

# Utility Processing

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
  - ADADBS REFRESHSTATS - Refresh Statistical Values
  - ADAICK PPTPRINT - Print/Dump Parallel Participant Table
  - ADAPLP IPLOGPRI - Print Sequential Intermediate Data Sets
  - ADARAI - Adabas Recovery Aid
  - ADAREP - Checkpoint Information Extended
  - ADARES CLCOPY - Copy/Merge Nucleus Cluster Command Logs
  - ADARES PLCOPY - Copy/Merge Nucleus Cluster Protection Logs
  - ADARES PLCOPY NOPPT - Ignore PPT
  - ADARES MERGE CLOG - Merge Nucleus Cluster Command Logs
  - ADARES BACKOUT and REGENERATE - Uniquely Identifying Checkpoints
  - ADASAV Processing Change
  - ADASAV RESTPLOG -- Uniquely Identifying Checkpoints
- 

## 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 PPTPRINT - Print/Dump Parallel Participant Table

The PPTPRINT 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 Data Sets**

The IPLOGPRI function is used to print the sequential intermediate data sets created from the PLOG merge process. Input to ADAPLP IPLOGPRI must be a MERGIN1/MERGIN2 data set 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 data sets, 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 data sets MERGIN1 and MERGIN2 required for automated CLOG copy/merge processing in nucleus cluster environments.

### **Notes:**

1. When intermediate data sets are used for both CLCOPY and PLCOPY, the data set names must be unique so that they are not overwritten.
2. The data set 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 data sets
//*
//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 data sets using dynamic allocation. PLCOPY copies/merges as much data as possible; if a nucleus is still writing to a protection log data set, PLCOPY 'partially' merges the data set.

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

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

### Notes:

1. When intermediate data sets are used for both CLCOPY and PLCOPY, the data set names must be unique so that they are not overwritten.
2. The data set 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 data sets
/*
//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 data set.

A sample job ADARESPM is provided on the MVSJOBS data set to illustrate the manual execution of the PLCOPY merge function. Two intermediate data sets must be supplied. ADARES analyzes the data sets to determine which is to be used as input and which for output. Specific crosschecks determine whether the correct intermediate data set 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 data sets have been supplied on the session startup 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
//MERGIN1 DD DISP=SHR,DSN=EXAMPLE.PLOG.MERGIN1
//MERGIN2 DD DISP=SHR,DSN=EXAMPLE.PLOG.MERGIN2
//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 data sets of all cluster nuclei are being supplied with DD names DDPLOGnn in the JCL.



### Warning:

**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 data set; 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.



### Warning:

**Without the PPT, ADARES cannot perform any extensive validations on the input data sets.**

## 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
/*
//DDASS0R1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.ASS0R1 <=== ASSO
//DDDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.DATAR1 <=== DATA
//DDPLOG01 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR1.NUC1 <=== PLOG1
NUC1
//DDPLOG02 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR2.NUC1 <=== PLOG2
NUC1
//DDPLOG03 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR1.NUC2 <=== PLOG1
NUC2
//DDPLOG04 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR2.NUC2 <=== PLOG2
NUC2
//DDPLOG05 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.PLOGR1.NUC3 <=== PLOG1
NUC3
//DDPLOG06 DD DISP=SHR,DSN=EXAMPLE.DByyyyyy.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.DByyyyyy.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 data sets to be included in the merge process. The maximum number is 32.

Sequential data sets 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 data set) 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
/*
//DDASSO1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSO1 <=== 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 data set.

For the following ADASAV functions:

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

the Work data sets/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 data sets/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 data sets/files reset will give parm-error 42 when started.

The DD/PLOGRn and DD/CLOGRn data sets 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.