

# Installation

It is important to review the entire installation procedure before starting the physical installation. Depending on which components you actually install, it may be useful to group certain installation activities together, even though they may not be in the same installation section.

Read the current version of the *Adabas Parallel Services Version 8.1 Release Notes* for specific information about late changes to this document.

The installation is described under the following headings:

- Prerequisites
- Using System Maintenance Aid
- z/OS Systems Installation
- Modifying the z/OS Common Storage (CSA) Key
- z/VSE Systems Installation
- BS2000/OSD Systems Installation

Notation *vrs* or *vr*: When used in this documentation, the notation *vrs* or *vr* stands for the relevant version, release, and system maintenance level numbers. For further information on product versions, see *version* in the *Glossary*.

---

## Prerequisites

This document covers the following topics:

- Required Operating Environment
- Prerequisite Software
- Adabas Parallel Services Router

## Required Operating Environment

Adabas Parallel Services can be installed on z/OS, z/VSE, and BS2000/OSD operating systems.

Before attempting to install Adabas Parallel Services, ensure that the host operating system is at the minimum required level. For information on the platform versions supported by Software AG products, access the Software AG web site at

<http://www.softwareag.com/corporate/products/bis/platforms/default.asp>.

To determine the particular platforms supported by this release of Adabas Parallel Services, review the product version availability information, available on Software AG's Empower web site:

<https://empower.softwareag.com>. Log into Empower. Once you have logged in, you can expand **Products** in the left menu of the web page and select **Product Version Availability** to access the Product Version Availability application. This application allows you to review support information for specific products

and releases.

Software AG provides support for the operating system versions supported by their respective manufacturers. Generally, when an operating system provider stops supporting a version of an operating system, Software AG will stop supporting that operating system version.

Although it may be technically possible to run a new version of Adabas or Adabas Parallel Services on an old operating system, Software AG cannot continue to support operating system versions that are no longer supported by the system's provider.

If you have questions about support, or if you plan to install Adabas Parallel Services on a release, version, or type of operating system other than those included at the link above, consult Software AG technical support to determine whether support is possible, and under what circumstances.

## Prerequisite Software

Adabas Parallel Services version 8.1.3 requires Adabas version 8.1, SM3 or later. This delivery includes required Adabas zaps that existed when Adabas Parallel Services 8.1.3 was released. Please check Software AG's Empower web site (<https://empower.softwareag.com>) for additional cluster-related Adabas zaps when installing Adabas Parallel Services.

If you install Adabas Online System (AOS), either the demo version delivered with Adabas 8.1.3 or the version 8.1.3 selectable unit is required. AOS is compiled under Natural version 3.1 and runs on that and all subsequent versions of Natural.

## Adabas Parallel Services Router

Adabas Parallel Services requires a compatible version of the Adabas router (ADASVC), which contains the Adabas Parallel Services component SVCCLU. The router must be installed on each operating system image in the parallel sysplex where either Adabas cluster nuclei or users are located.

## Using System Maintenance Aid

If you use Software AG's System Maintenance Aid (SMA), refer to the System Maintenance Aid documentation for information about the installation process.

The installation procedures described in this section correspond to the jobs that SMA creates to install the product.

If you do not use SMA, you can modify and use the sample JCL provided to unload the Adabas Parallel Services libraries from the installation tape.

## z/OS Systems Installation

This section describes the preparation for and installation of Adabas Parallel Services on z/OS systems.

### **Important:**

Before proceeding with the installation, ensure that the prerequisite environment has been established. Review the earlier sections of this part of the documentation for detailed information.

The installation procedure outlined in this section corresponds to the jobs that SMA creates to install the product. If you do not use SMA, you can modify and use the sample JCL provided in section *Unload the Installation Libraries to Disk* to unload the libraries from the installation tape.

- Step 1: Unload the Installation Libraries to Disk
- Step 2: APF-Authorize All Load Libraries
- Step 3: Customize ADACOM
- Step 4: Create a Startup Procedure for ADACOM
- Step 5: Create a Startup Procedure for Each Cluster Nucleus
- Step 6: Modify the z/OS Common Storage (CSA) Key
- Step 7: Start ADABAS Parallel Services

### Step 1: Unload the Installation Libraries to Disk

Sample JCL is provided in the base Adabas source library. This JCL can be modified and used to unload the Adabas Parallel Services libraries from the installation tape.

The Adabas Parallel Services installation tape is a standard label tape. Refer to the *Report of Tape Creation* that accompanies the tape for the volume serial number, density, media type, data set names, and data set sequence numbers.

The tape contains the installation data sets, a data set required by SMA, and one or more data sets containing maintenance fixes. Refer to the *Release Notes* for information about recommended fixes.

1. Allocate DASD.

The load library for Adabas Parallel Services requires one cylinder.

2. Copy the contents of the tape to disk.

To copy the Adabas Parallel Services data sets from the installation tape, use JCL similar to the following:

```
//COPY      JOB . . . .
//*
//*      Adabas Parallel Services Load Library
//*
//CPY1      EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//DDIN      DD DSN=ASMvrs.LOAD,DISP=OLD,UNIT=TAPE,
//           VOL=(,RETAIN,SER=volser),LABEL=n
//DDOUT     DD DSN=ASM.Vvrs.LOAD,DISP=(NEW,CATLG,DELETE),
//           UNIT=xxxxx,VOL=SER=vvvvvv,SPACE=(CYL,(cc,,bl))
//SYSIN     DD *
           COPY INDD=DDIN,OUTDD=DDOUT
/*
```

where:

<i>bl</i>	number of directory blocks for this library
<i>cc</i>	number of cylinders for this library
<i>vrs</i>	product version, revision, and system maintenance (SM) level
<i>volser</i>	volume serial number for the installation tape
<i>n</i>	position of the data set on the installation tape
ASM	product code for Adabas Parallel Services
<i>vvvvvv</i>	volume serial number of the disk used for the specified library
<i>xxxxx</i>	DASD device type

## Step 2: APF-Authorize All Load Libraries

Ensure that Adabas Parallel Services runs authorized.

To run authorized, the Adabas Parallel Services and Adabas version load libraries and all other load libraries in the STEPLIB concatenation must be APF-authorized.

- Ensure that all load libraries referenced in the STEPLIB concatenation for your Entire Net-Work, ADACOM, and Adabas cluster nuclei startup procedures are defined to the operating system as authorized libraries. If this is not done, ADACOM or the cluster nuclei will not initialize and may abnormally terminate, usually with an ABENDS047 or ABENDS306.

Either copy the base Adabas modules and the Adabas Parallel Services modules to an existing APF-authorized library or APF-authorize the Adabas Parallel Services library that was unloaded from the installation tape and the Adabas load library unloaded from the base Adabas installation tape.

## Step 3: Customize ADACOM

- Make any needed additions and modifications to the ADACOM member.

Read *ADACOM Initialization Parameters* for more information about specifying values for ADACOM parameters.

## Step 4: Create a Startup Procedure for ADACOM

An ADACOM initialization task is provided. This task must be active on the operating system image before any Adabas Parallel Services cluster nucleus is started. All load libraries in the STEPLIB concatenation of ADACOM must be APF-authorized.

ADACOM allocates the nucleus table for monitoring the active nuclei and the user table for monitoring users in the extended CSA (ECSA) above the 16MB line.

- The following is a sample job for running ADACOM:

```
//ADACOM PROC
//*-----*
//*      ADABAS PARALLEL SERVICES ADACOM Vv.r.s STARTUP      *
//*-----*
//ASMvrs EXEC PGM=ADARUN,REGION=4M,TIME=1440
//STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.LOAD
```

```

/*
//COMPRINT DD SYSOUT=*
//DDPRINT DD SYSOUT=X
//DDCARD DD *
ADARUN PROGRAM=ADACOM
/*
//DDKARTE DD *
ADACOM SVC=sv1,DBID=dbid1,NU=150
ADACOM SVC=sv1,DBID=dbid2,NU=150,F=Y
ADACOM SVC=sv2,DBID=dbid1,NU=500
ADACOM SVC=sv2,DBID=dbid3,NU=500
/*

```

The COMPRINT DD statement must be specified when running ADACOM. It defines an output data set for all general messages printed by ADACOM. For each SVC/DBID set specified in the ADACOM task, two subtasks are attached and SYSOUT data sets are dynamically allocated to receive all messages specific to that combination. The DD-name of the SYSOUT data sets are "Pssdddd" and "Dssdddd", where *ss* represents the last two digits of the SVC number and *dddd* are the five digits of the DBID.

ADACOM can also be installed as a started task; no special considerations apply.

You may also want to add the ADACOM task's start command to member COMMNDxx of SYS1.PARMLIB; this enables the ADACOM task to begin automatically at IPL.

## Step 5: Create a Startup Procedure for Each Cluster Nucleus

1. Customize a startup procedure to execute ADARUN.

For each Adabas cluster nucleus, customize the appropriate startup parameters and execute ADARUN from the Adabas load library.

2. Concatenate the Adabas Parallel Services load library ahead of the Adabas load library in the STEPLIB.
3. Allocate and format a Work data set for each nucleus.

All nuclei in an Adabas Parallel Services cluster share a common database resource; i.e., the same ASSO and DATA data sets. Each nucleus in the cluster must have its own Work data set; and all Work data sets within a cluster must have the same size and device type as defined in the general control block (GCB).

Use DISP=SHR on the DD card for the Work data set (DDWORKR1). During an offline or online restart/recovery, a nucleus may access the Work data sets belonging to other nuclei in the cluster.

4. Specify for each nucleus the ADARUN parameters CLUSTER, NUCID, CLUCACHETYPE, CLUCACHESIZE, CLULOCKSIZE, and CLUCACHEUNCHANGED.

Although each nucleus of an Adabas cluster shares the same database resource (DBID), each nucleus must have a unique NUCID value:

- a single (noncluster) nucleus: NUCID=0 (default)

- a cluster nucleus: NUCID=1-65000

Values for the CLUCACHESIZE and CLULOCKSIZE parameters are required for allocating the global data spaces. Read *Performance and Tuning* for sizing recommendations.

Use current values for all other ADARUN parameters, then reevaluate the values after monitoring the result. Ensure that each nucleus in the cluster is prepared to handle the entire workload for the common database, if necessary.

5. If protection logs are used, they must be dual or multiple logs and each nucleus must have its own. If one nucleus in the cluster runs with PLOGs, all nuclei in the cluster must run with PLOGs. The ADARUN PLOGRQ parameter must be the same for all nuclei (global parameter).

If user exit 2 or user exit 12 is supplied for one nucleus, the same user exit must be supplied for all nuclei in the cluster. User exit 12 must be used instead of user exit 2 if NCLOG/NPLOG is specified.

6. If command logs are used, each nucleus must have its own. If command logs are to be merged, they must be dual or multiple command logs and each nucleus in the cluster must have the same CLOG definition. To invoke automatic CLOG merging, CLOGMRG=YES must be specified in the ADARUN parameters or given as an operator or AOS/ADADBS command to any nucleus in the cluster.
7. The following sample JCL (job ASMNUC in the MVSJOBS data set) executes the Adabas ADARUN program to implement session parameters for an Adabas Parallel Services cluster nucleus.

Note that the Adabas Parallel Services library is concatenated ahead of the Adabas library in the STEPLIB.

```
//ADANUC      JOB      MSGCLASS=X,TIME=1440
//*-----
//* THIS IS A STARTUP JOB FOR A PARALLEL SERVICES NUCLEUS
//*
//* THE ADARUN PARMS HAVE TO BE CUSTOMIZED
//* DEPENDING ON THE USER'S ENVIRONMENT
//* DETAILS ARE PROVIDED IN THE OPERATIONS MANUAL
//*-----
//NUC          EXEC PGM=ADARUN,REGION=10M
//STEPLIB      DD      DISP=SHR,DSN=ASM.Vvrs.LOAD          <=== ASM LOAD
//              DD      DISP=SHR,DSN=ADABAS.Vvrs.LOAD      <=== ADABAS LOAD
//DDASSOR1     DD      DISP=SHR,DSN=EXAMPL.DByyy.ASSOR1    <=== ASSO
//DDDATAR1     DD      DISP=SHR,DSN=EXAMPL.DByyy.DATAR1    <=== DATA
//DDWORKR1     DD      DISP=SHR,DSN=EXAMPL.DByyy.WORKR1    <=== WORK
//DDPLOGR1     DD      DISP=SHR,DSN=EXAMPL.DByyy.PLOGR1    <=== PLOG1
//DDPLOGR2     DD      DISP=SHR,DSN=EXAMPL.DByyy.PLOGR2    <=== PLOG2
//DDDRUCK      DD      SYSOUT=X
//DDPRINT      DD      SYSOUT=X
//MPMDUMP      DD      SYSOUT=X
//DDCARD       DD      *
ADARUN PROG=ADANUC
ADARUN CLUSTER=LOCAL
ADARUN MODE=MULTI
ADARUN SVC=num          <--- INSERT YOUR SVC NUMBER
ADARUN DBID=yyy         <--- INSERT YOUR DATABASE ID
ADARUN NUCID=nnnnn     <--- INSERT YOUR NUCLEUS ID
ADARUN CLUCACHESIZE=size <--- INSERT YOUR CACHE SIZE
ADARUN CLULOCKSIZE=size <--- INSERT YOUR LOCK SIZE
ADARUN DEVICE=3390
ADARUN CT=60
```

```

ADARUN OPENRQ=NO          ---> DEFAULT = YES
ADARUN PLOGRQ=NO          ---> DEFAULT = YES
ADARUN LBP=900000
ADARUN LFIOP=300000
ADARUN LCP=10000
ADARUN LFP=12000
ADARUN LWP=350000
ADARUN LI=10000
ADARUN LS=20000
ADARUN LU=65535
ADARUN LP=1500
ADARUN NAB=16
ADARUN NISNHQ=1000        ---> FOR BATCH NATURAL INPL
ADARUN NT=8
ADARUN TT=600
ADARUN TNAA=600
ADARUN TNAE=600
ADARUN TLSCMD=300
ADARUN DUALPLS=6750        ---> 50 CYL
ADARUN DUALPLD=3390
ADARUN LOGGING=NO
/*

```

## Step 6: Modify the z/OS Common Storage (CSA) Key

- If necessary, modify the z/OS Common Storage (CSA) key.

With z/OS version 1.9, the default setting of the ALLOWUSERKEYCSA parameter has been changed to NO, prohibiting the allocation of CSA storage in a non-privileged storage key. In past releases of Adabas Parallel Services, its data structures in common storage (PLXCB and subordinate control blocks) were stored in user key storage. Due to the change of the ALLOWUSERKEYCSA default, you must either change the ALLOWUSERKEYCSA setting in z/OS 1.9 to YES or accept the ALLOWUSERKEYCSA default setting in z/OS 1.9 (NO). If you accept the default, you will need to modify your Adabas Parallel Services installation and the Adabas Parallel Services and ADACOM startup procedures to use a system key.

For more information, read *Modifying the z/OS Common Storage (CSA) Key*

## Step 7: Start ADABAS Parallel Services

1. Start the ADACOM initialization task on the operating system image that is hosting the Adabas Parallel Services cluster environment.
2. Start the Adabas Parallel Services cluster nuclei in any order.

The Adabas Parallel Services cluster is now ready to process user requests.

Rules for subsequent starts of Adabas Parallel Services are described in *Adabas Parallel Services Operations*.

## Modifying the z/OS Common Storage (CSA) Key

With z/OS version 1.9, the default setting of the ALLOWUSERKEYCSA parameter has been changed to NO, prohibiting the allocation of CSA storage in a non-privileged storage key. In past releases of Adabas Cluster Services and Adabas Parallel Services, its data structures in common storage (PLXCB and subordinate control blocks) were stored in user key storage. Due to the change of the

ALLOWUSERKEYCSA default, you must either:

- Change the ALLOWUSERKEYCSA setting in z/OS 1.9 to YES. In this case, you need do nothing more, but you incur the security risk that unauthorized programs might modify storage owned by Adabas Cluster Services and Adabas Parallel Services.
- Accept the ALLOWUSERKEYCSA default setting in z/OS 1.9 (NO), thus eliminating the security risk. If you choose this option, however, you will need to modify your Adabas Cluster Services or Adabas Parallel Services installation and the Adabas Cluster Services, Adabas Parallel Services, and ADACOM startup procedures to use a system key. This section provides information on modifying the Adabas Cluster Services and the startup procedures to use a system key and on the prerequisites for doing so.

This section covers the following topics:

- Requirements and Recommendations
- Steps to Modify the CSA Key

## Requirements and Recommendations

The following requirements and recommendations apply when modifying the CSA storage key:

- The Adabas version 8 ADASVC is required. This procedure, when used with the Adabas 8 ADASVC, supports Adabas 7 and 8 used with Adabas Cluster Services versions 7.4 and 8, and with Adabas Parallel Services versions 7.4, 7.5 and 8. Older releases of Adabas Cluster Services or Adabas Parallel Services may require that some fixes be applied before attempting this, so be sure to check the Knowledge Center of Software AG's Empower (at <https://empower.softwareag.com>) web site for them.
- APF authorization is required to be assigned a system key. Adabas Parallel Services and Adabas Cluster Services nuclei must run authorized, but utilities and classic nuclei may bypass APF authorization by applying an optional fix to ADASVC. If APF authorization is not in effect and the execution JCL has been modified as described below, the following message will appear during initiation:

```
IEF188I PROBLEM PROGRAM ATTRIBUTES ASSIGNED
```

## Steps to Modify the CSA Key

In general, you can modify the CSA key one system at a time for each database. On each system, you can apply the modifications to each DBID/SVC combination independently. One exception to this is when ADACOM controls more than one DBID/SVC pair. In this case, you must modify all DBID/SVC combinations defined to that ADACOM at the same time. All database member nuclei on the system using that ADASVC or ADACOM must be stopped and restarted.

The modification to the CSA storage key involves the use of program name ADARUNK, which is an alias for ADARUN. When your modifications are complete (see the steps below), your installation and all of the appropriate execution JCL will have been changed to use ADARUNK (PGM=ADARUNK) instead of ADARUN. Once this is done, z/OS will assign a system key to all tasks for the nuclei or ADACOM processes. Although only ADACOM tasks and nuclei associated with Adabas Cluster Services or Adabas Parallel Services obtain CSA storage and are required to perform these modification steps to avoid using a user key, the ADARUNK program can also be used for classic Adabas nuclei and utilities.



► **To modify the CSA storage key to a system key:**

1. Add the following entry for program ADARUNK to an active Program Properties Table description in SYS1.PARMLIB(SCHEDxx):

```
PPT PGMNAME(ADARUNK) KEY(4)
```

2. Activate the new PPT entry. This will occur automatically at the next IPL, or you can use the z/OS operator command SET SCH(xx) to activate it without an IPL.
3. Optionally, create permissions for the FACILITY class resource IEAABD.DMPAKEY to allow SYSUDUMP to include system key storage.

When executing with a system key, SYSUDUMP will not contain the system key storage. The following message will appear:

```
IEA848I DUMP SUPPRESSED - ABDUMP MAY NOT DUMP STORAGE FOR KEY 0-7
```

Since cluster nuclei and ADACOM may obtain large storage objects, we suggest including a //SVCDUMP DD statement in the execution JCL instead of a //SYSUDUMP DD statement. Otherwise, enable SYSUDUMP for system key tasks by permitting the user to have read access to the IEAABD.DMPAKEY resource in the FACILITY class. For more information, refer to your *z/OS Security Server (RACF) Security Administrator's Guide*.

4. Examine the Adabas load library to see if an entry exists for ADARUNK. If it does not, create an alias ADARUNK for member ADARUN. You can do this using programs such as PDSTOOLS or the freeware PDS or use the Linker/Binder to relink ADARUN and create the alias:

```
//LINK EXEC PGM=IEWL,PARM='Map,Let,List(All),Ncal'
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=<Adabas load library>,DISP=SHR
//SYSLIN DD *
      Include SYSLMOD(ADARUN)
      Order RUNMVS,USRZAP,RUNIND
      Mode AMode(31),RMode(24)
      Setcode AC(1)
      Entry RUNMVS
      Alias ADARUNK
      Name ADARUN(R)
/*
```

5. Stop all Adabas Parallel Services and Adabas Cluster Services member nuclei that use the same DBID/SVC pair.
6. Any existing user key PLXCB structures for a DBID/SVC combination must be deleted. This happens automatically at the next IPL or if ADASVC is reinstalled using ADASIP.

ADACOM may be used to deallocate the PLXCB without an IPL by specifying NU=0 for every DBID/SVC pair, and then stop and restart ADACOM.

The following sequence of operator commands may also be used to deallocate a PLXCB after stopping all nuclei but before stopping ADACOM:

```
F <ADACOM job>,DBID=<dbid>,SVC=<svc>,ADAEND
F <ADACOM job>,DBID=<dbid>,SVC=<svc>,NU=0
```

- If ADACOM is running and controlling the database, it must be stopped.

**Caution:**

If ADACOM is controlling more than one DBID/SVC pair, all pairs must be processed at the same time.

- Modify the startup JCL for each Adabas Parallel Services or Adabas Cluster Services member nucleus and ADACOM to specify program ADARUNK instead of ADARUN. For example, for an Adabas Parallel Services or Adabas Cluster Services nucleus you might specify:

```
//NUC240 EXEC PGM=ADARUNK
```

For an ADACOM startup procedure, you might specify:

```
//ALSvrs EXEC PGM=ADARUNK,REGION=4M,TIME=nnnnn
```

- Modify the JCL for each Adabas Parallel Services or Adabas Cluster Services member nucleus and ADACOM to include a //SVCDUMP DD DUMMY statement.
- Restart all ADACOM jobs and Adabas Parallel Services and Adabas Cluster Services nuclei that you stopped in Step 5.
- Repeat steps 5 through 10 for each DBID/SVC pair.

**Caution:**

If ADACOM is controlling more than one DBID/SVC pair, all pairs must be processed at the same time.

## z/VSE Systems Installation

This section describes the preparation for and installation of Adabas Parallel Services on z/VSE systems.

The installation procedure outlined in this section corresponds to the jobs that SMA creates to install the product. If you do not use SMA, you can modify and use the sample JCL provided in section 1. *Unload the Installation Libraries to Disk* of the z/OS systems installation to unload the libraries from the installation tape.

- Step 1: Install the Adabas Parallel Services Library
- Step 2: Customize the ADACOM Procedure
- Step 3: Create a Startup Procedure for ADACOM
- Step 4: Create a Startup Procedure for Each Cluster Nucleus
- Step 5: Start Adabas Parallel Services

## Step 1: Install the Adabas Parallel Services Library

### 1. Define the sublibrary.

Adabas Parallel Services users must define an additional sublibrary in the Adabas library for the Adabas Parallel Services components.

A sample job to accomplish this is as follows:

```
// JOB ASMDEF DEFINE NON-VSAM SUBLIB
// OPTION LOG
// DLBL SAGLIB,'ADABAS.Vvrs.LIBRARY',2099/365,SD
// EXTENT SYS010
// ASSGN SYS010,DISK,VOL=vvvvvv,SHR
// EXEC LIBR
DEFINE S=SAGLIB.ASMvrs REUSE=AUTO R=Y
LD L=SAGLIB OUTPUT=STATUS
/*
/ &
```

where

SYS010	is the logical unit for the Adabas library
vvvvvv	is the volume for the Adabas library
vrs	is the Adabas version/revision/system maintenance (SM) level

### 2. Restore the Adabas Parallel Services Sublibrary

A sample job to restore the Adabas Parallel Services components is as follows:

```
// JOB ASMRST RESTORE NON-VSAM
// OPTION LOG
// ASSGN SYS006,cuu
// PAUSE MOUNT ADABAS INSTALL TAPE cuu
// MTC REW,SYS006
// MTC FSF,SYS006,tt
// DLBL SAGLIB,'ADABAS.Vvrs.LIBRARY'
// EXTENT SYS010
// ASSGN SYS010,DISK,VOL=vvvvvv,SHR
// EXEC LIBR
RESTORE SUB=SAGLIB.ASMvrs -
TAPE=SYS006 LIST=Y R=Y
LD SUB=SAGLIB.ASMvrs OUTPUT=NORMAL
/*
// MTC REW,SYS006
/ &
```

where

SAGLIB	is the Adabas library name
SYS010	is the logical unit for the Adabas library
SYS006	is the Adabas installation tape
<i>cuu</i>	is the physical unit address of the tape drive
<i>tt</i>	is the number of tape marks to space forward (see the <i>Report of Tape Creation</i> )
<i>vvvvvv</i>	is the volume for the Adabas library
<i>vrs</i>	is the Adabas version/revision/system maintenance (SM) level

## Step 2: Customize the ADACOM Procedure

- Make any needed additions and modifications to the ADACOM member.

Read *ADACOM Initialization Parameters* for more information about specifying values for ADACOM parameters.

## Step 3: Create a Startup Procedure for ADACOM

An ADACOM initialization task is provided. This task must be active on the operating system image before any Adabas Parallel Services cluster nucleus is started.

ADACOM allocates the nucleus table for monitoring the active nuclei and the user table for monitoring users in the SVA above the 16MB line.

- The following is a sample job for running ADACOM:

```
// JOB ADACOM
// *-----*
// *      ADABAS PARALLEL SERVICES ADACOM Vv.r.s STARTUP      *
// *-----*
// LIBDEF PHASE,SEARCH=(SAGLIB.ASMvrs,SAGLIB.ADAvrs)
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADACOM
/*
ADACOM SVC=sv1,DBID=dbid1,NU=150
ADACOM SVC=sv1,DBID=dbid2,NU=150,F=Y
ADACOM SVC=sv2,DBID=dbid1,NU=500
ADACOM SVC=sv2,DBID=dbid3,NU=500
/*
/&
```

### Note:

ADACOM must run in a dynamic partition.

## Step 4: Create a Startup Procedure for Each Cluster Nucleus

1. Customize a startup procedure to execute ADARUN.

For each Adabas cluster nucleus, customize the appropriate startup parameters and execute ADARUN from the Adabas load library.

**Note:**

Each Adabas cluster nucleus must run in a dynamic partition.

2. Concatenate the Adabas Parallel Services load library ahead of the Adabas load library in the LIBDEF PHASE SEARCH statement.
3. Allocate and format a Work data set for each nucleus.

All nuclei in an Adabas Parallel Services cluster share a common database resource; i.e., the same ASSO and DATA data sets. Each nucleus in the cluster must have its own Work data set; and all Work data sets within a cluster must have the same size and device type as defined in the general control block (GCB).

4. Specify for each nucleus the ADARUN parameters CLUSTER, NUCID, CLUCACHESIZE, CLULOCKSIZE, and CLUCACHEUNCHANGED.

Although each nucleus of an Adabas cluster shares the same database resource (DBID), each nucleus must have a unique NUCID value:

- a single (noncluster) nucleus: NUCID=0 (default)
- a cluster nucleus: NUCID=1-65000

Values for the CLUCACHESIZE and CLULOCKSIZE parameters are required for allocating the global data spaces. Read *Performance and Tuning*.

Use current values for all other ADARUN parameters, then reevaluate the values after monitoring the result. Ensure that each nucleus in the cluster is prepared to handle the entire workload for the common database, if necessary.

5. If protection logs are used, they must be dual or multiple logs and each nucleus must have its own. If one nucleus in the cluster runs with PLOGs, all nuclei in the cluster must run with PLOGs. The ADARUN PLOGRQ parameter must be the same for all nuclei (global parameter). If user exit 2 or user exit 12 is supplied for one nucleus, the same user exit must be supplied for all nuclei in the cluster.
6. If command logs are used, each nucleus must have its own. If command logs are to be merged, they must be dual or multiple command logs and each nucleus in the cluster must have the same CLOG definition. To invoke automatic CLOG merging, CLOGMRG=YES must be specified in the ADARUN parameters or given as an operator or AOS/ADADBS command to any nucleus in the cluster.

The following sample JCL executes the Adabas ADARUN program to implement session parameters for an Adabas Parallel Services cluster nucleus.

**Note:**

The Adabas Parallel Services library is concatenated ahead of the Adabas library in the LIBDEF PHASE SEARCH statement.

```
// JOB ADANUC
//*-----
//*   THIS IS A STARTUP JOB FOR A PARALLEL SERVICES NUCLEUS
//*
//*   THE ADARUN PARMS HAVE TO BE CUSTOMIZED
```

```

/*  DEPENDING ON THE USER'S ENVIRONMENT
/*  DETAILS ARE PROVIDED IN THE OPERATIONS MANUAL
/*-----
/*  LIBDEF PHASE,SEARCH=(SAGLIB.ASMvrs,SAGLIB.ADAvrs)
/*  DLBL ASSOR1,'EXAMPL.DBYYY.ASSOR1'
/*  EXTENT SYS020
/*  DLBL DATAR1,'EXAMPL.DBYYY.DATAR1'
/*  EXTENT SYS021
/*  DLBL WORKR1,'EXAMPL.DBYYY.WORKR1'
/*  EXTENT SYS022
/*  DLBL PLOGR1,'EXAMPL.DBYYY.PLOGR1'
/*  EXTENT SYS027
/*  DLBL PLOGR2,'EXAMPL.DBYYY.PLOGR2'
/*  EXTENT SYS028
/*  ASSGN SYS009,PRINTER
/*  EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADANUC
ADARUN CLUSTER=LOCAL
ADARUN MODE=MULTI
ADARUN SVC=SSS          <--- INSERT YOUR SVC NUMBER
ADARUN DBID=YYY        <--- INSERT YOUR DATABASE ID
ADARUN NUCID=NNNNN     <--- INSERT YOUR NUCLEUS ID
ADARUN CLUCACHESIZE=XXXXXXXX <--- INSERT YOUR CACHE SIZE
ADARUN CLULOCKSIZE=XXXXXXXX <--- INSERT YOUR LOCK SIZE
ADARUN DEVICE=3390
ADARUN CT=60
ADARUN OPENRQ=NO      ---> DEFAULT = YES
ADARUN PLOGRQ=NO      ---> DEFAULT = YES
ADARUN LBP=900000
ADARUN LFIOP=300000
ADARUN LCP=10000
ADARUN LFP=12000
ADARUN LWP=350000
ADARUN LI=10000
ADARUN LS=20000
ADARUN LU=65535
ADARUN LP=1500
ADARUN NAB=16
ADARUN NISNHQ=1000    ---> FOR BATCH NATURAL INPL
ADARUN NT=8
ADARUN TT=600
ADARUN TNAA=600
ADARUN TNAE=600
ADARUN TLSCMD=300
ADARUN DUALPLS=6750   ---> 50 CYL
ADARUN DUALPLD=3390
ADARUN LOGGING=NO
/*
/&

```

## Step 5: Start Adabas Parallel Services

1. Start the ADACOM initialization task on the operating system image that is hosting the Adabas Parallel Services cluster environment.
2. Start the Adabas Parallel Services cluster nuclei in any order.

The Adabas Parallel Services cluster is now ready to process user requests.

Rules for subsequent starts of Adabas Parallel Services are described in *Adabas Parallel Services Operations*.

## BS2000/OSD Systems Installation

### Important:

Before proceeding with the installation, ensure that the prerequisite environment has been established. Review the earlier sections of this section for detailed information.

The installation procedure outlined in this section corresponds to the jobs that SMA creates to install the product. If you do not use SMA, you can modify and use the sample JCL provided in Step 1 to unload the libraries from the installation tape.

Please note that the ADACOM initialization task mentioned below starts two subtask jobs with the ID of the ADACOM task. These are started with the NO TIME LIMIT or NTL attribute. Please ensure that this is allowed in the ID where this is started. These subtask jobs will be automatically removed when ADACOM is stopped or canceled.

- Step 1: Install the Adabas Parallel Services library.
- Step 2: Customize ADACOM
- Step 3: Create a Startup Procedure for ADACOM
- Step 4: Create a Startup Procedure for Each Cluster Nucleus
- Step 5: Start Adabas Parallel Services

### Step 1: Install the Adabas Parallel Services library.

1. Copy the Library SRV $nnn$ .LIB from Tape to Disk

#### Note:

This step is not necessary if you have already copied the library SRV $nnn$ .LIB from another Software AG tape. For more information, refer to the element #READ-ME in this library.

The library SRV $nnn$ .LIB is stored on the tape as the sequential file SRV $nnn$ .LIBS containing LMS commands. The current version  $nnn$  can be obtained from the Report of Tape Creation. To convert this sequential file into an LMS-library, execute the following commands:

```
/IMPORT-FILE SUPPORT=*TAPE(FILE-NAME=SRV $nnn$ .LIBS, -
/ VOLUME=volser, DEV-TYPE=tape-device)
/ADD-FILE-LINK LINK-NAME=EDTSAM, FILE-NAME=SRV $nnn$ .LIBS, -
/ SUPPORT=*TAPE(FILE-SEQ=3), ACC-METH=*BY-CAT, -
/ BUF-LEN=*BY-CAT, REC-FORM=*BY-CAT, REC-SIZE=*BY-CAT
/START-EDT
@READ '/'
@SYSTEM 'REMOVE-FILE-LINK EDTSAM'
@SYSTEM 'EXPORT-FILE FILE-NAME=SRV $nnn$ .LIBS'
@WRITE 'SRV $nnn$ .LIBS'
@HALT
/ASS-SYSDTA SRV $nnn$ .LIBS
```

```

/ MOD-JOB-SW ON=1
/ START-PROG $LMS
/ MOD-JOB-SW OFF=1
/ ASS-SYSDTA *PRIMARY

```

where *tape-device* is the device type of the tape (for example, TAPE-C4) and *volser* is the volume serial number of the tape (see the *Report of Tape Creation*).

## 2. Copy the Procedure COPY.PROC from Tape to Disk

Call the procedure P.COPYTAPE in the library SRVnnn.LIB to copy the procedure COPY.PROC to disk:

```

/ CALL-PROCEDURE (SRVnnn.LIB,P.COPYTAPE), -
/ (VSNT=volser, DEVT=<tape-device>)

```

If you use a TAPE-C4 device, you can omit the parameter DEVT.

## 3. Copy all Product Files from Tape to Disk

Enter the procedure COPY.PROC to copy all Software AG product files from tape to disk:

```

/ ENTER-PROCEDURE COPY.PROC, DEVT=tape-device

```

If you use a TAPE-C4 device, you can omit the parameter DEVT. The results of this procedure are written to the file L.REPORT.SRV.

## Step 2: Customize ADACOM

- Make any needed additions and modifications to the ADACOM member.

Read *ADACOM Initialization Parameters* for more information about specifying values for ADACOM parameters.

## Step 3: Create a Startup Procedure for ADACOM

- An ADACOM initialization task is provided. This task must be active on the operating system image before any Adabas Parallel Services cluster nucleus is started.

ADACOM allocates the nucleus table for monitoring the active nuclei and the user table for monitoring users in the dataspace.

If you have applied zap AB814001, use the following sample job for running ADACOM:

```

/ BEGIN-PROC C
/ ASS-SYSOUT L.ADA99.COM.O
/ RE-F-LI *
/ SET-JOB-STEP
/ DEL-FI DDCARD
/ SET-JOB-STEP
/ CRE-FI DDCARD,PUB( )
/ SET-JOB-STEP
/ DEL-FI DDKART
/ SET-JOB-STEP
/ MOD-J-SW ON=( 4,5)
/ ASS-SYSDTA *SYSCMD
/ STA-PROG EDT

```



```

ADARUN PROG=ADACOM, IDTNAME=AIDT01
ADARUN SUBMPSZ=5000000
@W 'DDCARD' O
@H
/SE-F-LI EDTSAM, DDKART, REC-FORM=F (REC-SIZE=80)
/STA-PROG EDT
ADACOM IDTNAME=ADAIDT01, DBID=99, NU=400
ADACOM IDTNAME=ADAIDT01, DBID=199, NU=2000
@W '/' O
@H
/SET-JOB-STEP
/MOD-J-SW OFF=(4,5)
/MOD-TEST DUMP=YES
/ASS-SYSLST L.ADA99.COM.L
/SET-JOB-STEP
/SE-F-LI DDLIB, ADAvrs.MOD
/SE-F-LI BLSLIB00, ADAvrs.MOD
/SE-F-LI COMPRINT, COMPRINT.FILE, REC-FORM=F (REC-SIZE=160)
/SE-F-LI DDDRUCK, #DRUCK
/SE-F-LI DDCARD, DDCARD
/SE-F-LI DDKARTE, DDKART
/ASS-SYSDTA *PRIM
/STA-PROG *M(ADAvrs.MOD, ADARUN), R-M=*A(ALT=Y)
/ASS-SYSLST *PRIM
/ASS-SYSOUT *PRIM
/END-PROC

```

This has the advantage that the DDPRINT file can be seen while the job is active.

If you have *not* applied zap AB814001, use the following sample job for running ADACOM:

```

/BEGIN-PROC C
/ASS-SYSOUT L.ADA99.COM.O
/RE-F-LI *
/SET-JOB-STEP
/DEL-FI DDCARD
/SET-JOB-STEP
/CRE-FI DDCARD, PUB()
/SET-JOB-STEP
/DEL-FI DDKART
/SET-JOB-STEP
/DEL-FI L.ADACOM.99
/SET-JOB-STEP
/DEL-FI L.ADACOMP.99
/SET-JOB-STEP
/DEL-FI L.ADACOM.199
/SET-JOB-STEP
/DEL-FI L.ADACOMP.199
/SET-JOB-STEP
/MOD-J-SW ON=(4,5)
/ASS-SYSDTA *SYSCMD
/STA-PROG EDT
ADARUN PROG=ADACOM, IDTNAME=AIDT01
ADARUN SUBMPSZ=5000000
@W 'DDCARD' O
@H
/SE-F-LI EDTSAM, DDKART, REC-FORM=F (REC-SIZE=80)
/STA-PROG EDT
ADACOM IDTNAME=ADAIDT01, DBID=99, NU=400
ADACOM IDTNAME=ADAIDT01, DBID=199, NU=2000
@W '/' O
@H

```

```

/SET-JOB-STEP
/MOD-J-SW OFF=( 4,5)
/SE-F-LI EDTSAM,L.ADACOM.99,REC-FORM=F(REC-SIZE=80)
/STA-PROG EDT
*
@W ' / ' O
@H
/SET-JOB-STEP
/MOD-J-SW OFF=( 4,5)
/SE-F-LI EDTSAM,L.ADACOMP.99,REC-FORM=F(REC-SIZE=80)
/STA-PROG EDT
*
@W ' / ' O
@H
/SET-JOB-STEP
/MOD-J-SW OFF=( 4,5)
/SE-F-LI EDTSAM,L.ADACOM.199,REC-FORM=F(REC-SIZE=80)
/STA-PROG EDT
*
@W ' / ' O
@H
/SET-JOB-STEP
/MOD-J-SW OFF=( 4,5)
/SE-F-LI EDTSAM,L.ADACOMP.199,REC-FORM=F(REC-SIZE=80)
/STA-PROG EDT
*
@W ' / ' O
@H
/SET-JOB-STEP
/MOD-J-SW OFF=( 4,5)
/MOD-TEST DUMP=YES
/ASS-SYSLST L.ADA99.COM.L
/SET-JOB-STEP
/SE-F-LI DDLIB,ADAvrs.MOD
/SE-F-LI BLSLIB00,ADAvrs.MOD
/SE-F-LI COMPRINT,COMPRINT.FILE,REC-FORM=F(REC-SIZE=160)
/SE-F-LI DDRUCK,#DRUCK
/SE-F-LI DDCARD,DDCARD
/SE-F-LI DDKARTE,DDKART
/SE-F-LI DI100099,L.ADACOM.99
/SE-F-LI PI100099,L.ADACOMP.99
/SE-F-LI DI100199,L.ADACOM.199
/SE-F-LI PI100199,L.ADACOMP.199
/ASS-SYSDTA *PRIM
/STA-PROG *M(ADAvrs.MOD,ADARUN),R-M=*A(ALT=Y)
/ASS-SYSLST *PRIM
/ASS-SYSOUT *PRIM
/END-PROC

```

The output link names (**bold and green** in the sample job above) are derived as follows:

- Character 1: Either a "D" (ADADSP subtask) or a "P" (ADACOT subtask).
- Character 2: The fourth character of the IDTNAME.
- Character 3: The last character of the IDTNAME.
- Characters 4-8: The DBID, with leading zeros.

## Step 4: Create a Startup Procedure for Each Cluster Nucleus

1. Customize a startup procedure to execute ADARUN.

For each Adabas cluster nucleus, customize the appropriate startup parameters and execute ADARUN from the Adabas load library.

2. Concatenate the Adabas Parallel Services load library ahead of the Adabas load library in the BLSLIB search chain.

```
RUN-MODE=*ADV(ALT-LIB=Y)
```

3. Allocate and format a Work data set (file) for each nucleus.

All nuclei in an Adabas Parallel Services cluster share a common database resource; i.e., the same ASSO and DATA data sets (files). Each nucleus in the cluster must have its own Work data set (file); and all Work data sets (files) within a cluster must have the same size and device type as defined in the general control block (GCB).

Use SUP=DISK(SHARE-UPD=YES) on the link name for the work container (DDWORKR1). During an offline or online restart/recovery, a nucleus may access the Work data sets (files) belonging to other nuclei in the cluster.

4. For each nucleus, specify the ADARUN parameters CLUSTER, NUCID, CLUCACHESIZE, CLULOCKSIZE, and CLUCACHEUNCHANGED.

Although each nucleus of an Adabas cluster shares the same database resource (DBID), each nucleus must have a unique NUCID value:

- a single (noncluster) nucleus: NUCID=0 (default)
- a cluster nucleus: NUCID=1-65000

Adabas Parallel Services provides default global cache and lock area names for each cluster that are unique in the Adabas Parallel Services system.

Values for the CLUCACHESIZE and CLULOCKSIZE parameters are required for allocating the global data spaces. Read *Performance and Tuning*.

Use current values for all other ADARUN parameters, then reevaluate the values after monitoring the result. Ensure that each nucleus in the cluster is prepared to handle the entire workload for the common database, if necessary.

5. If protection logs are used, each nucleus must have its own dual or multiple logs. If one nucleus in the cluster runs with PLOGs, all nuclei in the cluster must run with PLOGs. The ADARUN PLOGRQ parameter must be the same for all nuclei (global parameter).

If user exit 2 or user exit 12 is supplied for one nucleus, the same user exit must be supplied for all nuclei in the cluster. User exit 12 must be used instead of user exit 2 if NCLOG/NPLOG is specified.

6. If command logs are used, each nucleus must have its own. If command logs are to be merged, they must be dual or multiple command logs and each nucleus in the cluster must have the same CLOG definition. To invoke automatic CLOG merging, CLOGMRG=YES must be specified in the ADARUN parameters or given as an operator or AOS/ADADBS command to any nucleus in the

cluster.

The following sample JCL executes the Adabas ADARUN program to implement session parameters for an Adabas Parallel Services cluster nucleus.

**Note:**

The Adabas Parallel Services library is concatenated ahead of the Adabas library in the BLSLIB chain.

```

/.ADANUC SET-LOGON-PARAMETERS
/REMARK This is a startup job for the Parallel Services Nucleus
/REMARK
/REMARK The parameters need to be customized according to the
/REMARK customer's environment.
/REMARK Details are available in the Operations Manual
/MOD-TEST DUMP=YES
/COPY-FILE L.NUC.SM,L.NUC.SM.OLD
/SET-JOB-STEP
/ASS-SYSLST L.NUC.SM
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK BLSLIB00,ADAvrs.MOD
/SET-FILE-LINK BLSLIB01,ADAvrs.SM.MOD
/SET-FILE-LINK DDASSOR1,ADAdd.ASSO ,SUP=DISK(SHARE-UPD=YES)
/SET-FILE-LINK DDDATAR1,ADAddddDATA ,SUP=DISK(SHARE-UPD=YES)
/SET-FILE-LINK DDWORKR1,ADAddd.WORK ,SUP=DISK(SHARE-UPD=YES)
/SET-FILE-LINK DDPLOGR1,ADAddd.PLOGR1 ,SUP=DISK(SHARE-UPD=YES)
/SET-FILE-LINK DDPLORR2,ADAddd.PLOGR2 ,SUP=DISK(SHARE-UPD=YES)
/START-PROGRAM (ADAvrs.MOD,ADARUN),PR-MO=ANY,RUN-MO=A(A-L=Y)
ADARUN PROG=ADANUC
ADARUN CLUSTER=LOCAL
ADARUN MODE=MULTI
ADARUN IDTNAME=ADaiiii <--- INSERT YOUR IDTNAME
ADARUN DBID=ddd <--- INSERT YOUR DATABASE ID
ADARUN NUCID=nnnnn <--- INSERT YOUR NUCLEUD ID
ADARUN CLUCACHESIZE=nnnn <--- INSERT YOUR CACHE SIZE
ADARUN CLULOCKSIZE=nnnn <--- INSERT YOUR LOCK SIZE
ADARUN DEVICE=2002
ADARUN CT=60
ADARUN OPENRQ=NO ---> DEFAULT = YES
ADARUN PLOGRQ=NO ---> DEFAULT = YES
ADARUN LBP=900000
ADARUN LFIOP=300000
ADARUN LCP=10000
ADARUN LFP=12000
ADARUN LWP=350000
ADARUN LI=10000
ADARUN LS=20000
ADARUN LU=65535
ADARUN LP=1500
ADARUN NAB=16
ADARUN NISNHQ=1000 ---> FOR BATCH NATURAL INPL
ADARUN NT=8
ADARUN TT=600
ADARUN TNAA=600
ADARUN TNAE=600
ADARUN TLSCMD=300
ADARUN DUALPLS=6750 ---> 50 CYL
ADARUN DUALPLD=2002
/LOGOFF NOSPOOL

```

## Step 5: Start Adabas Parallel Services

1. Start the ADACOM initialization task on the operating system image that is hosting the Adabas Parallel Services cluster environment.

As mentioned earlier, the ADACOM starts two subtask jobs with the same ID as the ADACOM job itself. If possible, these jobs write data that is useful in diagnosing errors. If you have applied zap AB814001, the data is written to SYSOUT and DDPRINT (if you have it defined). You can see the contents of DDPRINT using the utility SHOW-FILE while the job is running (this is not possible with the SYSOUT file); the contents of SYSOUT can only be seen when the ADACOM function is stopped. If you have not applied zap AB814001, the DDPRINT file is not written to by the subtask. The system generates a SYSOUT file in the following format:

```
L.<O/L>,<tsn>.ADABSP##.<timestamp>
```

where <O/L> is either "O" for SYSOUT or "L" for SYSLST, <tsn> is the task number of the ADACOM job, and <timestamp> is a 16-character printable hexadecimal string representing the STCK time when the subtask was started.

2. Start the Adabas Parallel Services cluster nuclei in any order.

The Adabas Parallel Services cluster is now ready to process user requests.

Rules for subsequent starts of Adabas Parallel Services are described in the section *Adabas Parallel Services Operations*.

**Note:**

A cluster is a group of databases sharing the processing of one logical database ID (DBID).