Parallel Participant Table (PPT)

The parallel participant table (PPT), located in a database's Associator, tracks all active Adabas nuclei in the cluster. As each nucleus becomes active, it registers itself in the PPT. Each PPT entry can be modified by the nucleus or by ADARES.

This chapter covers the following topics:

- Acquiring the PPT

- Registering Nuclei during Session Initialization

- Registering PLOG, CLOG, and Work Data Sets for Each Nucleus

- Nucleus Termination Processing

- Using PPT Information

- PPT Unavailable

- SBLKNUM Parameter for ADARES PLCOPY NOPPT

Acquiring the PPT

The PPT is an area of 32 contiguous blocks that the ADADEF utility allocates in the Associator when it defines a new database.

Registering Nuclei during Session Initialization

During session initialization as each Adabas cluster nucleus becomes active, it registers in the PPT based on the setting of its ADARUN NUCID parameter. The nucleus chooses the first of the 32 PPT blocks that is not yet assigned to another NUCID. If this is not the first run of this nucleus, the nucleus reuses its previously used PPT block. The relative block number of the chosen PPT block within the PPT (a number between 1 and 32) is called the *internal nucleus ID*.

Note:
A noncluster nucleus always occupies the first block of the PPT and its NUCID is always zero.

Registering PLOG, CLOG, and Work Data Sets for Each Nucleus

Each active nucleus writes information to the Work, protection log (PLOG), and command log (CLOG) data sets that it is using for that session. The internal nucleus ID is also written to the PLOG records.

As each Work, PLOG, and CLOG data set becomes active, it is registered in the PPT. Information about these data sets is logged in the PPT entry of its nucleus.

When the PLOG or CLOG is merged, its PPT entry is updated with the information. ADARES and the cluster nuclei synchronize their PPT updates against one another to ensure that only one modification occurs at a time.

Work data set, protection log, and command log information remains in the PPT, even when CLOGs or PLOGs are merged or the session terminates.

Nucleus Termination Processing

If the nucleus terminates normally, the PPT entry is retained and marked as inactive. If a nucleus terminates abnormally, its entry in the PPT remains unchanged until the recovery process is complete.

Using PPT Information

The information about each Adabas nucleus maintained in the PPT is used when ADARES PLCOPY is copying and merging all active protection logs in the cluster.

It is also used during nucleus initialization to ensure that the data set/file information required for autorestart and PLOG merge is available:

- If the previously used Work data set or file and PLOG data sets or files are already in use by another nucleus, the nucleus will not start.

- If a different Work data set or file is provided and the original Work data set or file contains autorestart information, the nucleus will start but a warning will be printed.

- If different PLOGs or no PLOGs are provided and the previous session had PLOGs still to be copied, the nucleus will not start until those PLOGs have been copied/merged if PLOGRQ=FORCE is specified. If PLOGRQ=FORCE is not specified, the nucleus will start but a warning will be printed.

PPT Unavailable

In the case where the PPT becomes unavailable for some reason, the new parameter NOPPT is added to the ADARES PLCOPY function for emergency use. It specifies that the PPT is to be ignored and that the PLOG data sets of all cluster nuclei are to be taken from the JCL.

If you need to merge CLOGs while the PPT is unavailable, run a normal CLCOPY and then use the CLOG MERGE function to merge the sequential data sets.

SBLKNUM Parameter for ADARES PLCOPY NOPPT

The parameter SBLKNUM for the ADARES PLCOPY NOPPT function can be used 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 data set. 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 data sets of all cluster nuclei as DD/PLOGnn data sets for the ADARES utility. ADARES copies and merges the relevant PLOGs and produces a sequential PLOG data set 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. 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 data sets of all cluster nuclei as DD/PLOGnn data sets 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 data set 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.