

Adabas Cluster Services

Operations

Version 7.4.2

September 2009

Adabas Cluster Services

This document applies to Adabas Cluster Services Version 7.4.2 and to all subsequent releases.

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

Copyright © Software AG 2009. All rights reserved.

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

Table of Contents









1 Operations	1
2 Initialization	3
Sequence	4
ADACOM Process	4
Adabas Cluster Nucleus Process	5
PPT Processing	6
3 Termination	9
Normal Termination	10
Abnormal Termination	11
4 Backout Processing	13
5 Restart/Recovery Processing	15
Offline Recovery (Session Autorestart)	16
Online Recovery	16
Automatic Restart Management (ARM)	17
Archive Recovery	17
6 Planning an Outage	19
7 Utility Processing	21
ADADBS OPERCOM Commands	22
ADADBS REFRESHSTATS - Refresh Statistical Values	23
ADAICK PPTPRINT - Print/Dump Parallel Participant Table	23
ADAPLP IPLOGPRI - Print Sequential Intermediate Datasets	23
ADARAI - Adabas Recovery Aid	24
ADAREP - Checkpoint Information Extended	24
ADARES CLCOPY - Copy/Merge Nucleus Cluster Command Logs	24
ADARES PLCOPY - Copy/Merge Nucleus Cluster Protection Logs	25
ADARES PLCOPY NOPPT - Ignore PPT	27
ADARES MERGE CLOG - Merge Nucleus Cluster Command Logs	29
ADARES BACKOUT and REGENERATE - Uniquely Identifying Checkpoints	30
ADASAV Processing Change	31
ADASAV RESTPLOG -- Uniquely Identifying Checkpoints	31
8 Cluster Nucleus Session End Statistics	33
General Nucleus Information	34
Input/Output Statistics	34
Command Statistics	35
User Statistics	36
Efficiency Statistics	36
External Cache Statistics (Cluster Nucleus Only)	37
External Lock Statistics (Cluster Nucleus Only)	42
Dataset Activity Statistics	47
9 Adabas Online System Cluster Environment Screens	49
Display Cluster Members	50
Nucleus File Status	51

Nucleus Status Flags	52
Cluster Usage	54
Maintain the User Table	61
Index	63

1 Operations

This documentation provides information about initialization, termination, backout, restart, and recovery processes in an Adabas cluster environment. It tells you how to plan an outage and how to use the utility functions that are provided specifically for cluster environments. Finally, it provides a sample of the session-end statistics produced for a clustered nucleus.

The Adabas Cluster Services Operations documentation is organized in the following topics:

	<i>Initialization</i>
	<i>Termination</i>
	<i>Backout Processing</i>
	<i>Restart/Recovery Processing</i>
	<i>Planning an Outage</i>
	<i>Utility Processing</i>
	<i>Cluster Nucleus Session End Statistics</i>
	<i>Adabas Online System Cluster Environment Screens</i>

2 Initialization

- Sequence 4
- ADACOM Process 4
- Adabas Cluster Nucleus Process 5
- PPT Processing 6

This section provides information about initialization in an Adabas cluster environment.

This chapter covers the following topics:

Sequence

Due to the interdependence among Entire Net-Work, ADACOM, and Adabas cluster nuclei, these programs have certain co-occurrence requirements.

- Whether currently running or not, ADACOM must have been executed in the local operating system image when there are cluster users but no cluster nuclei to set up the environment.
- Entire Net-Work can start before or after ADACOM or a nucleus is active.

Note that although ADACOM and several Adabas cluster nuclei could run without Entire Net-Work within a single operating system image, there could then be no Adabas Cluster Services programs for that database in any other operating system image.

- A nucleus will not start if another nucleus with the same NUCID (whether local or remote) is already active.

ADACOM Process

The ADACOM initialization process is recorded in PLInnn messages.

All PLInnn messages are printed to the console. PLI001-049 messages are specific to a particular SVC/DBID set and are written to the SYSOUT dataset dynamically allocated for that set; starting with PLI050, the messages apply to ADACOM in general and are written to the COMPRINT dataset.

A new Adabas Cluster Services control block is acquired if none yet exists.



Note: Prior to PLI050, the system default ADARUN parameter values for MODE, DBID, DEVICE, SVC, and AMODE are displayed. These are not the values for the current ADACOM.

```
PLI050 00161  INITIALIZING ADACOM
PLI002 00161  INITIALIZING DBID=dbid SVC=svc
                ACQUIRING NEW PLXCB
                PLXCB IS LOCATED AT address
PLI063 00161  PROCESSING: ADACOM SVC=svc,DBID=dbid,NU=users
                INITIALIZATION COMPLETE
```


Adabas Cluster Nucleus Process

Each Adabas cluster nucleus serializes during initialization by means of an enqueue. This is done because each nucleus puts information into the CSA nucleus table.

```

PLX050 00161 ADACLU INIT DBID=00161 NUCID=00132
PLX006 00161 PLXCB LOCATED AT 0C893000
ADAN03 00161 INITIALIZING NUCID=132 INTNUCID=2
ADAX20 00161 XCF TRANSPORT INITIALIZATION COMPLETE
ADAX28 00161 IXCJOIN XCFTI RET 00000004 RSN 00000004
ADAX28 00161 IXCQUERY XCFTI RET 00000000 RSN 00000000
PLX059 00161 INITIALIZATION OF ADACLU COMPLETE
ADAN03 00161 ADABAS COMING UP
ADAX31 00161 OPENING WORK DATASET FOR NUCID=132

```

Access to the coupling facility lock structure is established and acknowledged by messages issued by the operating system:

```

IXL014I IXLCONN REQUEST FOR STRUCTURE ADA_LOCK11
WAS SUCCESSFUL. JOBNAME: USADFM7 ASID: 00C0
CONNECTOR NAME: DB00006P00002N02 CFNAME: DCF1
IXL015I STRUCTURE ALLOCATION INFORMATION FOR
STRUCTURE ADA_LOCK11, CONNECTOR NAME DB00006P00002N02
  CFNAME      ALLOCATION STATUS/FAILURE REASON
  -----
  DCF1        STRUCTURE ALLOCATED
  DCF2        PREFERRED CF ALREADY SELECTED

```

Adabas Cluster Services follows the operating system messages with lock structure statistics:

```

ADAX70 00161 CONNECTED TO LOCK STRUCTURE ADA_LOCK11
ADAX70 00161 NUMBER OF LOCK ENTRIES          32,768
ADAX70 00161 MAX NUMBER OF RECORD ELEMENTS   11,184

```

Access to the coupling facility cache structure is established and acknowledged by messages issued by the operating system:

```

IXL014I IXLCONN REQUEST FOR STRUCTURE ADA_CACHE11
WAS SUCCESSFUL. JOBNAME: USADFM7 ASID: 00C0
CONNECTOR NAME: DB00006P00002N02 CFNAME: DCF1
IXL015I STRUCTURE ALLOCATION INFORMATION FOR
STRUCTURE ADA_CACHE11, CONNECTOR NAME DB00006P00002N02
  CFNAME      ALLOCATION STATUS/FAILURE REASON
  -----
  DCF1        STRUCTURE ALLOCATED
  DCF2        PREFERRED CF ALREADY SELECTED

```

Adabas Cluster Services follows the operating system messages with cache structure statistics:

```
ADAX57 00161 CONNECTED TO CACHE STRUCTURE ADA_CACHE11
ADAX57 00161 DIRECTORY ELEMENTS          11,490
ADAX57 00161 DATA      ELEMENTS          2,872
ADAX57 00161 DATA ELEMENT SIZE          1,024
ADAN19 00161 BUFFER FLUSH IS  A S Y N C H R O N O U S
ADAN01 00161 A D A B A S  Vv.r.s  IS ACTIVE
ADAN01 00161 MODE = MULTI
ADAN01 00161 RUNNING WITHOUT RECOVERY LOG
```

PPT Processing

The parallel participant table (PPT), which exists for both cluster and noncluster nuclei, is used to determine if any PLOGs still need to be copied from previous sessions. If the PPT indicates that PLOGs remain to be copied, the PLOG datasets are read and, if necessary, the user exit (user exit 2 or user exit 12) is invoked.

First Sysplex Cluster Nucleus Starts

The Adabas sysplex cluster nucleus that is the first to initialize checks all the PLOG entries from the previous session for all nuclei and marks any that are "still being written" to completed status. In this way, the user exit (user exit 2 or user exit 12) need not be called each time a cluster nucleus autostarts. The first nucleus then calls the user exit but waits only if the PLOGs that need to be copied are for its own NUCID.

Subsequent Sysplex Cluster Nucleus Starts

A subsequent cluster nucleus checks only its own PLOGs and invokes user exit 2 or user exit 12 if the PLOGs are still not copied/merged. It waits if the user exit instructs it to. If there is no user exit 2 or user exit 12, it overwrites the PLOGs.

Noncluster Nucleus Starts

A noncluster nucleus checks whether the previous session was a sysplex cluster session and has a pending autorestart. If so, the noncluster nucleus is not allowed to start.

If PLOGs from a previous sysplex cluster session remain to be copied, ADARES invokes the merge or the PLCOPY as required. A noncluster nucleus always uses block 1 of the PPT and can only overwrite it when PLOGs from previous sessions have been processed to completion.

A user exit 2 or user exit 12 controls the copy/merge process. If there is no user exit 2 or user exit 12, the PLOG and PPT entry are overwritten.

Different PLOG Detected

If PLOGRQ=FORCE is specified and an uncopied PLOG is detected that does not match that specified in the last session, a parameter error occurs. If the PLOG has been copied, the PPT entry is overwritten and the nucleus starts.

3 Termination

- Normal Termination 10
- Abnormal Termination 11

This section provides information about termination in an Adabas cluster environment.

This chapter covers the following topics:

Normal Termination

Entire Net-Work

Entire Net-Work may be stopped while ADACOM and/or cluster nuclei are active.

If the local Entire Net-Work stops while remote nuclei are still active, the remote nuclei are effectively no longer active. That is, users in the local operating system image will receive response code 148 for commands that are to be routed to any of the remote nuclei.

When Entire Net-Work is restarted, the environment is reset by the ADACOM module on operating system images that have users but no cluster nuclei. If on such an image

- the ADACOM module remained operational after initialization, it automatically resets the environment
- the ADACOM was quiesced after initialization, you must rerun it to reset the environment.

In lieu of ADACOM, the environment is reset on operating system images that have one or more active nuclei when the first user issues a command.

ADACOM

If ADACOM is used only to initialize a sysplex cluster environment, it can subsequently be stopped ("quiesced") for batch operation or retained in operation as a command manager. ADACOM can be restarted at any time.

On operating system images that have users but no cluster nuclei, Software AG recommends that you keep ADACOM in operation as well so that it is available to reset the environment if Entire Net-Work goes down for any reason and comes back up.

Adabas Cluster Nuclei

If the Adabas operator command ADAEND or HALT is issued, the nucleus will stop with no pending autorestart. The other active nuclei in the cluster continue processing normally.

```
ADAN51 00006 2001-02-13 23:05:54 OPERATOR TYPE-IN: ADAEND
ADAN42 00006 2001-02-13 23:05:54 FUNCTION ACCEPTED
```

The operating system issues the following lock structure statistics:

```
IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE ADA_LOCK11,
CONNECTOR DB00006P00132N02:
 00010019
 00000000 00000000 00000000 00000000
 00000000 00000000 00000000 00000000
 00000000 00000000 00000000 00000000
 00000001 00000000 00000000 00000000
 00000000 00000000 00000000 00000000
 00000000 00000000 00000000 00000000
 00000002 00000000 00000000 00000000
 00000000 00000000 00000000 00000000
 00000000 00000000 00000000 00000000
IXL031I CONNECTOR CLEANUP FOR LOCK STRUCTURE ADA_LOCK11,
CONNECTOR DB00006P00132N02, HAS COMPLETED.
INFO: 00010019 00000000 00000000 00000000 00000000 00000000
```

Adabas Cluster Services follows the operating system messages with

```
ADAX28 00161 IXCLEAVE XCFTT RET 00000000 RSN 00000000
ADAM97 00132 THIS ASCB/INITIATOR WILL BE TERMINATED BY MVS AT EOJ
```

Abnormal Termination

Entire Net-Work

The description for Entire Net-Work normal termination in section [Normal Termination](#) also applies to an Entire Net-Work abnormal termination. Adabas Cluster Services makes no distinction.

ADACOM

If ADACOM terminates abnormally, a PLInnn error message is produced to explain the problem.

Adabas Cluster Nuclei

When an Adabas sysplex cluster nucleus terminates abnormally, each surviving peer nucleus performs "online recovery". See the section [Restart/Recovery Processing](#) for more information.

The online recovery process synchronizes with a normal shut-down process that is already in progress for a failed peer nucleus. If the normal shut-down process has just begun, it is interrupted and canceled and the online recovery process replaces it. If the normal shut-down process is well underway, it proceeds and finishes; the online recovery process is not issued for the failed nucleus. The online recovery process

- synchronizes online recovery with a newly starting nucleus;
- waits three seconds for open transactions to complete before interrupting all open transactions;
- waits 0.3 seconds for active commands to complete before interrupting all active user commands;
- interrupts all activity going on in the nucleus;
- cleans up;
- disconnects from the lock and cache structures;
- performs session autorestart or waits for another nucleus to do it;
- reconnects to the lock and cache structures;
- prints messages when an online save, ADAEND, or HALT process is canceled; and
- resumes normal processing.

4 Backout Processing

Normal backout processing includes

- BT command processing;
- backing out an update command that received a nonzero response code; and
- internal transaction backout due to, for example, a timeout.

Cluster nuclei perform normal Adabas backout processing. However, each cluster nucleus invokes backout logic from its own Work dataset, ignoring the protection record timestamps.

5 Restart/Recovery Processing

- Offline Recovery (Session Autorestart) 16
- Online Recovery 16
- Automatic Restart Management (ARM) 17
- Archive Recovery 17

Restart/recovery occurs if a cluster nucleus fails. Restart/recovery uses the Work datasets of all nuclei to recover the database. The Work datasets are dynamically allocated from the dataset names recorded in the PPT. Adabas Cluster Services 7.4 supports offline and online recovery.

This chapter covers the following topics:

Offline Recovery (Session Autorestart)

- If a cluster nucleus session terminates, start one of the cluster nuclei to invoke autorestart.
- If a noncluster nucleus session terminates, restart the noncluster nucleus to invoke autorestart.

Offline recovery occurs if all active cluster nuclei in an Adabas sysplex cluster fail. Offline recovery relies only on information from the physical database and the Work datasets of each cluster nucleus. All information in the coupling facility is lost.

The first cluster nucleus to restart repairs any physical inconsistencies in the database and backs out all incomplete commands and transactions. The restarted nucleus obtains recovery information from blocks in the common database and from the Work datasets of all the failed nuclei.

The restarting nucleus retrieves the Work dataset names from the PPT block for each terminated nucleus and opens these datasets using dynamic allocation. From that point, normal recovery processing occurs:

- the breakpoint on each Work dataset is found;
- backward and forward repair is performed; and
- autobackout is performed.

While reading through the Work datasets, the restarting nucleus on the fly merges the protection records by their timestamps into chronological sequence.

Online Recovery

When one or more cluster nuclei have failed while one or more other nuclei in the same cluster remain active, online recovery processing is performed by collaboration of all surviving nuclei.

All surviving cluster nuclei quiesce their operations and reinitialize their working storage. Command processing is quiesced and the internal status variables, tables, and pools are repaired.

The peer nuclei compete for the recovery lock: when one of the nuclei obtains it, it invokes offline recovery processing. It repairs any physical inconsistencies in the database and backs out all incomplete command and transactions. Open transactions executed by the surviving nuclei are backed out as well. All information in the lock and cache structures is discarded.

Once this recovery processing has completed, normal processing resumes.

Users are affected by online recovery as follows:

- users assigned to failed nuclei lose their commands, transactions, sequential processes, and search results. They may receive response codes 9, 21, 148, or 251, depending on the status of their session at the time of the failure.
- users assigned to surviving nuclei may or may not lose their commands/transactions, depending on whether they managed to complete them in the quiesce phase. They retain their sequential processes and search results, but they may experience an increased response time. Users that do lose their commands/transactions will subsequently receive response code 9 and might possibly get response code 21 as well.

Automatic Restart Management (ARM)

Automatic restart management (ARM) is an OS/390 and z/OS facility that can be used to automatically restart a nucleus when it ABENDs. Automatic restart is suppressed when the ABEND is intentional; for example, when it results from a parameter error.

ARM can be used for Adabas nuclei in both cluster and noncluster environments.

The ADARUN parameter `ARMNAME` is used to identify the element in the ARM 'policy' that is to be activated. Each element specifies when, where, and how often an automatic restart is to be attempted.

If an ARM policy has not been defined, the `ARMNAME` parameter has no effect.

Archive Recovery

Archive recovery occurs if the container datasets of the database are damaged or restart/recovery is not effective.

Archive recovery

- restores the database; and
- regenerates the updates from the protection logs.

The protection logs to be regenerated are the output of the ADARES PLCOPY protection log copy and merge process that occurs in sysplex cluster environments. The restore/regenerate process is the same in both cluster and noncluster environments.

6

Planning an Outage

The Adabas Cluster Services solution permits the database administrator to migrate an Adabas nucleus to another operating system image in the sysplex so that a planned outage due to system changes or preventive maintenance on one machine need not impact any other part of the system.

▶ **To migrate the nucleus to a different operating system image**

- 1 Quiesce the nucleus using ADAEND.
- 2 Start the nucleus in a different operating system image.

The nucleus automatically accepts commands.

7 Utility Processing

▪ ADADBS OPERCOM Commands	22
▪ ADADBS REFRESHSTATS - Refresh Statistical Values	23
▪ ADAICK PPTPRINT - Print/Dump Parallel Participant Table	23
▪ ADAPLP IPLOGPRI - Print Sequential Intermediate Datasets	23
▪ ADARAI - Adabas Recovery Aid	24
▪ ADAREP - Checkpoint Information Extended	24
▪ ADARES CLCOPY - Copy/Merge Nucleus Cluster Command Logs	24
▪ ADARES PLCOPY - Copy/Merge Nucleus Cluster Protection Logs	25
▪ ADARES PLCOPY NOPPT - Ignore PPT	27
▪ ADARES MERGE CLOG - Merge Nucleus Cluster Command Logs	29
▪ ADARES BACKOUT and REGENERATE - Uniquely Identifying Checkpoints	30
▪ ADASAV Processing Change	31
▪ ADASAV RESTPLOG -- Uniquely Identifying Checkpoints	31

Like normal users, utilities are assigned to a local nucleus, if available; to a remote nucleus, otherwise.

Adabas Cluster Services automatically synchronizes with database changes made by utilities.

See the *Adabas Utilities* documentation for specific information about utility functions used in an Adabas parallel sysplex cluster environment.

This chapter covers the following topics:

ADADBS OPERCOM Commands

Changes have been made for ADADBS OPERCOM command processing in an Adabas cluster nucleus environment.

Global Commands

The following ADADBS OPERCOM commands have a "GLOBAL" option for routing the following commands to all nuclei in an Adabas cluster:

ADAEND, CANCEL, FEOFCL, FEOFPL, HALT.

For example:

```
ADADBS OPERCOM ADAEND,GLOBAL
```

When "GLOBAL" is specified, a response code 148 is returned to ADADBS if any one of the nuclei is down. When "GLOBAL" is *not* specified, a specific NUCID from the cluster must be specified and the command is sent to that NUCID.

Routing a Command to a Specific Nucleus

The NUCID option allows you to direct the OPERCOM commands to a particular nucleus in the cluster for execution.

The OPERCOM function's NUCID option is specified in a manner similar to the ADARUN NUCID parameter.

For example:

```
ADADBS OPERCOM DSTAT,NUCID=3
```

sends the DSTAT command to the Adabas cluster nucleus designated with NUCID=3.

For inherently global commands, such as changing the setting of the TT parameter, the NUCID parameter is ignored.

Routing a Command to All Cluster Nuclei

When the NUCID option in the ADADBS OPERCOM function is not specified, the command is sent to all cluster nuclei and information is displayed for each nucleus in sequence.

ADADBS REFRESHSTATS - Refresh Statistical Values

The REFRESHSTATS function resets statistical values maintained by the Adabas nucleus for its current session. Parameters may be used to restrict the function to particular groups of statistical values.

In cluster environments, you must specify the specific nucleus (NUCID) for which statistical values are to be refreshed. If NUCID is not specified, statistical values will be refreshed for all active nuclei in the cluster.

ADAICK PTPRINT - Print/Dump Parallel Participant Table

The PTPRINT function has been added to the Adabas ADAICK utility to support an Adabas cluster environments. It is used to dump/print the parallel participant table (PPT) for the Adabas cluster.

Each of the 32 blocks (RABNs) allocated for the PPT represents a single nucleus in the cluster and comprises

- a single header of fixed length; and
- multiple entries of variable length.

Note that in the dump/print, 'PPH' is the tag for the PPT header and 'PPE' is the tag for the PPT entries.

ADAPLP IPLOGPRI - Print Sequential Intermediate Datasets

The IPLOGPRI function is used to print the sequential intermediate datasets created from the PLOG merge process. Input to ADAPLP IPLOGPRI must be a MERGIN1/MERGIN2 dataset created by the ADARES utility and specified in the JCL with DD name DDPLOG.

ADARAI - Adabas Recovery Aid

Adabas cluster products support the Adabas Recovery Aid (ADARAI).

ADARAI maintains a recovery log (RLOG) for each database; all nuclei in the cluster support a database write to the same RLOG and concurrent updates to the RLOG are controlled by a lock.

The ADARAI LIST function supports Adabas version 7 and above RLOGs; Adabas version 6 RLOGs are not supported.

ADAREP - Checkpoint Information Extended

Given that each cluster nucleus has its own PLOG datasets, checkpoints are no longer identified only by their name, PLOG number, and PLOG block number, but also by the ID of the nucleus that writes the checkpoint.

Several new parameters have been introduced for utilities that need to identify checkpoints on the PLOG.

ADARES CLCOPY - Copy/Merge Nucleus Cluster Command Logs

When the ADARUN parameter `CLOGMRG=YES`, the ADARES CLCOPY automatically merges the command logs for each cluster nucleus into a single log file for the cluster.

Sample JCL has been added for allocating the intermediate datasets MERGIN1 and MERGIN2 required for automated CLOG copy/merge processing in nucleus cluster environments.



Notes:

1. When intermediate datasets are used for both CLCOPY and PLCOPY, the dataset names must be unique so that they are not overwritten.
2. The dataset BLKSIZE used must be greater than or equal to the largest CLOG BLKSIZE plus eight. The LRECL must be set to the BLKSIZE minus four.

```

//ALLOC JOB
//*
//* Example to allocate the ADARES CLCOPY intermediate datasets
//*
//CM1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTFIL DD DISP=(NEW,CATLG),SPACE=(CYL,(3,10)),UNIT=SYSDA,
// VOL=SER=volser,DCB=(RECFM=VB,LRECL=23472,BLKSIZE=23476),
// DSN=EXAMPLE.CLOG.MERGIN1
//INPFIL DD *
/*
//SYSIN DD *
REPRO INFILE(INPFIL) -
OUTFILE(OUTFIL)
/*
//*
//CM2 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTFIL DD DISP=(NEW,CATLG),SPACE=(CYL,(3,10)),UNIT=SYSDA,
// VOL=SER=volser,DCB=(RECFM=VB,LRECL=23472,BLKSIZE=23476),
// DSN=EXAMPLE.CLOG.MERGIN2
//INPFIL DD *
/*
//SYSIN DD *
REPRO INFILE(INPFIL) -
OUTFILE(OUTFIL)
/*

```

ADARES PLCOPY - Copy/Merge Nucleus Cluster Protection Logs

In an Adabas nucleus cluster environment, the protection logs (and optionally, the command logs) of all individual nuclei in the cluster must be merged into single log files in chronological order for the cluster database shared by all the nuclei as a whole. The chronological order is determined by timestamps on all individual nucleus log records, which are synchronized across systems in a parallel sysplex environment by the Sysplex Timer.

Protection logs are automatically merged when an ADARES PLCOPY is executed. In an Adabas cluster environment, the PLCOPY process accesses the parallel participant table (PPT) to determine which protection logs to copy and opens the appropriate datasets using dynamic allocation. PLCOPY copies/merges as much data as possible; if a nucleus is still writing to a protection log dataset, PLCOPY 'partially' merges the dataset.

The merge begins with the lowest timestamp from all protection logs being merged and ends with the lowest of the ending timestamps from all datasets. Records beyond this point are written to an 'intermediate' dataset, which must be supplied as input to the subsequent merge. A cross-check ensures that the correct intermediate dataset has been supplied.

The following sample JCL illustrates the allocation of the intermediate datasets MERGIN1 and MERGIN2 which are required for automated PLOG copy/merge processing in nucleus cluster environments.



Notes:

1. When intermediate datasets are used for both CLCOPY and PLCOPY, the dataset names must be unique so that they are not overwritten.
2. The dataset BLKSIZE used must be greater than or equal to the largest PLOG BLKSIZE plus eight. The LRECL must be set to the BLKSIZE minus four.

```
//ALLOC JOB
//*
//* Example to allocate the ADARES PLCOPY intermediate datasets
//*
//CM1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTFIL DD DISP=(NEW,CATLG),SPACE=(CYL,(3,10)),UNIT=SYSDA,
// VOL=SER=volser,DCB=(RECFM=VB,LRECL=23472,BLKSIZE=23476),
// DSN=EXAMPLE.PLOG.MERGIN1
//INPFIL DD *
/*
//SYSIN DD *
REPRO INFILE(INPFIL) -
OUTFILE(OUTFIL)
/*
//*
//CM2 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTFIL DD DISP=(NEW,CATLG),SPACE=(CYL,(3,10)),UNIT=SYSDA,
// VOL=SER=volser,DCB=(RECFM=VB,LRECL=23472,BLKSIZE=23476),
// DSN=EXAMPLE.PLOG.MERGIN2
//INPFIL DD *
/*
//SYSIN DD *
REPRO INFILE(INPFIL) -
OUTFILE(OUTFIL)
/*
```

ADARES expects that at least one of the protection logs being merged is at 'completed' status. If this is not the case, ADARES reports that there is no data to be copied.

Sample user exits (USEREX2P and UEX12) are provided to illustrate the necessary change for the intermediate dataset.

A sample job ADARESPM is provided on the MVSJOBS dataset to illustrate the manual execution of the PLCOPY merge function. Two intermediate datasets must be supplied. ADARES analyzes the datasets to determine which is to be used as input and which for output. Specific crosschecks determine whether the correct intermediate dataset has been supplied; if not, ADARES will not

continue. Continuing without the correct input can result in lost updates and inconsistencies if the output is used for REGENERATE or BACKOUT functions.

Once DD statements for the PLOG datasets have been supplied on the session start-up JCL, you do not need to supply them again for ADARES as these are opened using dynamic allocation. If the DD statements are supplied, they are ignored.

ADARESPM Job

The following sample JCL illustrates the ADARES PLCOPY merge function:

```
//ADARESPM JOB
//*
//* ADARES : COPY/MERGE DUAL/MULTIPLE PROTECTION LOG
//* TWO COPIES OF OUTPUT ARE TO BE CREATED
//* FOR USE WITH AN ADABAS NUCLEUS CLUSTER
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1
//MARGIN1 DD DISP=SHR,DSN=EXAMPLE.PLOG.MARGIN1
//MARGIN2 DD DISP=SHR,DSN=EXAMPLE.PLOG.MARGIN2
//DDSIAUS1 DD DSN=EXAMPLE.DByyyyy.PLOG1(+1),
// VOL=SER=ADAxxx,UNIT=TAPE,DISP=(NEW,CATLG)
//DDSIAUS2 DD DSN=EXAMPLE.DByyyyy.PLOG2(+1),
// VOL=SER=ADAxxx,UNIT=TAPE,DISP=(NEW,CATLG)
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=svc,DEVICE=3380,DBID=yyyyy
/*
//DDKARTE DD *
ADARES PLCOPY TWOCOPIES
/*
```

ADARES PLCOPY NOPPT - Ignore PPT

NOPPT is for emergency use when the PPT has been overwritten. It specifies that the PPT is to be ignored and that the PLOG datasets of all cluster nuclei are being supplied with DD names DDPLOGnn in the JCL.



Caution: Use this parameter cautiously since it ignores the PPT and all control-type information typically coming from the PPT.

When you use this parameter, you must supply

- the correct intermediate dataset; and
- the correct input protection logs from all nuclei with DD names DDPLOG01-nn.

The optional parameter SBLKNUM can be used to specify the starting block number for the sequential merge output.



Caution: Without the PPT, ADARES cannot perform any extensive validations on the input datasets.

ADARESIP Job

The following sample JCL illustrates the ADARES PLCOPY NOPPT merge function:

```
//ADARESIP JOB
//*
//* ADARES : COPY/MERGE DUAL/MULTIPLE PROTECTION LOGS FROM ALL
//* NUCLEI IN AN ADABAS NUCLEUS CLUSTER
//* PPT IS TO BE IGNORED
//* THIS IS ONLY FOR EMERGENCY USE WHEN THE PPT HAS BEEN
//* OVER-WRITTEN - USE CAUTION WHEN SUBMITTING
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDPLOG01 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1.NUC1 <=== PLOG1
NUC1
//DDPLOG02 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR2.NUC1 <=== PLOG2
NUC1
//DDPLOG03 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1.NUC2 <=== PLOG1
NUC2
//DDPLOG04 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR2.NUC2 <=== PLOG2
NUC2
//DDPLOG05 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1.NUC3 <=== PLOG1
NUC3
//DDPLOG06 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR2.NUC3 <=== PLOG2
NUC3
//MERGIN1 DD DISP=SHR,DSN=EXAMPLE.PLOG.MERGIN1 <=== INTERMEDIATE 1
//MERGIN2 DD DISP=SHR,DSN=EXAMPLE.PLOG.MERGIN2 <=== INTERMEDIATE 2
//DDSIAUS1 DD DSN=EXAMPLE.DByyyyy.PLOG1(+1), <=== PLOG COPY
// VOL=SER=ADAxxx,UNIT=TAPE,DISP=(NEW,CATLG)
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//DDCARD DD *
ADARUN PROG=ADARES,MODE=MULTI,SVC=svc,DEVICE=3380,DBID=yyyyy
```



```

/*
//DDKARTE DD *
ADARES PLCOPY NOPPT
/*

```

ADARES MERGE CLOG - Merge Nucleus Cluster Command Logs

In an Adabas cluster environment, command logs (CLOGs) from the cluster nuclei may be manually merged using the ADARES MERGE CLOG NUMLOG=nn function.

The NUMLOG parameter is required: it specifies the number of command log datasets to be included in the merge process. The maximum number is 32.

Sequential datasets are expected as input to the MERGE CLOG function; therefore, the ADARES CLCOPY function (with ADARUN CLOGMRG=NO, the default) must be executed prior to the ADARES MERGE function.

The timestamp contained in the CLOGLAYOUT=5 format of the CLOG is required for the proper merging of command logs records.

ADARESCM Job

The following sample job ADARESCM (see the JOBS dataset) illustrates the execution of the ADARES MERGE CLOG function:

```

//ADARESCM JOB
//*
//* ADARES : MERGE SEQUENTIAL COMMAND LOGS
//* FOR USE WITH AN ADABAS NUCLEUS CLUSTER
//*
//RES EXEC PGM=ADARUN
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD <=== ADABAS LOAD
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDWORKR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.WORKR1 <=== WORK
//DDCLOG01 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.CLOGR1.NUC1 <=== CLOG1
NUC1
//DDCLOG02 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.CLOGR1.NUC2 <=== CLOG1
NUC2
//DDCLOG03 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.CLOGR2.NUC3 <=== CLOG2
NUC3
//DDSIAUS1 DD DSN=EXAMPLE.DByyyyy.CLOGM, <=== OUTPUT OF
// VOL=SER=ADAxxx,UNIT=TAPE,DISP=(NEW,CATLG) CLOG MERGE
//DDDRUCK DD SYSOUT=X
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X

```

```
//DDCARD DD *  
ADARUN  PROG=ADARES,MODE=MULTI,SVC=svc,DEVICE=3380,DBID=yyyyy  
/*  
//DDKARTE DD *  
ADARES MERGE CLOG,NUMLOG=3  
/*
```

ADARES BACKOUT and REGENERATE - Uniquely Identifying Checkpoints

After the protection log merge process, the block number will not necessarily be the same. To uniquely identify the checkpoint in this situation, it is necessary to also specify the `NUCID` for all ADARES functions that can specify a `TOBLK / FROMBLK` parameter; that is, `BACKOUT` and `REGENERATE`.



Notes:

1. `BACKOUT DPLOG` and `BACKOUT MPLOG` are not allowed for a cluster database. The `PLOG` must be merged before the `BACKOUT` can be performed.
2. The merge process ensures that there is at most one checkpoint per block. It records the (old) block number prior to the merge and the `NUCID` that wrote the checkpoint. When you then specify the block number and `NUCID` as reported in `ADAREP`, ADARES is able to uniquely identify the block.
3. In an Adabas nucleus cluster environment, `ADAREP` includes the `NUCID` when printing all checkpoint information.

The additional parameters that are required in an Adabas nucleus cluster environment are `NUCID`, `TONUCID`, `FROMNUCID`.

If the `NUCID` is the same for the starting and ending checkpoint, only the `NUCID` needs to be specified.



Note: An `ADAREP CPEXLIST` function can be used to determine the original block number and `NUCID` that wrote the checkpoint. This is the block number prior to the merge and the one that ADARES `REGENERATE` and `BACKOUT` expects.

ADASAV Processing Change

Sample JCL is located in the ADASAVRW member of the JOBS dataset.

For the following ADASAV functions:

RESTONL (database), RESTONL GCB; RESTORE (database), RESTORE GCB,

the Work datasets/files of all cluster (or noncluster) nuclei for the database that may have been active at the time of the ABEND must be reset. This can be done either

- manually (e.g., by using ADAFRM WORKRESET FROMRABN=1,SIZE=1B); or
- by specifying the Work datasets/files with DD names/link names DD/WORKRn (n=1-9) or DD/WORKnn (nn=10-32) in the JCL for the RESTONL/RESTORE function.

Otherwise, the nuclei that did not have their Work datasets/files reset will give parm-error 42 when started.

The DD/PLOGRn and DD/CLOGRn datasets are not reset in the restore process. They must be either copied/merged by ADARES PLCOPY/CLCOPY or reset by ADAFRM.

ADASAV RESTPLOG -- Uniquely Identifying Checkpoints

After the protection log merge process, the block number will not necessarily be the same. To uniquely identify the checkpoint in this situation, it is necessary to also specify the NUCID parameter for the ADASAV RESTPLOG function when specifying the SYN1 or SYN4 parameter.



Note: An ADAREP CPEXLIST function can be used to determine the original block number and NUCID that wrote the checkpoint. This is the block number prior to the merge and the one that ADASAV RESTPLOG expects.

8

Cluster Nucleus Session End Statistics

▪ General Nucleus Information	34
▪ Input/Output Statistics	34
▪ Command Statistics	35
▪ User Statistics	36
▪ Efficiency Statistics	36
▪ External Cache Statistics (Cluster Nucleus Only)	37
▪ External Lock Statistics (Cluster Nucleus Only)	42
▪ Dataset Activity Statistics	47

In addition to the end-of-session statistics printed by every Adabas nucleus, the statistics for a cluster nucleus also include external cache and lock statistics.

If you are running the selectable unit Adabas Online System (AOS), all of the statistics shown in the following sample output are displayed.

If you are running only the demo version of AOS delivered with Adabas, the statistics displayed are limited as follows:

Section	Displays statistics only for...
External Cache Statistics	totals, DS, and NI
External Lock Statistics	buffer flush, hold ISN, new data RABN, and global update command sync locks

This chapter covers the following topics:

General Nucleus Information

```
The  A d a b a s  nucleus session
Started 2001-02-13 22:58 and ended 2001-02-13
23:05
Duration      00000:06:59  hours
Wait-time     00000:02:26  hours
Cpu-time      00000:00:53  hours
```

Input/Output Statistics

I/O Counts (Including Initialization)

```
READS      WRITES
-----
ASSO          4710      6913
DATA          1750      2853
WORK           3       7251
PLOG           0         0
CLOG           0         0
-----
Total         6463      17017
```

Log Reads and Buffer Efficiency

Log. reads	173,393
Buffer eff.	26.8

Distribution of ASSO/DATA I/Os by Volser Number (Excluding Initialization)

Vol-ser	High RABN	Count
WRKM01 (ASSO	8082)	11599
WRKM01 (DATA	5990)	4603
TOTAL		16202

Command Statistics

Count of Calls Executed and Threads Used

A d a b a s	executed	10,249 calls
	in	8 threads

Distribution of Commands by Source

Source	Number
Remote commands	0
Local commands	10,102
Internal commands	144
Operator commands	3

Distribution of Commands by Thread

Thread	Number
1	2,657
2	1,803
3	1,401
4	1,300
5	1,193
6	977
7	917
8	1
Total	10,249

Distribution of Commands by File

File	Number
0	4,282
30	5,968
Total	10,250

Distribution of Commands by Type

Cmd-type	Number
A1/4	1,968
CL	44
ET	4,040
N1/2	2,000
OP	43
UC	7
REST	2,148
Total	10,250

User Statistics

There were 43 users participating
Most calls (303) initiated by user USADFMB2
Most I/O-s (331) initiated by user USADFMB2
Most thr.-time (00:00:08) was used by user USADFMB1

Efficiency Statistics

46 Formats had to be translated
0 Formats had to be overwritten
0 Autorestarts were done
0 Throw-backs due to ISN problem
0 Throw-backs due to space problem

143 Bufferflushes were done

Buffer Flush Information

Flush phases	212
Blocks flushed	28,503
Flush I/Os	8,756
Flush requests:	
Return immediately	52,658
Return after logical flush	0
Return after entire flush	15

Actual High-water Marks for Major Pools (Except the Bufferpool)

AREA	ADARUN	PARM	HIGH-WATER-MARK
AB -POOL	NAB=	2000	51712 (0 %)
CQ -POOL	NC =	96000	3840 (4 %)
DUQ -POOL	LDE=	5000	0 (0 %)
FI -POOL	LFP=	20000	6560 (32 %)
HQ -POOL	NH =	16856	588 (3 %)
SC -POOL	LCP=	10000	0 (0 %)
TBI -POOL	LI =	10000	0 (0 %)
TBS -POOL	LQ =	100000	0 (0 %)
UQ -POOL	NU =	500	8844 (6 %)
UQF -POOL	NU =	500	1512 (3 %)
WORK-POOL	LWP=	800000	114296 (14 %)
XID -POOL	XID=		(0 %)

External Cache Statistics (Cluster Nucleus Only)

Cast-out dir	:	188
Synchronous	:	188
Asynchronous	:	0
Unlock cast-out:		212
Synchronous	:	132
Asynchronous	:	80
Directory reads:		3
Synchronous	:	0
Asynchronous	:	3

Totals

Reads	:	15,006
Synchronous	:	15,006
Asynchronous	:	0
In cache	:	6,245
Not in cache	:	8,761
Structure full:	:	0
Writes	:	66,726
Synchronous	:	66,726
Asynchronous	:	0
Written	:	66,726
Not written	:	0
Structure full:	:	0
Validates	:	327,623
Block invalid	:	0
Cast-out reads	:	28,503
Synchronous	:	28,503
Asynchronous	:	0
Deletes	:	0
Timeouts	:	0

Address Converter (AC)

Reads	:	8
Synchronous	:	8
Asynchronous	:	0
In cache	:	0
Not in cache	:	8
Structure full:	:	0
Writes	:	2,004
Synchronous	:	2,004
Asynchronous	:	0
Written	:	2,004
Not written	:	0
Structure full:	:	0
Validates	:	5,983
Block invalid	:	0
Cast-out reads	:	72

Synchronous	:	72
Asynchronous	:	0
Deletes	:	0
Timeouts	:	0

Data Storage (DS)

Reads	:	2,775
Synchronous	:	2,775
Asynchronous	:	0
In cache	:	26
Not in cache	:	2,749
Structure full:	:	0
Writes	:	4,972
Synchronous	:	4,972
Asynchronous	:	0
Written	:	4,972
Not written	:	0
Structure full:	:	0
Validates	:	9,965
Block invalid	:	0
Cast-out reads	:	2,921
Synchronous	:	2,921
Asynchronous	:	0
Deletes	:	0
Timeouts	:	0

Data Storage Space Table (DSST)

Reads	:	2
Synchronous	:	2
Asynchronous	:	0
In cache	:	0
Not in cache	:	2
Structure full:	:	0
Writes	:	2,004
Synchronous	:	2,004
Asynchronous	:	0
Written	:	2,004
Not written	:	0

Cluster Nucleus Session End Statistics

Structure full:	0
Validates :	4,490
Block invalid :	0
Cast-out reads :	69
Synchronous :	69
Asynchronous :	0
Deletes :	0
Timeouts :	0

File Control Block (FCB)

Reads :	5
Synchronous :	5
Asynchronous :	0
In cache :	0
Not in cache :	5
Structure full:	0
Writes :	4,970
Synchronous :	4,970
Asynchronous :	0
Written :	4,970
Not written :	0
Structure full:	0
Validates :	56,029
Block invalid :	0
Cast-out reads :	119
Synchronous :	119
Asynchronous :	0
Deletes :	0
Timeouts :	0

Normal Index (NI)

Reads	:	12,057
Synchronous	:	12,057
Asynchronous	:	0
In cache	:	6,219
Not in cache	:	5,838
Structure full:		0
Writes	:	44,096
Synchronous	:	44,096
Asynchronous	:	0
Written	:	44,096
Not written	:	0
Structure full:		0
Validates	:	25,685
Block invalid	:	0
Cast-out reads	:	22,973
Synchronous	:	22,973
Asynchronous	:	0
Deletes	:	0
Timeouts	:	0

Upper Index (UI)

Reads	:	159
Synchronous	:	159
Asynchronous	:	0
In cache	:	0
Not in cache	:	159
Structure full:		0
Writes	:	8,680
Synchronous	:	8,680
Asynchronous	:	0
Written	:	8,680
Not written	:	0
Structure full:		0
Validates	:	225,471
Block invalid	:	0

Cluster Nucleus Session End Statistics

Cast-out reads :	2,349
Synchronous :	2,349
Asynchronous :	0
Deletes :	0
Timeouts :	0

File Statistics for Files with More than 25% of the Total Cache Statistics

File 30:	
Reads :	14,998
Writes :	64,710
Validates :	323,105

External Lock Statistics (Cluster Nucleus Only)

General Control Block (GCB) Lock

Obtains - Conditional :	0
Granted :	0
Rejected :	0
Unconditional :	0
Synchronous :	0
Asynchronous :	0
Releases - Issued :	0
Synchronous :	0
Asynchronous :	0

Security Lock

Obtains - Conditional :	0
Granted :	0
Rejected :	0
Unconditional :	0
Synchronous :	0
Asynchronous :	0
Releases - Issued :	0
Synchronous :	0
Asynchronous :	0

File Space Table (FST) Lock

Obtains - Conditional	:	1
Granted	:	1
Rejected	:	0
Unconditional	:	1
Synchronous	:	2
Asynchronous	:	0
Releases - Issued	:	2
Synchronous	:	2
Asynchronous	:	0

File Lock Table Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	5
Synchronous	:	5
Asynchronous	:	0
Releases - Issued	:	5
Synchronous	:	5
Asynchronous	:	0

Online Save Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	0
Synchronous	:	0
Asynchronous	:	0
Releases - Issued	:	0
Synchronous	:	0
Asynchronous	:	0

Buffer Flush Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	152
Synchronous	:	152
Asynchronous	:	0
Releases - Issued	:	152
Synchronous	:	152
Asynchronous	:	0

Global ET Sync Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	0
Synchronous	:	0
Asynchronous	:	0
Releases - Issued	:	0
Synchronous	:	0
Asynchronous	:	0

Recovery Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	0
Synchronous	:	0
Asynchronous	:	0
Releases - Issued	:	0
Synchronous	:	0
Asynchronous	:	0

Hold ISN Locks

Obtains - Conditional	:	3972
Granted	:	3972
Rejected	:	0
Unconditional	:	0
Synchronous	:	3972
Asynchronous	:	0
Releases - Issued	:	3972
Synchronous	:	3972
Asynchronous	:	0

Unique Descriptor Locks

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	0
Synchronous	:	0
Asynchronous	:	0
Releases - Issued	:	0
Synchronous	:	0
Asynchronous	:	0

ETID Locks

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	0
Synchronous	:	0
Asynchronous	:	0
Releases - Issued	:	0
Synchronous	:	0
Asynchronous	:	0

New Data RABN Locks

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	1000
Synchronous	:	1000
Asynchronous	:	0
Releases - Issued	:	1000
Synchronous	:	1000
Asynchronous	:	0

Checkpoint Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	4
Synchronous	:	4
Asynchronous	:	0
Releases - Issued	:	4
Synchronous	:	4
Asynchronous	:	0

ET Data Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	0
Synchronous	:	0
Asynchronous	:	0
Releases - Issued	:	0
Synchronous	:	0
Asynchronous	:	0

Global Update Command Sync Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	143
Synchronous	:	143
Asynchronous	:	0
Releases - Issued	:	143
Synchronous	:	143
Asynchronous	:	0

Parameter Lock

Obtains - Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	0
Synchronous	:	0
Asynchronous	:	0
Releases - Issued	:	0
Synchronous	:	0
Asynchronous	:	0

Dataset Activity Statistics

ADAI03 DDWORKR1	3 READS	7251 WRITES
ADAI03 DDDATAR1	1750 READS	2853 WRITES
ADAI03 DDASSOR1	4710 READS	6913 WRITES

9

Adabas Online System Cluster Environment Screens

▪ Display Cluster Members	50
▪ Nucleus File Status	51
▪ Nucleus Status Flags	52
▪ Cluster Usage	54
▪ Maintain the User Table	61

This chapter describes the Adabas Online System screens that apply to a cluster environment.

Display Cluster Members

From the Session Monitoring menu, a new function *Display cluster members* (option A) produces the following screen:

```

16:21:45          ***** A D A B A S  BASIC SERVICES *****          2002-07-19
DBID 105          - Display Cluster Members -          PACA002

Total number of nuclei in the cluster: 4
I Sel I Nuc ID I System ID I Jobname I Status I Available Services I
-----
I _ I 1 I DAEMVS I ADANUC01 I Active I All I
I _ I 2 I DAEMVS I ADANUC02 I Inactive I Lock I
I _ I 3 I DDZMVS I ADANUC03 I Active I All I
I _ I 4 I DDZMVS I ADANUC04 I Active I All I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I
I I I I I I I I I

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help                Exit          Refresh                Menu
  
```

The screen includes a list of nuclei participating in the cluster and information about the current status of each nucleus.

- ▶ **To select a nucleus for additional processing**
 - Type "S" in the Sel column opposite that nucleus.

- ▶ **To display additional information about a nucleus**
 - Type "D" in the Sel column opposite that nucleus.

For an Adabas cluster nucleus that has a nonzero nucleus ID, its entry in the parallel participant table (PPT) is displayed in a screen similar to the following:

```

16:21:45          ***** A D A B A S  BASIC SERVICES *****          2002-07-19
DBID 105          - Display PPT Entry -          PACA002

Nuc ID. . .      3 Active Nucleus

Name              Status                               DataSet Name
-----
WORK1
PLOGR1  Ready to be copied/merged          SAG.ADABAS.DB105.PLOGR1
PLOGR2  Being written by nucleus          SAG.ADABAS.DB105.PLOGR2

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help          Exit          Refresh          Menu
    
```

Nucleus File Status

From the Resource Utilization menu, the *Nucleus file status* (option N) has been added and is the equivalent of the DNFV operator command.

```

16:03:17          ***** A D A B A S  BASIC SERVICES *****          2002-05-29
DBID 1955          - Nucleus File Status -          PACUN02
NucID 1021

          Locking
File      NucID  Access count  Update count  State
-----
24          0          0          0          Access
25          0          0          0          Access, Update
    
```

```

Last page
PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help      Repos    Exit     Refresh  -         +         Menu
    
```

In an Adabas cluster environment, the file may be locked for exclusive use by another cluster nucleus. If this is the case and the file is in the nucleus file status table, the Locking NucID column for the file shows the ID of the nucleus that has exclusive control.

The Access count / Update count fields display the number of access or update users, respectively, that refer to the specified file in their user queue elements (UQEs). These users either have specified the file in an OP command with R-option or are using the file in an as yet incomplete transaction.

A State field indicates when the file is used for access only or for access and update. The State field indicates to what extent a nucleus can use a file on its own. If the requested use exceeds the given state, the nucleus must first communicate with the other nuclei in the cluster in order to upgrade the state.

Nucleus Status Flags

From the Resource Utilization menu, a second screen has been added to the *System status* (option S), which displays I/O counts for the ASSO, DATA, WORK, and PLOG data sets; remote and local call distribution; and other current session status information.

```

18:50:16          ***** A D A B A S  BASIC  SERVICES *****          2002-05-30
DBID 1955          - System Status -          PACUS02
NucID: 1022

          Physical
          Reads          Writes          Call Distribution
          -----
ASSO          370          67          Remote Logical .....          0
DATA          3          18          Remote Physical .....          0
WORK          2          104          Local Logical .....          860
PLOG          67          Local Physical .....          0

Logical Reads .....          349          Logical Reads (binary) .....          0000015D
Buffer Efficiency ....          0.9          No. of HQEs active .....          0
          No. of UQEs in User Queue ..          2
Format Translations ..          51          No. of CQEs waiting in CQ ..          0
Format Overwrites ....          0          Total intern. Autorestarts .          0
Throw Backs for ISN ..          0          No. of PLOG switches .....          0
Throw Backs for Space.          0          No. of Bufferflushes .....          18
          page 1 of 2
    
```



```
PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help                Exit      Refresh                +      Menu
```

Press PF8 to display an additional screen that indicates if one or more of the following are in progress:

- Online database save running;
- ADAEND in progress;
- Online file save running;
- READONLY/UTIONLY transition;
- READONLY status;
- Update processing suspended;
- ET-sync in progress;
- UTIONLY status; and
- Exclusive-DB-control utility running.

Otherwise, "Adabas operation normal" is displayed.

```
16:47:41          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
DBID 1955          -  System Status  -          PACUS02
NucID: 1021
```

```

                Nucleus Status Flags
                -----
                Adabas operation normal

```

page 2 of 2

```
PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help                Exit      Refresh                +      Menu
```

Cluster Usage

From the Resource Utilization menu, *Cluster usage* (option X) displays nucleus cluster statistics that are equivalent of those displayed using the DXCACHE, DXLOCK, and DXFILE operator commands.

The equivalent direct command is

`DISPLAY CLUSTERSTATUS`

```

16:10:31          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
                                     - Cluster Usage -                      PACUX02

                Code      Service
                ----      -
                C          Cache statistics
                F          File statistics
                L          Lock statistics
                ?          Help
                .          Exit
                ----      -

Code ..... _
File Number .. 0
Database ID .. 1955 (WIS1955)          NucID .. 1021

Command ==>
PF1----- PF2----- PF3----- PF4----- PF6----- PF10----- PF11----- PF12-----
Help          Exit          Fuse          Flist          Menu
    
```

This section covers the following topics:

- [Cache Statistics](#)
- [File Statistics](#)

- [Lock Statistics](#)

Cache Statistics

Choosing *cache statistics* (option C) from the Cluster Usage menu displays the following menu:

```

16:14:23          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
                   - Cache Statistics -                               PACUX12

      Code   Service
      ----   -
      K     Cast-out / Directory
      P     Publishing requests
      X     Individual cache blocks
      .     Exit
      ?     Help
      ----   -

Code .....
Database ID .. 1955   (WIS1955)                NucID .. 1021

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help          Exit      Refresh          Menu
    
```

The rest of this section describes each of the options on this screen.

- [Cast-out / Directory](#)
- [Publishing Requests](#)
- [All Cache Blocks](#)

Cast-out / Directory

Choosing *cast-out / directory* (option K) from the Cache Statistics menu display the following:

```

16:14:23          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
DBID 1955          - Cast-out / Directory -          PACUX12
NucID 1021

      Cast-out Directory Reads          Directory Reads
      -----
Total .....          28          Total .....          5
  Sync .....          1          Sync .....          1
  Async ....          27          Async ....          4

      Unlock Cast-out Calls
      -----
Total .....          28
  Sync .....          1
  Async ....          27

PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help                Exit      Refresh                Detail      Menu
    
```

Counters have a multiplier column with the following values:

Value	The total shown is in ...
blank	(factor of 1)
K	kilo (factor of 1,000)
M	mega (factor of 1,000,000)
G	giga (factor of 1,000,000,000)

If a number has a multiplier shown, it has been divided by the multiplier, showing the significant digits to 9 places with no decimal point.

Press PF9 to see the entire value. This value is the exact count up to 20 digits in length.

Publishing Requests

Choosing *publishing requests* (option P) from the Cache Statistics menu display the following:

```

16:26:21          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
DBID 1955          - Publishing Requests -          PACUX12
NucID 1021

          Publishing Request Category
          -----
          Update sync .....          34
          BT or CL or ET ....          162
          Redo threshold ....          2
          Full bufferpool ...          0
          All blocks .....          84
          Specific RABN .....          0
          File DS blocks ....          4

PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help          Exit          Refresh          Detail          Menu
    
```

All Cache Blocks

Choosing *all cache blocks* (option X) from the Cache Statistics menu display the following:

```

16:27:05          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
DBID 1955          - All Cache Blocks -          PACUX12
NucID 1021

          Reads                               Writes
          -----                               -----
          Total .....          167          Total .....          38,176
          Sync .....          24          Sync .....          15,148
          Async .....          143          Async .....          23,028

          In cache .....          49          Written .....          38,176
          Not in cache ..          118          Not written .....          0
          Struc. full ...          0          Struc. full .....          0

          Cast-out Reads                       Other
          -----                               -----
          Total .....          212          Validates .....          187,677
          Sync .....          212          Invalid .....          43
          Async .....          0          Deletes .....          0
    
```

					Timeouts		0
					Redo processes		0
PF1-----	PF2-----	PF3-----	PF4-----	PF7-----	PF8-----	PF9-----	PF12-----
Help	Repos	Exit	Refresh	PrevBlk	NxtBlk	Detail	Menu

Use PF7 and PF8 to scroll through the cache blocks; use PF2 to reposition.

Statistics are displayed for the following:

- All cache blocks
- Address converter (AC) cache blocks
- Data Storage (DS) cache blocks
- Data Storage space table (DSST) cache blocks
- File control block (FCB) cache blocks
- Normal index (NI) cache blocks
- Upper index (UI) cache blocks

Press PF9 from the above screen to display the following detail screen:

```

16:27:05          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
DBID 1955          - All Cache Blocks -          PACUX12
NucID 1021
Reads              Writes
-----
Total .....      167 Total .....      38,176
  Sync .....      24  Sync .....      15,148
  Async .....     143  Async .....      23,028

  In cache..       49  Written ...      38,176
  Not in ...      118  Not writ ..       0
  Stru.full.       0   Stru.full .       0

Cast-out Reads    Other
-----
Total .....      212 Validates ...      187,677
  Sync .....      212  Invalid ...        43
  Async .....       0  Deletes .....       0
                               Timeouts ..         0
                               Redo procs ..        0

                                Press Enter to continue
    
```

File Statistics

Choosing *file statistics* (option F) from the Cluster Usage menu for file 25 displays the following menu:

```

16:37:02          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
DBID 1955          - File 25 Statistics -          PACUX22
NucID 1021
  Reads
  -----
  Total .....          67          Writes
  Sync .....          0          Total .....          20,157
  Async .....         67          Sync .....          7,583
                                     Async .....         12,574
  In cache .....          0          Written .....         20,157
  Not in cache ..         67          Not written .....          0
  Struc. full ...          0          Struc. full .....          0
  Cast-out Reads
  -----
  Total .....          78          Other
  Sync .....          78          Validates .....         79,248
  Async .....          0          Invalid .....          0
                                     Deletes .....          0
                                     Timeouts .....          0
                                     Redo processes .....          0
PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help   Repos   Exit   Refresh                Detail   Menu
    
```

Lock Statistics

Choosing *lock statistics* (option L) from the Cluster Usage menu displays the following menu:

```

16:38:16          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
                                     - Lock Statistics -          PACUX32
Code  Service
-----
A    Buffer flush lock
B    Checkpoint lock
C    DSF lock
D    ETID lock
E    File-lock-table lock
F    FST lock
G    GCB lock
H    Global ET sync lock
.    Exit
?    Help
-----
Code  Service
-----
I    Global update command sync lock
J    Hold ISN lock
K    New-Data-RABN lock
L    Online save lock
M    Parameter lock
N    Recovery lock
O    RLOG lock
P    Security lock
Q    Spats lock
R    Unique descriptor lock
-----
    
```

```

Code ..... _
Database ID .. 1955   (WIS1955)           NucID .. 1021

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help          Exit       Refresh          Menu
    
```

Each of the options on the Lock Statistics menu displays statistics for a particular lock. For each lock, the screen displays obtain and release information about the various types of that lock that are currently in use by a cluster nucleus:

- The system may obtain locks conditionally or unconditionally, synchronously or asynchronously. A conditional request for a lock may be granted or rejected.
- Releases may be performed synchronously or asynchronously.

Hold ISN Lock

Choosing *hold ISN lock* (option J) from the Lock Statistics menu displays the following:

```

16:38:16          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
DBID 1955          - Hold ISN Lock -          PACUX32
NucID 1021

      Obtains                               Releases
      -----                               -----
      Conditional ....          16,017      Issued .....          16,017
      Granted .....          16,017      Sync .....          15,971
      Rejected ....           0          Async .....           46
      Unconditional ..           0

      Sync .....           158
      Async .....          15,859

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help      Repos    Exit       Refresh          PrevLok  NxtLok  Menu
    
```

Use PF7 and PF8 to scroll through the locks; use PF2 to reposition.

Maintain the User Table



Note: This option is available in Adabas nucleus cluster environments only.

A new function has been added to the Session Opercoms menu to support the CLUFREEUSER command. When option V (maintain user table) is selected, the following screen is displayed:

```

16:59:29          ***** A D A B A S  BASIC  SERVICES *****          2002-05-29
                   - User Table Maintenance -                          PACIV02

                   Code      Service
                   ----      -
                   C         Begin CLUFREEUSER process
                   ?         Help
                   .         Exit
                   ----      -

Code ..... _
TNA ..... 0_____
UID ..... _____
Force ..... _
Global ..... _

Database ID .. 1955 (WIS1955)          NucID .. 1022

Command ==>
PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help           Exit           Menu
    
```

The CLUFREEUSER command is only valid in cluster environments. It can be issued against the local nucleus only or, with the Global option, against all active and inactive nuclei in the cluster.

The command is used to delete leftover user table elements (UTEs) in common storage that are no longer associated with user queue elements (UQEs) in a nucleus where

TNA	is a decimal number specifying the timeout value in seconds. UTEs that are not used during the time specified may be deleted if other conditions are fulfilled. If TNA is not specified, UTEs may be deleted without regard to their recent use.	
UID	is a character string or hexadecimal byte string as follows:	
	ccccccc	where the argument is 1-8 letters, digits, or embedded '-' signs without surrounding apostrophes.

	'ccccccc'	where the argument is 1-8 characters with surrounding apostrophes.
	X'xxxxxxxxxxxxxxxx'	where the argument is an even number of 2-16 hexadecimal digits enclosed by 'X'.
	<p>A character string must be enclosed in apostrophes if it contains characters other than letter, digits, or embedded '-' signs. If a specified character string is less than 8 characters long, it is implicitly padded with blanks. If a specified hexadecimal string is shorter than 16 hexadecimal digits, it is implicitly padded with binary zeros.</p> <p>If the last 8 bytes of a user's 28-byte communication ID match a specific user ID or user ID prefix, that user's UTE may be deleted if other conditions are fulfilled.</p> <p>If UID not specified, UTEs may be deleted regardless of their user IDs.</p>	
FORCE	Delete leftover UTEs even if the users are due a response code 9, subcode 20. If FORCE is not specified, such UTEs are not deleted. Before using the FORCE parameter, ensure that the users owning the UTEs to be deleted will not expect any of their transactions to remain open. Specify FORCE on this screen by marking the Force field with any character.	
GLOBAL	Delete leftover UTEs throughout the Adabas cluster if they are no longer associated with UQEs and are eligible according to the other specified parameters. Additionally and subject to the other rules, delete leftover UTEs if their assigned nuclei have terminated since their last use. If GLOBAL is not specified, only UTEs assigned to the local nucleus and used since the nucleus start are eligible for deletion. Specify GLOBAL on this screen by marking the Global field with any character.	

Index

A

ADACOM
starting, 4

C

Cluster
display usage statistics
using Basic Services, 54
COMPRINT
messages written to, 4

D

Downtime
planning, 19

E

Entire Net-Work
abnormal termination, 11
starting, 4

M

Maintenance
preventive, 19

N

NUCID
ADADBS utility OPERCOM parameter
for command routing, 22
Nucleus
migrating to another OS/390 image, 19
routing utility-issued operator commands, 22
serialization during initialization, 5
starting, 4
Nucleus file
display status
using Basic Services, 51

O

Outage
planning, 19

P

Parameters
ADADBS OPERCOM
NUCID, 22

S

Serialization
of nucleus, 5
Session
display status
using Basic Services, 52
System
display status
using Basic Services, 52

U

Utilities
ADADBS OPERCOM
NUCID parameter, 22
nucleus cluster processing, 22
resynchronizing operations of, 22

