX61 00006 2000-09-06 19:29:23           Asynchronous     :           0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
```

Buffer Flush Lock

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

Global ET Sync Lock

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

Recovery Lock

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

Hold ISN Locks

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

Unique Descriptor Locks

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

ETID Locks

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

New Data RABN Locks

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

Checkpoint Lock

```
ADAX61 00006 2000-09-06 19:29:23 13. Checkpoint lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :          0
ADAX61 00006 2000-09-06 19:29:23           Granted          :          0
ADAX61 00006 2000-09-06 19:29:23           Rejected         :          0
ADAX61 00006 2000-09-06 19:29:23           Unconditional    :          6
ADAX61 00006 2000-09-06 19:29:23           Synchronous      :          6
ADAX61 00006 2000-09-06 19:29:23           Asynchronous     :          0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued          :          6
ADAX61 00006 2000-09-06 19:29:23           Synchronous      :          6
ADAX61 00006 2000-09-06 19:29:23           Asynchronous     :          0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
```

ET Data Lock

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

Global Update Command Sync Lock

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

Parameter Lock

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

DXMSG - Display Messaging Performance Statistics

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

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

Sample Output

```
17:28:45 ADAI29 Oper cmd: DXMSG
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messaging statistics
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message Control Block statistics
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Allocated                    224
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Used                           2
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Total requests             1,567
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Statistics for ACMD-type messages
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages sent                771
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages arrived             796
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages accepted            796
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Replies sent                 794
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Statistics for XCF  transport service
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages subject to MXMSG
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message duration in seconds
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Minimum       0.000021
```

```
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Maximum          0.050979
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Mean             0.004110
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Std Dev          0.004507
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message count                686
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1000   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >   10   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >    1   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100  ms    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >   10  ms    8%              56
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >    1  ms   81%             553
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100  us   11%              74
17:28:45 ADAX16 00226 2007-06-01 17:28:44 <= 100  us    0%               3
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Messages not subject to MXMSG
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message duration in seconds
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Minimum          0.000938
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Maximum          0.047989
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Mean             0.010455
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Std Dev          0.009466
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message count                 83
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1000   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >   10   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >    1   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100  ms    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >   10  ms   37%              31
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >    1  ms   61%              51
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100  us    1%               1
17:28:45 ADAX16 00226 2007-06-01 17:28:44 <= 100  us    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Total for all messages
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message duration in seconds
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Minimum          0.000021
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Maximum          0.050979
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Mean             0.004795
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Std Dev          0.003194
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAX16 00226 2007-06-01 17:28:44 Message count                769
17:28:45 ADAX16 00226 2007-06-01 17:28:44 > 1000   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >   10   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >    1   s    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100  ms    0%               0
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >   10  ms   11%              87
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >    1  ms   79%             604
17:28:45 ADAX16 00226 2007-06-01 17:28:44 >  100  us   10%              75
17:28:45 ADAX16 00226 2007-06-01 17:28:44 <= 100  us    0%               3
```

```
17:28:45 ADAX16 00226 2007-06-01 17:28:44
17:28:45 ADAN41 00226 2007-06-01 17:28:44 Function completed
```

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

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

DXSTAT - Display Cache and Lock Statistics

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

The full set of global 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 global cache statistics in the sections *Totals*, *Data Storage*, and *Normal Index* are displayed using this command. All file cache statistics for files and all global lock statistics are displayed.

Sample Output

```
ADAI29 OPER CMD: DXSTAT
```

Global Cache Statistics

```
ADAX61 00006 2000-09-06 19:29:23 Global 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
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  Area 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  Area full      :                  0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates       :             65,552
ADAX61 00006 2000-09-06 19:29:23  Block invalid  :                  0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads  :              1,727
ADAX61 00006 2000-09-06 19:29:23  Synchronous    :                265
ADAX61 00006 2000-09-06 19:29:23  Asynchronous   :              1,462
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes         :                  0
ADAX61 00006 2000-09-06 19:29:23  Timeouts       :                  0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 AC:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads           :                 11
ADAX61 00006 2000-09-06 19:29:23  Synchronous    :                 11
ADAX61 00006 2000-09-06 19:29:23  Asynchronous   :                  0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23  In cache       :                  6
ADAX61 00006 2000-09-06 19:29:23  Not in cache   :                  5
ADAX61 00006 2000-09-06 19:29:23  Area 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  Area full      :                  0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates       :              8,772
ADAX61 00006 2000-09-06 19:29:23  Block invalid  :                  0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads  :                 38
ADAX61 00006 2000-09-06 19:29:23  Synchronous    :                 38
ADAX61 00006 2000-09-06 19:29:23  Asynchronous   :                  0
```

```
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes        :                          0
ADAX61 00006 2000-09-06 19:29:23  Timeouts      :                          0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 DS:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads         :                      1,609
ADAX61 00006 2000-09-06 19:29:23  Synchronous  :                          0
ADAX61 00006 2000-09-06 19:29:23  Asynchronous :                      1,609
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23  In cache     :                        855
ADAX61 00006 2000-09-06 19:29:23  Not in cache :                        754
ADAX61 00006 2000-09-06 19:29:23  Area 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  Area full    :                          0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates     :                      6,603
ADAX61 00006 2000-09-06 19:29:23  Block invalid :                         0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads :                     1,461
ADAX61 00006 2000-09-06 19:29:23  Synchronous  :                          0
ADAX61 00006 2000-09-06 19:29:23  Asynchronous :                      1,461
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes        :                          0
ADAX61 00006 2000-09-06 19:29:23  Timeouts      :                          0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 DSST:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads         :                          1
ADAX61 00006 2000-09-06 19:29:23  Synchronous  :                          1
ADAX61 00006 2000-09-06 19:29:23  Asynchronous :                          0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23  In cache     :                          0
ADAX61 00006 2000-09-06 19:29:23  Not in cache :                          1
ADAX61 00006 2000-09-06 19:29:23  Area 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  Area full     :                        0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates      :                    3,969
ADAX61 00006 2000-09-06 19:29:23  Block invalid :                        0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads :                       34
ADAX61 00006 2000-09-06 19:29:23  Synchronous   :                       33
ADAX61 00006 2000-09-06 19:29:23  Asynchronous  :                        1
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes        :                        0
ADAX61 00006 2000-09-06 19:29:23  Timeouts      :                        0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 FCB:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads          :                        2
ADAX61 00006 2000-09-06 19:29:23  Synchronous   :                        2
ADAX61 00006 2000-09-06 19:29:23  Asynchronous  :                        0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23  In cache      :                        0
ADAX61 00006 2000-09-06 19:29:23  Not in cache  :                        2
ADAX61 00006 2000-09-06 19:29:23  Area 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  Area 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  Area 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  Area 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  Area 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  Area 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  Area 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  Area full   :                        0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates   :                    3,969
ADAX61 00006 2000-09-06 19:29:23  Block invalid :                     0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads :                    34
ADAX61 00006 2000-09-06 19:29:23  Synchronous  :                      33
ADAX61 00006 2000-09-06 19:29:23  Asynchronous :                       1
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes     :                        0
ADAX61 00006 2000-09-06 19:29:23  Timeouts    :                        0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 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  Area 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  Area full    :                       0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Validates     :                  61,531
ADAX61 00006 2000-09-06 19:29:23  Block invalid :                      0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Cast-out reads :                  1,677
ADAX61 00006 2000-09-06 19:29:23  Synchronous  :                     221
ADAX61 00006 2000-09-06 19:29:23  Asynchronous :                   1,456
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Deletes       :                       0
ADAX61 00006 2000-09-06 19:29:23  Timeouts     :                       0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 File     9:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Reads         :                       8
ADAX61 00006 2000-09-06 19:29:23  Synchronous  :                       6
ADAX61 00006 2000-09-06 19:29:23  Asynchronous :                       2
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23  In cache     :                       0
ADAX61 00006 2000-09-06 19:29:23  Not in cache :                       8
ADAX61 00006 2000-09-06 19:29:23  Area 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  Area 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
```

Global Lock Statistics

```
ADAX61 00006 2000-09-06 19:29:23 Global lock statistics:
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 1. GCB lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :         0
ADAX61 00006 2000-09-06 19:29:23           Granted         :         0
ADAX61 00006 2000-09-06 19:29:23           Rejected        :         0
ADAX61 00006 2000-09-06 19:29:23          Unconditional    :         0
ADAX61 00006 2000-09-06 19:29:23          Synchronous      :         0
ADAX61 00006 2000-09-06 19:29:23          Asynchronous     :         0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued          :         0
ADAX61 00006 2000-09-06 19:29:23          Synchronous      :         0
ADAX61 00006 2000-09-06 19:29:23          Asynchronous     :         0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 2. Security lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :         0
ADAX61 00006 2000-09-06 19:29:23           Granted         :         0
ADAX61 00006 2000-09-06 19:29:23           Rejected        :         0
ADAX61 00006 2000-09-06 19:29:23          Unconditional    :         0
ADAX61 00006 2000-09-06 19:29:23          Synchronous      :         0
ADAX61 00006 2000-09-06 19:29:23          Asynchronous     :         0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued          :         0
ADAX61 00006 2000-09-06 19:29:23          Synchronous      :         0
ADAX61 00006 2000-09-06 19:29:23          Asynchronous     :         0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 3. FST lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional      :         0
ADAX61 00006 2000-09-06 19:29:23           Granted         :         0
ADAX61 00006 2000-09-06 19:29:23           Rejected        :         0
ADAX61 00006 2000-09-06 19:29:23          Unconditional    :         0
ADAX61 00006 2000-09-06 19:29:23          Synchronous      :         0
ADAX61 00006 2000-09-06 19:29:23          Asynchronous     :         0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued          :         0
ADAX61 00006 2000-09-06 19:29:23          Synchronous      :         0
ADAX61 00006 2000-09-06 19:29:23          Asynchronous     :         0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
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
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
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
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
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
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
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
ADAX61 00006 2000-09-06 19:29:23 16. Parameter lock
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Obtains - Conditional       :            0
ADAX61 00006 2000-09-06 19:29:23              Granted         :            0
ADAX61 00006 2000-09-06 19:29:23              Rejected        :            0
ADAX61 00006 2000-09-06 19:29:23          Unconditional       :            0
ADAX61 00006 2000-09-06 19:29:23              Synchronous     :            0
ADAX61 00006 2000-09-06 19:29:23              Asynchronous    :            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23 Releases - Issued            :            0
ADAX61 00006 2000-09-06 19:29:23              Synchronous     :            0
ADAX61 00006 2000-09-06 19:29:23              Asynchronous    :            0
ADAX61 00006 2000-09-06 19:29:23
ADAX61 00006 2000-09-06 19:29:23
ADAN41 00006 2000-09-06 19:29:23 Function completed
```

MXCANCEL - Dynamically Change MXCANCEL Parameter Setting

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

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

For more information about the MXCANCEL setting for a nucleus, read about the ADARUN MXCANCEL parameter in *MXCANCEL: Timeout Threshold for Canceled Peer Nucleus*, elsewhere in this guide.

MXCANCELWARN - Dynamically Change MXCANCELWARN Parameter Setting

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

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

For more information about the MXCANCELWARN setting for a nucleus, read about the ADARUN `MXCANCELWARN` parameter in *MXCANCELWARN : Timeout Threshold for Canceled Peer Nucleus Warning*, elsewhere in this guide.

MXMSG - Dynamically Change MXMSG Parameter Setting

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

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

For more information about the MXMSG setting for a nucleus, read about the ADARUN `MXMSG` parameter in *MXMSG: Timeout Threshold for Internucleus Command Processing*, elsewhere in this guide.

MXMSGWARN - Dynamically Change MXMSGWARN Parameter Setting

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

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

For more information about the MXMSGWARN setting for a nucleus, read about the ADARUN MXMSGWARN parameter in *MXMSGWARN : Timeout Threshold for Internucleus Command Processing Warning*, elsewhere in this guide.

MXWTOR - Dynamically Change MXWTOR Parameter Setting

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

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

For more information about the MXWTOR setting for a nucleus, read about the ADARUN MXWTOR parameter in *MXWTOR : Self-Termination Operator Query Interval*, elsewhere in this guide.

Index