

Changes Since Adabas Cluster Services Version 7.2.2

This section provides information on changes to Adabas Cluster Services since version 7.2.2.

The chapter covers the following topics:

- Online Invert Support
 - Enforcement of Correct DSST Entries
 - Changed ADARUN Parameters
 - User Table Cleanup
 - New Parameter for ADARES PLCOPY NOPPT
 - Limited Support for Distributed Transactions
 - PLOG/CLOG Messages Changes
 - New ADAPLP Function
 - Triggers and Stored Procedures Facility Support
 - Support for System-Managed Processes
 - LOCAL Parameter Change
 - Cluster Nuclei
 - Online Recovery
 - New Formula for Lock Structure Size Estimation
-

Online Invert Support

Adabas Cluster Services 7.4 now supports the online invert function. Support of online reorder will be added in a subsequent release or version.

Enforcement of Correct DSST Entries

When deferred publishing of updates to the coupling facility cache structure is in effect (the default), Cluster Services 7.2.2 might produce incorrect DSST entries. Cluster Services 7.4.2 now contains logic for enforcing correct DSST entries.

Changed ADARUN Parameters

In order to better integrate with the new add-on product Adabas Parallel Services, which features multiple update nuclei running in the same operating system image and working on the same database, the ADARUN parameters used for invoking Cluster Services have been changed as follows:

Old Parameter Name	New Parameter Name
PLXID	NUCID
PLXCACHENAME	CLUCACHENAME
PLXGROUPNAME	CLUGROUPNAME
PLXLOCKNAME	CLULOCKNAME

Furthermore, there is the new parameter `CLUSTER` which specifies what kind of Adabas nucleus is to be run:

CLUSTER Parameter	Meaning
NO (default)	single, noncluster nucleus
LOCAL	Parallel Services nucleus
SYSPLEX	Cluster Services nucleus

For compatibility, Cluster Services 7.4 still supports the old `PLXID` and `PLXyyyyyNAME` parameters. Software AG recommends that you change them and use the `CLUSTER`, `NUCID` and `CLUyyyyyNAME` parameters instead.

The DD-names that the ADARES utility uses for the intermediate merge datasets in the automated PLOG and CLOG merge process have been changed:

Old DD-name	New DD-name
MERGIN(T)I	MERGIN1
MERGIN(T)O	MERGIN2

First, the DD-names have been made one character shorter (zap AU722069), so that they can be used as symbolic names for Parallel Services on VSE. Second, the implied reference to one intermediate dataset being for 'input' and the other being for 'output' has been removed, since ADARES determines by itself which one is to be read and which one to be written. There is no need to switch the DD-names between consecutive PLCOPY runs.

User Table Cleanup

The new operator command and ADADBS OPERCOM and AOS function `CLUFREEUSER` is available for situations where leftover User Table Elements in common storage need to be cleaned up. This function deletes such User Table Elements according to specified criteria.

By default, the cluster nucleus receiving this function deletes all User Table Elements assigned to itself which it does not know of (i.e., for which it has no User Queue Elements). If the GLOBAL option is specified, all User Table Elements are deleted that are assigned to nuclei that are no longer active, or that have no associated User Queue Elements in their assigned nuclei.

For more information about the User Table Cleanup, see the section *Maintain the User Table*.

New Parameter for ADARES PLCOPY NOPPT

The new parameter SBLKNUM has been introduced for the ADARES PLCOPY NOPPT function. This function is for use in an emergency case when a last set of PLOGs must be copied and merged after the parallel participant table (PPT) in the Associator has been destroyed. In this case the ADARES utility does not know the last serial block number of the sequential PLOGs produced so far. By default, it starts numbering the blocks at 1 on the next sequential PLOG dataset. If this sequential PLOG is concatenated to its predecessor in a REGENERATE function, ADARES would report an error in the PLOG block number sequence. The SBLKNUM parameter can be used to specify the correct starting block number for the sequential PLOG produced by the PLCOPY NOPPT function.

There are two ways to copy and merge the PLOGs when the PPT has been destroyed:

1. Without using the SBLKNUM parameter.
 - Run ADARES PLCOPY NOPPT, specifying all direct-access PLOG datasets of all cluster nuclei as DD/PLOGnn datasets for the ADARES utility. ADARES copies and merges the relevant PLOGs and produces a sequential PLOG dataset starting with PLOG block number 1.
 - In case the database or individual files need to be restored and regenerated, run the regenerate in two (or more) job steps. The first REGENERATE should cover all sequential PLOGs produced by PLCOPY functions without the NOPPT parameter; these PLOGs may be concatenated into one sequential input. If the REGENERATE is performed with ET logic (REGENERATE database or with CONTINUE parameter), specify the NOAUTOBACKOUT parameter. The second REGENERATE should cover the sequential PLOG produced by the PLCOPY with NOPPT.
2. With using the SBLKNUM parameter:
 - Determine the highest PLOG block number written so far by the regular PLOG merge processes. This number can be determined from an up-to-date checkpoint listing, from an ADARAI LIST protocol, or from the job protocol of the most recent PLCOPY execution.
 - Run ADARES PLCOPY NOPPT, specifying all direct-access PLOG datasets of all cluster nuclei as DD/PLOGnn datasets for the ADARES utility. Specify the SBLKNUM parameter with value $n+1$, where n is the highest PLOG block number written so far, as determined in the previous step. ADARES copies and merges the relevant PLOGs and produces a sequential PLOG dataset starting with the specified PLOG block number.
 - In case the database or individual files need to be restored and regenerated, no special arrangements need to be made to run the REGENERATE in separate job steps.

Limited Support for Distributed Transactions

Adabas Cluster Services 7.4.2 can in a limited way participate in distributed Adabas transactions coordinated by the Adabas Transaction Manager, as follows:

- In each distributed transaction, a single cluster database is allowed to participate. Different cluster databases can participate in different distributed transactions.
- All nuclei of a cluster database participating in distributed transactions must be run with the new ADARUN parameter setting `DTP=ET`.
- Cluster databases can participate in distributed transactions only if all resource managers involved in these transactions are Adabas databases running with `DTP=RM`. `DTP=ET` cannot be supported when an external transaction framework is included in the transaction (CICS/RMI or RRMS).

Full support by cluster databases for distributed transactions will be provided in a subsequent release of Adabas Cluster Services.

PLOG/CLOG Messages Changes

The messages in the PLOG/CLOG merge protocol of the ADARES PLCOPY/CLCOPY functions have been revised to be more comprehensive and meaningful (zaps AU722055 and AU722069).

New ADAPLP Function

The new ADAPLP function IPLOGPRI prints out the contents of an intermediate dataset (MERGINI or MERGINO) in the PLOG merge process (zap AU722068). The dataset is specified in the JCL via DD-name/link name DD/PLOG. The output is similar to that of the SPLOGPRI function.

Triggers and Stored Procedures Facility Support

The triggers and stored procedures facility (SPT) is now fully supported in a sysplex cluster environment:

- The ADARUN SPT parameter is now global and must be set the same on all active nuclei.
- When a REFRESH is executed, the trigger table is passed to all active nuclei.
- Only the first nucleus to initialize reads the trigger file to create the trigger table; all subsequent nuclei obtain the trigger table from one of the already active nuclei.

If you choose to use triggers and stored procedures in a cluster environment, you must set the Error Action field in the SPT profile under Adabas Online System to "halt" or "reject", but *not* "ignore". Note that in "reject" mode, the way a command is handled across two nuclei can be different if the command has a trigger associated with it.

Support for System-Managed Processes

Adabas Cluster Services 7.4 supports certain processes that are managed by the operating system. These processes are supported for the cache and lock structures used by Adabas Cluster Services.

Two types of system-managed processes are supported: *rebuild processes* and *duplexing rebuild processes*. See the following IBM documentation for additional information:

- MVS Setting Up a Sysplex
- MVS Programming: Sysplex Services Guide
- MVS System Commands (the SETXCF operator command)

The minimum system requirements for support of system-managed processes is z/OS version 1, release 2 or above, and a coupling facility with CF level 12.

System-managed rebuild dynamically rebuilds a coupling facility structure in the same or another coupling facility, while an active application (in this case, Adabas Cluster Services) may be using the structure. It can be used to reduce planned outage time for applications using the coupling facility.

System-managed duplexing rebuild dynamically duplicates a coupling facility structure into another coupling facility, while an application may be using the structure. It can be used to reduce unplanned outage time for applications using the coupling facility.

The options of the SETXCF operator command initiating these system-managed processes are SETXCF START,REBUILD and SETXCF START,REBUILD,DUPLEX, respectively.

LOCAL Parameter Change

The meaning of the LOCAL parameter has been adapted to that of a non-cluster database. It specifies whether or not the database can be accessed via Net-Work from other systems. If LOCAL=NO, the cluster database can be accessed from all systems to which a Net-Work connection exists. If LOCAL=YES, all cluster nuclei must run on the same system image and the database can be accessed only from that system.

Note:

For a cluster database, the LOCAL parameter applies to the common DBID target established by the cluster nuclei. In contrast, the FORCE parameter (force overwrite of ID Table Entry) applies to the NUCID parameter of each individual nucleus.

Cluster Nuclei

Cluster nuclei now perform an automatic check during session start to verify that resource locks via Global Resource Serialization (GRS) are effective against one another. If GRS has been incorrectly configured such that an ENQ performed by a starting nucleus has no effect against one of the already active nuclei, the starting nucleus will terminate with parm-error 106.

Online Recovery

At the beginning of online recovery after a peer nucleus failure, the nuclei participating in the recovery process will now in most cases print out the NUCID of the nucleus performing the autorestart.

New Formula for Lock Structure Size Estimation

The minimum lock structure size can be roughly estimated as

$$(NU*2 + NH + LDEUQP/16 + MAXFILES*4 + 50) * 240 + 500,000 \text{ bytes}$$

where 'MAXFILES' is the maximum number of files in the database (set in ADADEF or ADAORD) and NU, NH and LDEUQP are the ADARUN parameters of the cluster nuclei. The term in parenthesis (NU*2 + NH + LDEUQP/16 + MAXFILES*4 + 50) is the minimum number of lock records that the cluster nuclei expect to have available.

Further help is available at IBM's Coupling Facility Structure Sizer website at <http://www-1.ibm.com/servers/eserver/zseries/cfsizer>:

1. Read the text at the 'DISCLAIMER' link.
2. Click on 'OEM LOCK' link in the left panel.
3. Fill in the template:
 - Set the 'Number of Lock Entries' to a power of 2 (32,768 or higher).
 - Set the 'Number of Systems Sharing the Lock Structure' to the number of system images where cluster nuclei are to run.
 - Set the 'AAI' field to the number in parenthesis above, that is:

$$NU*2 + NH + LDEUQP/16 + MAXFILES*4 + 50$$
 - Check the 'OEM Lock Structure' box.
4. Click on the 'Submit' button.
5. The CFSIZER tool then proposes a lock structure size.

Note:

This link and procedure were valid in December 2002. Software AG cannot guarantee the future availability of this website, nor the accuracy of its results.

Note that the provided formula and procedure yield a minimum lock structure size. If it turns out that the lock structure of an Adabas cluster incurs a lot of 'false contention' (i.e., the appearance of resource conflicts which do not really exist), then it may be advisable to increase the lock structure in order to reduce false contention and thus improve performance.