

# Initialization

This chapter covers the following topics related to Adabas Parallel Services initialization:

- Sequence
  - ADACOM Initialization Process
  - Adabas Parallel Services Cluster Nucleus Process
  - PPT Processing
- 

## Sequence

Due to the interdependence among ADACOM(s) and Adabas cluster nuclei, these programs have certain co-occurrence requirements.

- ADACOM must be running on the operating system before any nucleus in a cluster that it manages is started.
- A nucleus will not start if another nucleus with the same NUCID is already active under the same router (SVC or IDTNAME).

## ADACOM Initialization 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 Router ID / DBID set and are written to the output data set/file specifically allocated for that set; starting with *PLI050*, the messages apply to ADACOM in general and are written to the COMPRINT data set.

### Note:

Prior to *PLI050*, the system default ADARUN parameter values for *MODE*, *DBID*, *DEVICE*, *SVC* or *IDTNAME*, 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
```

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

## Adabas Parallel Services Cluster Nucleus Process

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

```

PLX050 00006 ADACLU init  DBID=00006  NUCID=06001
PLX003 00006 There are nucs/ADACOMs active
PLX006 00006 Max users for image 00000200
PLX006 00006 PLXCB is located at 1127E258
ADAN03 00006 Initializing NUCID=6001  INTNUCID=4
ADAX2I 00006 SS-2,memstate, id 06001 ind 00 ext 00000      03
ADAX2A 00006 TI-0, initialised, rc 00000004
PLX087 00006 Attempting to acquire dataspace
PLX097 00006 Dataspace acquired
PLX059 00006 Initialization of ADACLU complete
ADAL01 00006 2002-06-25 17:50:45 CLOG not active
ADAN03 00006 ADABAS coming up
ADAX31 00006 Opening Work data set for NUCID=6001
ADAX70 00006 Connected to lock structure
ADAX70 00006 Number of lock entries          39,608
ADAX70 00006 Max number of record elements  39,608
ADAX57 00006 Connected to cache structure
ADAX57 00006 Directory elements             60,493
ADAX57 00006 Data elements                  15,123
ADAX57 00006 Data element size             1,024
ADAN19 00006 Buffer flush is asynchronous
ADAN01 00006 ADABAS Vv.r.s is active
ADAN01 00006 MODE = MULTI
ADAN01 00006 Running without recovery log

```

Access to the global lock area is established and acknowledged in the following message:

```
ADAX70 00006 Connected to lock structure
```

This message is followed by lock area statistics:

```

ADAX70 00006 Number of lock entries          32,768
ADAX70 00006 Max number of record elements  11,184

```

Access to the global cache area is established and acknowledged in the following message:

```
ADAX57 00006 Connected to cache structure
```

If the ADARUN parameter CLUCACHETYPE=V64 was specified, this message appears:

```
ADAX57 00006 Connecting to S64 cacache at address
```

This message is followed by cache area statistics:

```

ADAX57 00006 Directory elements             11,490
ADAX57 00006 Data elements                  2,872
ADAX57 00006 Data element size             1,024
ADAN19 00006 Buffer flush is asynchronous
ADAN01 00006 ADABAS Vv.r.s is active
ADAN01 00006 Mode = MULTI
ADAN01 00006 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 data sets are read and, if necessary, a user exit is invoked (user exit 2 for dual logs or user exit 12 for multiple logs).

This section covers the following topics:

- First Cluster Nucleus Starts
- Subsequent Cluster Nucleus Starts
- Noncluster Nucleus Starts
- Different PLOG Detected

## First Cluster Nucleus Starts

The Adabas Parallel Services 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 for dual logs or user exit 12 for multiple logs) 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 Cluster Nucleus Starts

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

## Noncluster Nucleus Starts

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

If PLOGs from a previous Adabas Parallel Services 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 (user exit 2 for dual logs or user exit 12 for multiple logs) controls the copy/merge process. If there is no user exit 2 or 12, the PLOG and PPT entry are overwritten.

## Different PLOG Detected

If PLOGRQ=FORCE 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.