

Adabas Parallel Services

Operations

Version 8.5.4

October 2019

This document applies to Adabas Parallel Services Version 8.5.4 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2019 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software GmbH and/or its subsidiaries is located at <https://softwareag.com/licenses>.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software GmbH.

Document ID: ASM-OP-854-20251030

Table of Contents

Preface	v
1 Conventions	1
2 About this Documentation	3
Document Conventions	4
Online Information and Support	4
Data Protection	5
3 Initialization	7
Sequence	8
ADACOM Initialization Process	8
Adabas Parallel Services Cluster Nucleus Process	9
PPT Processing	10
4 Termination	13
Normal Termination	14
Abnormal Termination	14
5 Using Alert and Timeout Settings	17
Using Cancel Alerts	18
Using Self-Termination Alerts	19
Using the Self-Termination Operator Query	20
Using Messaging Performance Statistics	21
6 Collaboration Between Cluster Nuclei	23
Buffer Flush Independence	24
Update Command Synchronization	25
7 Backout Processing	27
8 Restart/Recovery Processing	29
Offline Recovery (Session Autorestart)	30
Online Recovery	30
Automatic Restart Management (ARM)	31
Archive Recovery	31
9 Utility Processing	33
ADACHK PPTPRINT - Print/Dump Parallel Participant Table	34
ADADBS OPERCOM Commands	34
ADADBS REFRESHSTATS - Refresh Statistical Values	35
ADAPLP IPLOGPRI - Print Sequential Intermediate Data Sets	36
ADARAI - Adabas Recovery Aid	36
ADAREP - Checkpoint Information Extended	36
ADARES CLCOPY - Copy/Merge Nucleus Cluster Command Logs	36
ADARES PLCOPY - Copy/Merge Nucleus Cluster Protection Logs	38
ADARES MERGE CLOG - Merge Nucleus Cluster Command Logs	44
ADARES BACKOUT and REGENERATE— Uniquely Identifying Checkpoints	45
ADASAV Processing Change	46
ADASAV RESTPLOG -- Uniquely Identifying Checkpoints	47
10 Cluster Nucleus Session End Statistics	49

General Nucleus Information	50
Input/Output Statistics	50
Command Statistics	52
User Sessions	53
Efficiency Statistics	54
Messaging Statistics	55
Global Cache Statistics (Cluster Nucleus Only)	55
Global Lock Statistics (Cluster Nucleus Only)	57
WORK Data Set I/O Statistics (Cluster Nucleus Only)	58
Data Set Activity Statistics	59
11 Performance and Tuning	61
ADARUN Parameter Settings	62
Allocating Work Data Set Space	62
Using Close (CL) Commands	62
Deferred Publishing	63
Tuning Buffer Flushes	64
Optimizing Global Cache and Lock Areas	64
Minimizing Communication with Global Areas	66
Optimizing Block Sizes	67
12 Adabas SMF Records	69
Record Structure	70
Record Size Limits	70
Record Subtypes	71
Statistical Recording	72
Record Sections	72
ASMFREC Mapping Macro	89
SMF User Exit	90
IBM Type 89 SMF Records	91
13 Adabas Online System Cluster Environment Screens	93
Scrolling through Nucleus IDs in a Cluster	94
Displaying Cluster Members	94
Nucleus File Status	96
Nucleus Status Flags	97
Cluster Usage	100
Estimating Sizes for the Cache Structure in a Cluster Environment	107
Estimating Sizes for the Lock Structure in a Cluster Environment	111
Maintain the User Table	114
Index	117

Preface

This document provides information about initialization, termination, backout, restart, and recovery processes in an Adabas Parallel Services cluster environment. It tells you how to plan an outage and how to use the utility functions that are provided specifically for cluster environments. Finally, it provides a sample of the session-end statistics produced for a clustered nucleus and describes how to improve performance and tune Adabas Parallel Services.

This document is organized as follows:

<i>Initialization</i>	Describes topics related to Adabas Parallel Services initialization.
<i>Termination</i>	Describes topics related to Adabas Parallel Services termination.
<i>Using Alert and Timeout Settings</i>	Describes alert and timeout settings that can be specified to help you prevent or handle critical situations where system problems affect the availability of cluster members.
<i>Collaboration Between Cluster Nuclei</i>	Describes enhancements for making the nuclei in a cluster more independent of one another.
<i>Backout Processing</i>	Describes Adabas Parallel Services backout processing.
<i>Restart/Recovery Processing</i>	Describes Adabas Parallel Services restart and recovery processing.
<i>Utility Processing</i>	Describes Adabas Utilities processing pertinent to Adabas Parallel Services.
<i>Cluster Nucleus Session End Statistics</i>	Describes the statistics collected for a cluster nucleus session.
<i>Performance and Tuning</i>	Describes performance and tuning you can do for Adabas Parallel Services.
<i>Adabas SMF Records</i>	Describes Adabas SMF records and how they can be produced.
<i>Adabas Online System Cluster Environment Screens</i>	Describes the Adabas Online System version 7.4 screens that apply to the Adabas Parallel Services cluster environment.

1 Conventions

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*.

2 About this Documentation

- Document Conventions 4
- Online Information and Support 4
- Data Protection 5

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <code>folder.subfolder.service</code> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

Product Training

You can find helpful product training material on our Learning Portal at <https://learn.software-ag.com>.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

3 Initialization

- Sequence 8
- ADACOM Initialization Process 8
- Adabas Parallel Services Cluster Nucleus Process 9
- PPT Processing 10

This chapter describes Adabas Parallel Services initialization.

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 `PLI nnn` messages.

All `PLI nnn` 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 cache 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.

4 Termination

- Normal Termination 14
- Abnormal Termination 14

This chapter describes Adabas Parallel Services termination.

Normal Termination

This section covers the following topics:

- [ADACOM](#)
- [Cluster Nuclei](#)

ADACOM

ADACOM must stay in operation in an active Adabas Parallel Services cluster environment. ADACOM will not terminate normally if any nucleus that it manages is still active.

When all nuclei that it manages have terminated, you can terminate ADACOM using an ADAEND command. See the section *Cluster Operator Commands* in the *Adabas Parallel Services Reference Guide*.

Cluster Nuclei

If the Adabas operator command ADAEND or HALT is issued, the nucleus stops with no pending autorestart. The other active nuclei in the cluster continue processing normally.

```
ADAN51 00006 2002-06-25 18:03:29 Operator type-in: ADAEND
ADAN42 00006 2002-06-25 18:03:29 Function accepted
ADAX2B 00006 TT-1, SMM terminating
PLX087 00006 Attempting to delete dataspaces
PLX092 00006 Dataspaces deleted
ADAM97 06001 This ASCB/Initiator will be terminated by MVS at E0J
```

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

```
ADAX5B 00006 Disconnecting from S64 cache
```

Abnormal Termination

This section covers the following topics:

- [ADACOM](#)

- Cluster Nuclei

ADACOM

If ADACOM terminates abnormally, a `PLInnn` error message is produced to explain the problem. All active Parallel Services nuclei in clusters managed by this instance of ADACOM will also abend. ADACOM is the owner of the cache and lock data spaces used by the cluster nuclei. If ADACOM goes away, the data spaces also go away, and the nuclei will abend when they attempt to continue accessing the data spaces.



Note: This is not true of Adabas Cluster Services, where the cache and lock structures exist until all connectors (Adabas nuclei) terminate.

Cluster Nuclei

When an Adabas Parallel Services cluster nucleus terminates abnormally, each surviving peer nucleus performs online recovery. Read [Restart/Recovery](#) elsewhere in this guide for more information.

5 Using Alert and Timeout Settings

- Using Cancel Alerts 18
- Using Self-Termination Alerts 19
- Using the Self-Termination Operator Query 20
- Using Messaging Performance Statistics 21

In a cluster environment, the Adabas nuclei working on the same cluster database must collaborate to keep the database physically and logically consistent while processing user commands. To do this, they issue intracluster commands to one another. If one cluster member does not respond to an intracluster command from another cluster member within a specified time period, the sending member cancels the unresponsive member.

Adabas Parallel Services provides alert and timeout settings that are designed to help you prevent or handle critical situations where system problems might cause the prolonged unavailability of one cluster member, thus endangering the ability of the cluster member (or in severe cases, the entire cluster) to provide service. These settings include:

- A cancel alert that generates an operator message can be invoked when a cluster member is unresponsive to an intracluster command for a specified period of time. If the cluster member does not respond before the message times out, it is canceled.
- Self-termination alerts that generate operator messages can be invoked when a canceled cluster member does not terminate as requested (in a specified period of time) and the other cluster members prepare to self-terminate.
- An operator query that prints an operator console message and requests a response from the operator. This can be invoked by cluster members that are preparing to self-terminate because a canceled peer member remains active.
- Messaging statistics provide information about the performance of message transmission events. These statistics can be used to determine the impact of messaging on system performance and to determine how to set the ADARUN parameters related to the alert and timeout settings.

Using Cancel Alerts

In cluster environments, the cluster members issue intracluster commands to one another to ensure that the database is kept physically and logically consistent. If one cluster member does not respond to an intracluster command from another cluster member within the time specified by the ADARUN `MXMSG` parameter, the sending member cancels the unresponsive member.

You can invoke a cancel alert before the unresponsive peer member is canceled. This alert generates an operator message that provides early warning information before the unresponsive cluster member is canceled.

The cancel alert is governed by the setting of the ADARUN `MXMSGWARN` parameter. This optional parameter specifies the number of seconds after which a cluster nucleus should generate an operator message warning about an outstanding intracluster response. If the cluster member does not respond within the time specified by ADARUN `MXMSGWARN`, message ADAX9C is issued. This warning message can be used to notify you sometime before the unresponsive cluster member is canceled.

As complements to the ADARUN `MXMSG` and `MXMSGWARN` parameters, two operator commands, `MXMSG` and `MXMSGWARN`, are provided that allow you to change the corresponding ADARUN settings dynamically, while the database is running. In addition, the `DPARM` operator command's output information includes the settings of the ADARUN parameters related to cluster alert and timeout enhancements.

For more information about the `MXMSG` and `MXMSGWARN` parameters, read *MXMSG: Timeout Threshold for Internucleus Command Processing* and *MXMSGWARN: Timeout Threshold for Internucleus Command Processing Warning*, in the *Adabas Parallel Services Reference Guide*. For information about the `DPARM`, `MXMSG`, and `MXMSGWARN` operator commands, read *Cluster Operator Commands*, in the *Adabas Parallel Services Reference Guide*.

Using Self-Termination Alerts

In cluster environments, if one cluster nucleus has issued a cancellation request for a second unresponsive cluster nucleus, but the canceled peer cluster nucleus does not terminate within the time specified by the ADARUN `MXCANCEL` parameter, the sending nucleus will either return response code 124, subcode 28 (if the intracluster communication occurred on behalf of an Adabas command) or terminate itself abnormally (if the intracluster communication occurred on behalf of an internal process that must not fail).

You can invoke a self-termination alert before a nucleus terminates itself because a canceled peer nucleus fails to terminate. This alert generates an operator message that provides early warning information regarding the pending self-termination.

Self-termination alerts are governed by the setting of the new ADARUN `MXCANCELWARN` parameter. This optional parameter specifies the number of seconds after which a requesting cluster nucleus should generate an operator message warning about the inability of a canceled peer nucleus to terminate quickly. If the canceled peer nucleus does not terminate within the time specified by ADARUN `MXCANCELWARN`, message ADAX9G is issued. This warning message can be used to notify you that the nucleus issuing the message is in danger of terminating itself.

As complements to the ADARUN `MXCANCEL` and `MXCANCELWARN` parameters, two operator commands, `MXCANCEL` and `MXCANCELWARN`, are provided that allow you to change the corresponding ADARUN settings dynamically, while the database is running. In addition, the `DPARM` operator command's output information has been enhanced to include the settings of ADARUN parameters related to Adabas Parallel Services alert and timeout enhancements.

For more information about the `MXCANCEL` and `MXCANCELWARN` parameters, read *MXCANCEL: Timeout Threshold for Canceled Peer Nucleus* and *MXCANCELWARN: Timeout Threshold for Canceled Peer Nucleus Warning*, in the *Adabas Parallel Services Reference Guide*. For information about the `DPARM`, `MXCANCEL`, and `MXCANCELWARN` operator commands, read *Cluster Operator Commands*, in the *Adabas Parallel Services Reference Guide*.

Using the Self-Termination Operator Query

You can invoke an operator query when a cluster member is in the process of self-terminating because a canceled peer nucleus fails to terminate. This gives you a chance to terminate the canceled cluster member manually, thus avoid the self-termination of the member that issued the ineffective cancel request.

This operator query prints a console message (message ADAX9J) explaining the situation and requesting instructions, waiting for a specified time for a response. The valid responses to message ADAX9J are:

- R (print the ADAX9J message again and continue to wait for resolution of this issue, but without setting a new wait period for the response)
- T (terminate the querying nucleus with message ADAX99 and user abend 79)
- W (continue to wait for another time period of length MXWTOR)

The amount of time the operator query waits for a response is governed by the setting of the ADARUN MXWTOR parameter. This optional parameter specifies the number of seconds the nucleus should wait for the operator response. If the operator does not respond in this time frame and if the canceled peer nucleus still has not terminated, the requesting nucleus issues message ADAX99 and terminates itself.

However, if the canceled cluster member terminates after all (whether due to operator intervention or another reason), the cluster nucleus that issued the operator query stays alive; it retracts the query and initiates an online recovery process.

As a complement to the ADARUN MXWTOR parameter, an operator command, MXWTOR, is provided that allows you to change the MXWTOR setting dynamically, while the database is running. In addition, the DPARM operator command's output information includes the settings of ADARUN parameters related to Adabas Cluster Services alert and timeout enhancements.

For more information about the MXWTOR parameter, read *MXWTOR : Self-Termination Operator Query Interval*, in the *Adabas Parallel Services Reference Guide*. For information about the DPARM and MXWTOR operator commands, read *Cluster Operator Commands*, in the *Adabas Parallel Services Reference Guide*.

Using Messaging Performance Statistics

Adabas Parallel Services messaging statistics provide information about the performance of message transmission events. These statistics can be used to determine the impact of messaging on system performance and to determine how to set the ADARUN MXMSG and MXMSGWARN parameters related to the other alert and timeout enhancements in Adabas Parallel Services.

The performance statistics are provided in the termination statistics of an Adabas nucleus as well as in response to the DXMSG operator command. The performance statistics are split into those that are subject to the ADARUN MXMSG parameter setting and those that are not; after each is reported separately in the output, a combined report is provided containing the summarization of the two for all messages.

For more information about the DXMSG operator command, read *Cluster Operator Commands*, in the *Adabas Parallel Services Reference Guide*.

6 Collaboration Between Cluster Nuclei

- Buffer Flush Independence 24
- Update Command Synchronization 25

The Adabas nuclei in a cluster frequently need to collaborate with one another to keep the database consistent while each of them reads and updates data in the database. The nuclei share recently used data in the global cache area, synchronize their operations using locks in the global lock area, and send messages to one another to perform joint actions that multiple or all nuclei must be informed about or participate in.

If a system hosting a cluster nucleus is temporarily unable to let that nucleus execute normally, for example because it is overloaded, a collaboration process in the cluster may stall if the nucleus cannot respond to intracluster messages in a timely fashion. In such cases, one unhealthy nucleus may impact the processing performed by the other nuclei in the cluster, an effect known as “sympathy sickness.” Adabas Parallel Services has configuration options to eliminate this negative effect for two important collaboration processes.

Buffer Flush Independence

When cluster nuclei process update commands, they store the updated ASSO and DATA blocks in the global cache area, either immediately (if the LRDP parameter equals zero) or by the end of the associated transaction (with **deferred publishing**, LRDP>0). Later, a buffer flush writes all updated blocks from the global cache to the database. Any nucleus in the cluster may perform a buffer flush when needed, but only one at a time.

Before a buffer flush writes updated ASSO and DATA blocks to the database, all protection data for the updates in those blocks must be written to the WORK data sets, so that at all times, in the event that one or more of the nuclei fail (terminate abnormally), the following session autorestart can use that protection data to back out any incomplete updates. In earlier versions of Adabas Parallel Services, when one nucleus performed a buffer flush, every other nucleus in the cluster needed to collaborate by writing its latest protection data to its WORK data set, in order to ensure the protection data is available if the nucleus fails during or after the buffer flush. If one nucleus was slow or unable to respond to the request from the buffer flush, the flush would stall. Without a functioning buffer flush, updated blocks waiting to be flushed out would over time fill up the global cache and then all update processing would stall, too.

Adabas Parallel Services Version 8.4 introduced the global parameter CLUPUBLPROT, which allows a buffer flush to complete even without the collaboration of the other nuclei in the cluster. When CLUPUBLPROT is set to YES, each cluster nucleus “publishes” its protection data before it writes related updated ASSO or DATA blocks to the global cache. A nucleus “publishes” protection data by writing it either to the WORK data set or to the global cache as well. This makes the protection data available to the other nuclei in the cluster. Then, if the nucleus is slow or unable to collaborate with a buffer flush performed by a peer nucleus, the buffer flush can read the latest protection data of the unresponsive nucleus from the cache and write it to WORK by itself. This way, the buffer flush is independent of the ability of the other nuclei to collaborate and can complete even without collaboration.

When CLUPUBLPROT is set to YES, cluster nuclei publish their protection data either in the global cache or on the WORK data set before they publish related updated ASSO or DATA blocks in the cache. There is a performance aspect to consider. All protection data must eventually be written to WORK, so writing it to the cache first is overhead. On the other hand, writing data to the cache area is much faster than an I/O to disk; issuing a WORK I/O every time updated ASSO or DATA blocks are to be written to the cache can slow down the processing of update commands significantly.

The parameter CLUWORK1CACHE specifies how many different WORK blocks a cluster nucleus may keep in the global cache at the same time (when CLUPUBLPROT=YES). CLUWORK1CACHE limits the amount of cache space used for protection data and implicitly regulates the use of cache writes versus WORK I/Os for publishing protection data. If CLUWORK1CACHE is specified as zero, protection data is only written to the WORK data set and never to the global cache. This may lead to a significant number of additional WORK writes and is not recommended. A large number for CLUWORK1CACHE may lead to a significant number of additional cache writes for WORK blocks. A number in-between may lead to some additional WORK writes and some additional cache writes. Finding the best balance for overall performance may require trials with different CLUWORK1CACHE settings.

In summary, specifying CLUPUBLPROT=YES (with CLUWORK1CACHE set appropriately) is recommended to make the nuclei in a cluster more independent of one another when they perform buffer flushes.

Update Command Synchronization

If an Adabas nucleus fails (terminates abnormally), the following session autorestart recovers from the failure by undoing incomplete updates or transactions that may have been written to the database before the failure and redoing complete transactions that may not have been written to the database. This happens either during an online recovery process triggered by a nucleus failure in a cluster (if one or more nuclei in the cluster are still active) or when the next nucleus starts (if no nucleus stayed active).

Historically, the session autorestart logic relied on, and took advantage of, the presence of a point in time before the nucleus failure where none of the nuclei in the cluster was processing any update commands. During normal processing, the nuclei in a cluster took care to create such points in time by delaying the selection of new update commands for processing until all active update commands have finished. They did this regularly after every buffer flush. The process to get to a point where no update command is in progress in the entire cluster (called “update command synchronization”) requires the collaboration of all nuclei.

Adabas Version 8.4 introduced the global parameter UPDATECONTROL, which offers the choice to do without these update command synchronization processes. When UPDATECONTROL is set to NODELAY, the nuclei in the cluster do not delay the start of new update commands after buffer flushes. In the case of a nucleus failure, the following session autorestart does no longer

rely on the presence of points in time where no update processing was in progress. This eliminates the regular update command synchronization processes in which all nuclei in the cluster must collaborate.

In summary, specifying `UPDATECONTROL=NODELAY` (in conjunction with `INDEXUPDATE=ADVANCED`, which is a prerequisite) is recommended to make the nuclei in a cluster more independent of one another after each buffer flush.

7 Backout Processing

Normal backout processing includes:

- BT command processing;
- backing out an update command that received a nonzero response code; and
- internal transaction backout due to, for example, a timeout.

Cluster nuclei perform normal Adabas backout processing. Each nucleus reads the protection data needed to back out updates from its own Work data set/file. It does not need to read protection data from the other Work data sets/files in the cluster.

8 Restart/Recovery Processing

- Offline Recovery (Session Autorestart) 30
- Online Recovery 30
- Automatic Restart Management (ARM) 31
- Archive Recovery 31

Restart/recovery occurs if a cluster nucleus fails. Restart/recovery uses the Work data sets/files of all nuclei to recover the database. The Work data sets/files are dynamically allocated from the data set names recorded in the PPT. Adabas Parallel Services version supports offline and online recovery.

Offline Recovery (Session Autorestart)

- If a cluster nucleus session terminates abnormally, start one of the cluster nuclei to perform the autorestart.
- If a noncluster nucleus session terminates abnormally, restart the noncluster nucleus to perform the autorestart.

Offline recovery occurs if all active cluster nuclei in an Adabas Parallel Services cluster fail. Offline recovery relies only on information from the physical database and the Work data sets/files of each cluster nucleus. All information in the global cache and lock areas is lost.

The first cluster nucleus to restart repairs any physical inconsistencies in the database and backs out all incomplete commands and transactions. The restarted nucleus obtains recovery information from blocks in the common database and from the Work data sets/files of all the failed nuclei.

The restarting nucleus retrieves the Work data set/file names from the PPT block for each terminated nucleus and opens these data sets/files using dynamic allocation. From that point, normal recovery processing occurs:

- the breakpoint on each Work data set/file is found;
- backward and forward repair is performed; and
- autobackout is performed.

While reading through the Work data sets/files, the restarting nucleus on the fly merges the protection records by their timestamps into chronological sequence.

Online Recovery

When one or more cluster nuclei have failed while one or more other nuclei in the same cluster remain active, online recovery processing is performed by collaboration of all surviving nuclei.

All surviving cluster nuclei quiesce their operations and reinitialize their working storage. Command processing is quiesced and the internal status variables, tables, and pools are repaired.

The peer nuclei compete for the recovery lock: when one of the nuclei obtains it, it invokes offline recovery processing. It repairs any physical inconsistencies in the database and backs out all incomplete command and transactions. Open transactions executed by the surviving nuclei are

backed out as well. All information in the global lock and cache areas is discarded. Please keep in mind that the nucleus performing online recovery may need additional LWP for the online recovery procedures depending on other ADARUN parameters such as NU.

Once this recovery processing has completed, normal processing resumes.

Users are affected by online recovery as follows:

- users assigned to failed nuclei lose their commands, transactions, sequential processes, and search results. They may receive response codes 9, 21, 148, or 251, depending on the status of their session at the time of the failure.
- users assigned to surviving nuclei may or may not lose their commands/transactions, depending on whether they managed to complete them in the quiesce phase. They retain their sequential processes and search results, but they may experience an increased response time. Users that do lose their commands/transactions will subsequently receive response code 9 and might possibly get response code 21 as well.

Automatic Restart Management (ARM)

Automatic restart management (ARM) is a z/OS facility that can be used to automatically restart a nucleus when it abends. Automatic restart is suppressed when the ABEND is intentional; for example, when it results from a parameter error.

ARM can be used for Adabas nuclei in both cluster and noncluster environments.

The ADARUN parameter `ARMNAME` (read *ADARUN Parameter Usage in Cluster Environments* in the *Adabas Parallel Services Reference Guide*) is used to identify the element in the ARM *policy* that is to be activated. Each element specifies when, where, and how often an automatic restart is to be attempted.

If an ARM policy has not been defined, the `ARMNAME` parameter has no effect.

Archive Recovery

Archive recovery occurs if the container data sets of the database are damaged or restart recovery is not effective.

Archive recovery:

- restores the database; and
- regenerates the updates from the protection logs.

The protection logs to be regenerated are the output of the ADARES PLCOPY protection log copy and merge process that occurs in Adabas Parallel Services cluster environments. The restore/regenerate process is the same in both cluster and noncluster environments.

9 Utility Processing

▪ ADACHK PTPRINT - Print/Dump Parallel Participant Table	34
▪ ADADBS OPERCOM Commands	34
▪ ADADBS REFRESHSTATS - Refresh Statistical Values	35
▪ ADAPLP IPLOGPRI - Print Sequential Intermediate Data Sets	36
▪ ADARAI - Adabas Recovery Aid	36
▪ ADAREP - Checkpoint Information Extended	36
▪ ADARES CLCOPY - Copy/Merge Nucleus Cluster Command Logs	36
▪ ADARES PLCOPY - Copy/Merge Nucleus Cluster Protection Logs	38
▪ ADARES MERGE CLOG - Merge Nucleus Cluster Command Logs	44
▪ ADARES BACKOUT and REGENERATE— Uniquely Identifying Checkpoints	45
▪ ADASAV Processing Change	46
▪ ADASAV RESTPLOG -- Uniquely Identifying Checkpoints	47

Read your *Adabas Utilities* documentation for specific information about changes to utilities for use in an Adabas cluster environment.

ADACHK PTPRINT - Print/Dump Parallel Participant Table

The PTPRINT function is included in the Adabas ADACHK utility to support 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.

ADADBS OPERCOM Commands

Changes have been made for ADADBS OPERCOM command processing in an Adabas Parallel Services cluster nucleus environment.

This section covers the following topics:

- [Global Commands](#)
- [Routing a Command to a Specific Nucleus](#)
- [Routing a Command to All Cluster Nuclei](#)

Global Commands

The following ADADBS OPERCOM commands have a GLOBAL option for making the request to all nuclei in an Adabas cluster:

- ADAEND
- CANCEL
- FEOFCL
- FEOFPL
- HALT

For example:

```
ADADBS OPERCOM ADAEND,GLOBAL
```

When GLOBAL is specified, the command is automatically propagated to all active cluster nuclei. 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 function's NUCID parameter.

The following example sends the DSTAT command to the Adabas cluster nucleus designated with NUCID=3:

```
ADADBS OPERCOM DSTAT,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 Adabas 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.

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/link name DD/PLOG.

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

Sample JCL has been added for allocating the intermediate data sets/files MERGIN1 and MERGIN2 required for automated CLOG copy/merge processing in nucleus cluster environments.



Notes:

1. When intermediate data sets/files 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.

3. If ADARUN CLOGMRG=NO is specified, at the end of an ADARES CLCOPY job no additional invocations of the ADARES CLCOPY will occur. The additional invocations only occur in an ADARUN CLOGMRG=YES environment.

z/OS

```
//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)
/*
```

z/VSE

```
//JOB ALLOC
/* Example to allocate the ADARES CLCOPY intermediate data sets
//EXEC PGM=IDCAMS
DEFINE CLUSTER (NAME (MERGIN1.CLCOPY) -
VOLUME(xxxxxx) CYLINDERS(3,10) RECSZ(23472) -
/*
//EXEC PGM=IDCAMS
DEFINE CLUSTER (NAME (MERGIN2.CLCOPY) -
VOLUME(xxxxxx) CYLINDERS(3,10) RECSZ(23472) -
/*
/&
```

ADARES PLCOPY - Copy/Merge Nucleus Cluster Protection Logs

In an Adabas nucleus cluster environment, the protection logs (and optionally, the command logs: see the ADARES MERGE CLOG function and the ADARUN CLOGMRG parameter) 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.

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/files 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/files. 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.

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.

A sample user exit (USEREX2P for dual logs or UEX12 for multiple logs) is provided to illustrate the necessary change for the intermediate data set.



Caution: Sample user exits and programs are not supported under any maintenance contract agreement.

- [z/OS Sample Jobs](#)
- [z/VSE Sample Jobs](#)

z/OS Sample Jobs

This section describes the sample jobs provided for z/OS systems.

ALLOC Job

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)
/*
```

ADARESPM Job

A sample job **ADARESPM** is provided on the **JOBS** 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 cross-checks 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 **DDPLOGRn** statements 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.

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
//MARGIN1 DD DISP=SHR,DSN=EXAMPLE.PLOG.MARGIN1
//MARGIN2 DD DISP=SHR,DSN=EXAMPLE.PLOG.MARGIN2
//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 or is otherwise unavailable. It specifies that the PLOG data sets of all cluster nuclei are being supplied with DD names DDPLOGnn in the JCL.



Caution: 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.



Caution: 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
//*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDATAR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.DATAR1 <=== DATA
//DDPLOG01 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1.NUC1 <=== PLOG1 NUC1
//DDPLOG02 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR2.NUC1 <=== PLOG2 NUC1
//DDPLOG03 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1.NUC2 <=== PLOG1 NUC2
//DDPLOG04 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR2.NUC2 <=== PLOG2 NUC2
//DDPLOG05 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR1.NUC3 <=== PLOG1 NUC3
//DDPLOG06 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.PLOGR2.NUC3 <=== PLOG2 NUC3
//MARGIN1 DD DISP=SHR,DSN=EXAMPLE.PLOG.MARGIN1 <=== INTERMEDIATE 1
//MARGIN2 DD DISP=SHR,DSN=EXAMPLE.PLOG.MARGIN2 <=== INTERMEDIATE 2
//DDZIAUS1 DD DSN=EXAMPLE.DByyyyy.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
/*
//

```

z/VSE Sample Jobs

This section describes the sample jobs provided for z/VSE systems.

ALLOC Job

The following sample JCL is provided for allocating the intermediate data sets MARGIN1 and MARGIN2 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.

- The data set block size used must be greater than or equal to the largest PLOG block size plus eight. The record length must be set to the block size minus four.

```
// JOB MRGFILE
// ASSGN SYS004,SYSIPT
// ASSGN SYS005,DISK,VOL=VSE209,SHR
// DLBL UOUT,'QA.DB1010.MERGIN1',0
// EXTENT SYS005,VSE209,1,0,10575,300
// EXEC PGM=OBJMAINT
./ CARD DLM=$$
./ COPY
$$
/*
// ASSGN SYS004,SYSIPT
// ASSGN SYS005,DISK,VOL=VSE209,SHR
// DLBL UOUT,'QA.DB1010.MERGIN2',0
// EXTENT SYS005,VSE209,1,0,11625,300
// EXEC PGM=OBJMAINT
./ CARD DLM=$$
./ COPY
$$
/*
/&
```

ADARESPM Job

A sample job ADARESPM is provided 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 cross-checks 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 PLOGRn statements 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 PLOGRn statements are supplied, they are ignored.

The following sample JCL illustrates the ADARES PLCOPY merge function:

```
// JOB ADARESPM
/*
/* ADARES : COPY/MERGE DUAL/MULTIPLE PROTECTION LOG
/* TWO COPIES OF OUTPUT ARE TO BE CREATED
/* FOR USE WITH AN ADABAS NUCLEUS CLUSTER
/*
// LIBDEF PHASE,SEARCH=(SAGLIB.ASM742,SAGLIB.ADA742)
// DLBL ASSOR1,'EXAMPLE.DByyyy.ASSOR1'
// DLBL DATAR1,'EXAMPLE.DByyyy.DATAR1'
// DLBL MERGIN1,'MERGIN1.PLCOPY'
// DLBL MERGIN2,'MERGIN2.PLCOPY'
```

```
// TLBL SIAUS1,'EXAMPLE.DByyyyy.PLOG1'
// TLBL SIAUS2,'EXAMPLE.DByyyyy.PLOG2'
// ASSGN SYS021,TAPE
// EXEC ADARUN,SIZE=ADARUN
PROG=ADARES,MODE=MULTI,SVC=svc,DEVICE=3380,DBID=yyy
yy
/*
ADARES PLCOPY TWOCOPIES
/*
/&
```

ADARES PLCOPY NOPPT—Ignore PPT

NOPPT is for emergency use when the PPT has been overwritten or is otherwise unavailable. It specifies that the PPT is to be ignored and that the PLOG data sets of all cluster nuclei are being supplied with link names PLOGnn in the JCL.

 **Caution:** 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 link names PLOG01-nn.

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

ADARESIP Job

The following sample job ADARESIP is used with the ADARES PLCOPY NOPPT function:

```
// JOB ADARESIP
/*
/* ADARES : COPY/MERGE DUAL/MULTIPLE PROTECTION LOG
/* FOR USE WITH AN ADABAS NUCLEUS CLUSTER
/* CAUTION: NOPPT EXECUTION!
// LIBDEF PHASE,SEARCH=(SAGLIB.ASM742,SAGLIB.ADA742)
// DLBL ASSOR1,'EXAMPLE.DByyyyy.ASSOR1'
// DLBL DATAR1,'EXAMPLE.DByyyyy.DATAR1'
// DLBL PLOG01,'EXAMPLE.DByyyyy.PLOGR1.NUC1'
// DLBL PLOG02,'EXAMPLE.DByyyyy.PLOGR2.NUC1'
// DLBL PLOG03,'EXAMPLE.DByyyyy.PLOGR1.NUC2'
// DLBL PLOG04,'EXAMPLE.DByyyyy.PLOGR2.NUC2'
// DLBL PLOG05,'EXAMPLE.DByyyyy.PLOGR1.NUC3'
// DLBL PLOG06,'EXAMPLE.DByyyyy.PLOGR2.NUC3'
// DLBL MERGIN1,'MERGIN1.PLCOPY'
// DLBL MERGIN2,'MERGIN2.PLCOPY'
// TLBL SIAUS1,'EXAMPLE.DByyyyy.PLOG1'
// ASSGN SYS021,TAPE
```

```
// EXEC ADARUN,SIZE=ADARUN
PROG=ADARES,MODE=MULTI,SVC=svc,DEVICE=3380,DBID=yyyyy
/*
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/files to be included in the merge process. The maximum number is 32.

Sequential data sets/files are expected as input to the MERGE CLOG function; therefore, the ADARES CLCOPY function must be executed prior to the ADARES MERGE function.

The timestamp contained in the CLOGLAYOUT=8 format of the CLOG is required for the proper merging of command logs records.

z/OS

The following sample job ADADESCM for z/OS systems (see the JOBS data set) illustrates the execution of the ADARES MERGE CLOG function:

```
//ADADESCM 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
/*
//DDASSOR1 DD DISP=SHR,DSN=EXAMPLE.DByyyyy.ASSOR1 <=== ASSO
//DDATAR1 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
/*
//

```

z/VSE

The following sample job for z/VSE systems (see the JOBS data set) illustrates the execution of the ADARES MERGE CLOG function:

```

//JOB ADARESCM
/*
/* ADARES : MERGE SEQUENTIAL COMMAND LOGS
/* FOR USE WITH AN ADABAS NUCLEUS CLUSTER
/*
// LIBDEF PHASE,SEARCH=(SAGLIB.ASM742,SAGLIB.ADA742)
// DLBL ASSOR1,'EXAMPLE.DByyyyy.ASSOR1'
// DLBL DATAR1,'EXAMPLE.DByyyyy.DATAR1'
// DLBL WORKR1,'EXAMPLE.DByyyyy.WORKR1'
// DLBL ASSOR1,'EXAMPLE.DByyyyy.ASSOR1'
// DLBL CLOG01,'EXAMPLE.DByyyyy.CLOGR1.NUC1'
// DLBL CLOG02,'EXAMPLE.DByyyyy.CLOGR1.NUC2'
// DLBL CLOG03,'EXAMPLE.DByyyyy.CLOGR2.NUC3'
// TLBL SIAUS1,'EXAMPLE.DByyyyy.CLOGM'
// ASSGN SYS021,TAPE
// EXEC ADARUN,SIZE=ADARUN
ADARUN PROG=ADARES,MODE=MULTI,SVC=svc,DEVICE=3380,DBID=YYYYY
/*
ADARES MERGE CLOG,NUMLOG=3
/*
/&

```

ADARES BACKOUT and REGENERATE— Uniquely Identifying Checkpoints

The protection log merge process usually changes the numbering of the PLOG blocks. The PLOG block number of a checkpoint on the original PLOG data set will not necessarily be the same as the block number after the merge. 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.

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.



Note: 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`, and `FROMNUCID`.

If the `NUCID` is the same for the starting and ending checkpoint, only the `NUCID` needs to be specified.



Notes:

1. 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.
2. `BACKOUT DPLOG` and `BACKOUT MPLOG` are not allowed for a cluster database. The `PLOG` must be merged before the backout can be performed.

ADASAV Processing Change

Sample JCL is located in the `ADASAVRW` member of the `JOBS` data set.

For the following `ADASAV` functions, 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:

- `RESTONL` (database)
- `RESTONL GCB`
- `RESTORE` (database)
- `RESTORE GCB`

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.

10 Cluster Nucleus Session End Statistics

▪ General Nucleus Information	50
▪ Input/Output Statistics	50
▪ Command Statistics	52
▪ User Sessions	53
▪ Efficiency Statistics	54
▪ Messaging Statistics	55
▪ Global Cache Statistics (Cluster Nucleus Only)	55
▪ Global Lock Statistics (Cluster Nucleus Only)	57
▪ WORK Data Set I/O Statistics (Cluster Nucleus Only)	58
▪ Data Set Activity Statistics	59

In addition to the end-of-session statistics printed by every Adabas nucleus, the statistics for a cluster nucleus also include global cache and lock statistics.

If you are running the selectable unit Adabas Online System (AOS), all of the statistics shown in the following sample output are displayed.

If you are running only the demo version of AOS delivered with Adabas, the statistics displayed are limited as follows:

Section	Displays statistics only for...
Global Cache Statistics	totals, DS, and NI
Global Lock Statistics	buffer flush, hold ISN, new data RABN, and global update command sync locks

General Nucleus Information

```
A d a b a s  n u c l e u s  s e s s i o n  s t a t i s t i c s
From 2014-03-11 10:56:39 to 2014-03-11 14:46:57

Duration           3:50:18
Wait-time          3:50:13
CPU time           0:00:02
```

Input/Output Statistics

I/O Counts (Including Initialization)

Container	Reads	Writes
-----	-----	-----
ASSO	107	72
DATA	27	14
WORK	3	397
PLOG	0	0
CLOG	0	0
-----	-----	-----
Total	137	483

Work and PLOG I/Os and Blocks

PLOG protection blocks	32
PLOG different blocks	24
PLOG protection I/Os	14
WORK1 protection blocks	42
WORK1 different blocks	23
WORK1 protection I/Os	22
WORK1 publishing blocks	21
WORK1 publishing I/Os	5
WORK1 publishing waits	6

- "WORK1 protection blocks" shows the total number of blocks written to WORK Part 1. Some blocks may have been written more than once (with increasing amounts of data).
- "WORK1 different blocks" shows the number of different blocks written to WORK Part 1 - that is, counting each block once if it was written multiple times in a row. This number correlates with the actual amount of protection data written to WORK.
- "WORK1 protection I/Os" shows the number of I/Os performed to write the protection data to WORK Part 1. Where possible, Adabas writes multiple consecutive blocks in a single I/O operation, if sufficient I/O buffers are available (based on the NWORK1BUFFERS parameter).
- The "PLOG protection blocks", "PLOG different blocks" and "PLOG protection I/Os" statistics are similar.
- The "WORK1 publishing" statistics report on extra WORK writes induced by the CLUPUBLPROT=YES setting in a cluster. They show the number of blocks written, the number of I/O operations performed for those writes, and the number of waits incurred for writing protection data to WORK before ASSO and DATA blocks containing related updates could be written to the global cache. They are printed only if at least one of the numbers is nonzero.

Log Reads and Buffer Efficiency

Logical reads	12,778
Buffer efficiency	95.3

Distribution of ASSO/DATA I/Os by Volser Number (Excluding Initialization)

VOLSER	High RABN	Reads	Writes

ASSO:			
SMSZ10	6,286	107	72

ASSO total		107	72
DATA:			
SMSZ10	5,992	27	14

DATA total	27	14

Total	134	86

Command Statistics

Distribution of Commands by Source

Source	Commands

Remote	0
Local	579
Internal	289
Operator	3

Total	871

Distribution of Commands by Thread

Thread	Commands

1	852
2	19
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0

Total	871

Distribution of Commands by File

File	Commands
0	324
1	400
19	147
Total	871

Distribution of Commands by Type

Type	Commands
A1/4	400
CL	12
LF	2
L1/4	147
OP	10
REST	288
UC	12
Total	871

User Sessions

User sessions	<i>nnn</i>
Most calls (<i>nnn</i>)	initiated by: User <i>userid</i> Job <i>jobname</i> SECUID <i>secuid</i>
Most I/Os (<i>nnn</i>)	initiated by: User ' <i>userid</i> ' Job <i>jobname</i> SECUID <i>secuid</i>
Most thread time (<i>hh:mm:ss</i>)	used by: User ' <i>userid</i> ' Job <i>jobname</i>

Efficiency Statistics

Format translations	157
Format overwrites	0
Auto-restarts	0
Throw-backs due to ISN	0
Throw-backs due to space	0

Buffer Flush Information

Buffer flushes	20
Flush phases	20
Blocks flushed	81
Flush I/Os	73
Flush requests:	
Return immediately	0
Return after logical flush	0
Return after entire flush	29
Buffer flush V2 timeouts	4



Note: The "Buffer flush V2 timeouts" statistic is printed only if it is nonzero or the CLUPUBLPROT parameter is set to YES.

Actual High-water Marks for Major Pools (Except the Buffer Pool)

Pool	ADARUN parm	High watermark
AB -POOL	NAB= 300	16896 (1 %)
CQ -POOL	NC = 500	192 (0 %)
DUQ -POOL	LDE= 7000	0 (0 %)
FI -POOL	LFP= 100000	160 (0 %)
HQ -POOL	NH = 8000	5200 (1 %)
Redo-POOL	LRDP= 300000	112 (0 %)
SC -POOL	LCP= 10000	0 (0 %)
TBI -POOL	LI = 4000	0 (0 %)
TBS -POOL	LQ = 30000	0 (0 %)
UQ -POOL	NU = 500	1820 (0 %)
UQF -POOL	NU = 500	216 (0 %)
Wrk1-IO B	NWORK1= 1	1 (100 %)
WORK-POOL	LWP= 500000	73112 (14 %)
Work Part1	LP = 500	6 (1 %)
Work Part2	LWKP2= 200	0 (0 %)
Work Part3	-- = 794	0 (0 %)
XID -POOL	XID=	(0 %)

Messaging Statistics

```

Message Control Block statistics
Allocated                768
Used                    0
Total requests          0

Statistics for ACMD-type messages
Messages sent           0
Messages arrived       0
Messages accepted      0
Replies sent           0

```

Global Cache Statistics (Cluster Nucleus Only)

Publishing requests by type and total cache statistics are presented first. The following is a sample:

```

Update sync      :      20
BT or CL or ET  :     316
Redo threshold  :      0
Full bufferpool:      0
All blocks      :      28
Specific RABN   :      0
File DS blocks  :      0
DSST blocks     :     13
File NI blocks  :      0

Totals:

Cast-out dir    :      40
  Synchronous   :      0
  Asynchronous  :      40

Unlock cast-out:      20
  Synchronous   :      0
  Asynchronous  :      20

Directory reads:      3
  Synchronous   :      0
  Asynchronous  :      3

Reads           :      51
  Synchronous   :      51
  Asynchronous  :      0

```

Cluster Nucleus Session End Statistics

```
In cache      :          0
Not in cache  :          51
Structure full:          0

Writes       :      3,007
Synchronous  :      3,007
Asynchronous :          0

Written      :      3,007
Not written  :          0
Structure full:          0

Validates    :     19,302
Block invalid:          0

Cast-out reads :          81
Synchronous   :          81
Asynchronous  :          0

Deletes      :          6
Timeouts     :          0

Redo processes :          0
```

After this, specific cache statistics in the following format are presented for the Address Converter (AC), Data Storage (DS), the Data Storage Space Table (DSST), File Control Block (FCB), normal index (NI), upper index (UI), WORK blocks (if at least one of the statistics is nonzero), and for every file in the database:

```
Reads        :          4
Synchronous  :          4
Asynchronous :          0

In cache     :          0
Not in cache :          4
Structure full:          0

Writes       :          13
Synchronous  :          13
Asynchronous :          0

Written      :          13
Not written  :          0
Structure full:          0

Validates    :     2,118
Block invalid:          0

Cast-out reads :          13
Synchronous   :          13
Asynchronous  :          0
```

Deletes	:	0
Timeouts	:	0
Redo processes	:	0

Global Lock Statistics (Cluster Nucleus Only)

Lock statistics are displayed in a format similar to the following sample:

Obtains		
Conditional	:	0
Granted	:	0
Rejected	:	0
Unconditional	:	12
Synchronous	:	10
Asynchronous	:	2
Releases	:	12
Synchronous	:	5
Asynchronous	:	7

Statistics are displayed for many of the following kinds of locks, as appropriate for your database session:

- Buffer flush locks;
- Cancel locks;
- Checkpoint locks;
- Container locks;
- DBID target assignment locks;
- Delta Save Facility (DSF) locks;
- ETID locks;
- File/ISN read locks;
- File Space Table (FST) locks;
- File-lock-table locks;
- Format AC/AC1 locks;
- General Control Block (GCB) locks;
- Global update command sync locks;
- Global ET sync locks;
- Hold ISN locks;
- LOB tracker locks;

- New data RABN locks;
- Online save locks;
- Parameter locks;
- PETU table locks;
- Recovery locks;
- Recovery Log (RLOG) locks;
- Replication handshake locks;
- (Re)usable NI space locks;
- (Re)usable UI space locks;
- Security locks;
- Stored Procedures and Triggers (Spats) locks;
- TBWK4 table locks;
- Unique descriptor locks;
- User locks; and
- XIDE locks.

WORK Data Set I/O Statistics (Cluster Nucleus Only)

Internal NUCID :	2
External NUCID :	10002
Reads :	3
Writes :	0
Internal NUCID :	3
External NUCID :	10001
Reads :	1
Writes :	0

These statistics include the I/O operations of a cluster nucleus to all WORK datasets in a cluster for the following purposes:

- Initialization and validation of the access to a WORK dataset
- Writing of protection data for an unresponsive peer nucleus
- Reading of protection data during online and offline recovery (session autorestart)

Any I/Os to the own WORK dataset of the nucleus during normal processing (writing of protection data for updates, reading of protection data for backout operations, writing and reading of search

results) are not included. These are covered by the [Input/Output Statistics](#) displayed near the top of the end-of-session statistics.

Data Set Activity Statistics

14:46:58	ADAI03	00035	DDWORKR1	3 reads
14:46:58	ADAI03	00035	DDWORKR1	397 writes
14:46:58	ADAI03	00035	DDDATAR1	27 reads
14:46:58	ADAI03	00035	DDDATAR1	14 writes
14:46:58	ADAI03	00035	DDASSOR1	107 reads
14:46:58	ADAI03	00035	DDASSOR1	72 writes

11 Performance and Tuning

▪ ADARUN Parameter Settings	62
▪ Allocating Work Data Set Space	62
▪ Using Close (CL) Commands	62
▪ Deferred Publishing	63
▪ Tuning Buffer Flushes	64
▪ Optimizing Global Cache and Lock Areas	64
▪ Minimizing Communication with Global Areas	66
▪ Optimizing Block Sizes	67

ADARUN Parameter Settings

Software AG recommends that you use your existing Adabas ADARUN parameters (or the default values) for each nucleus in an Adabas cluster, and then tune the values after analyzing the performance of the cluster.

Session statistics can be used to determine the best settings for each parameter. The statistics can be displayed using operator commands during the session; the statistics are also printed automatically at the end of a session or in response to an ADADBS REFRESHSTATS command.

For parameters that allot processing resources to the cluster nuclei (such as NU, NH, LP, etc.), Software AG recommends that you set them large enough that each individual cluster nucleus could handle the entire load on the database if the other nuclei were to terminate abnormally.

Allocating Work Data Set Space

Each Adabas cluster nucleus requires its own Work data set to hold its temporary data.

The individual sizes of the different Work parts (1, 2, and 3) as specified by ADARUN parameters such as LP and LWKP2 can be different among the nuclei; however, the overall size of each Work data set must be the same, because the total Work size is stored in the Adabas general control block (GCB). Software AG recommends that you use the same LP and LWKP2 values on each nucleus active for the same database.

For each nucleus, you need to specify shared access to DD/WORKR1. During an offline or online recovery, a nucleus may access the Work data sets belonging to other nuclei in the cluster.

Using Close (CL) Commands

Users are assigned to a nucleus for their entire sessions and should therefore issue Adabas close (CL) commands as appropriate. The close command ends the user's session, making the user eligible for reassignment to another nucleus when the user again issues an Adabas open (OP) command. This allows Adabas Parallel Services to rebalance the workload over the participating nuclei.

Deferred Publishing

Publication of updated blocks to the global cache area is usually deferred until just before the end of the associated transaction. Multiple updates to a block may produce only a single write of the block to the cache rather than a cache write for each update.

The greater the number of database update in parallel transactions, the greater the expected improvement in performance.



Note: Deferred publishing creates an asymmetry between users on the update nucleus, who see uncommitted updates (unless they read with hold), and users on other cluster nuclei, who may or may not see uncommitted updates.

- [Redo Pool](#)
- [ADARUN Parameter LRDP](#)

Redo Pool

Since the write of updated blocks to the cache may fail due to conflicting updates to the same blocks by other nuclei in the cluster, every cluster nucleus must be capable of redoing the updates it has not yet written to the cache. The nucleus maintains information about these updates in the "redo pool".

ADARUN Parameter LRDP

The size of the redo pool is specified by the new ADARUN parameter LRDP. The LRDP parameter is effective only in a cluster nucleus, that is, when a nonzero NUCID is specified.

If LRDP is not specified, the nucleus takes as default the value of the LFIOP parameter. If LRDP is explicitly set to zero, the nucleus writes each update immediately to the cache.

Different nuclei in the same cluster can have different settings of LRDP. It is also possible, although not recommended, to run one nucleus with LRDP=0 and a peer nucleus with LRDP>0.



Note: If one nucleus runs with LRDP=0 and a peer nucleus runs with LRDP>0 and the different cluster nuclei concurrently update the same Data Storage blocks, incorrect DSST entries may be produced. These are reported by ADADCK. Such errors are harmless and do not affect the results of the application programs.

The nucleus reports on the use (high watermark) of the redo pool in a shutdown statistic and in the response to the DRES command from the operator console or from ADADBS OPERCOM.

Tuning Buffer Flushes

When the update load on the database is so high that the buffer flush becomes the bottleneck, you can improve performance by reducing the duration of buffer flushes.

Instead of starting one I/O per volume, a buffer flush can initially start a predetermined number of I/Os on each volume and then start a new one when another I/O on the same volume finishes. This occurs independently on each volume.

The ADARUN parameters `LFIOP` and `FMXIO` (see the *Adabas Operations* documentation for details) can be used to control buffer flushes. The `LFIOP` parameter enables asynchronous buffer flush operation and sets the I/O pool size. The `FMXIO` parameter sets the limit on the number of I/O operations that can be started in parallel by `LFIOP` flush processing.

- [Effect of ASYTVS Parameter Setting](#)
- [Dynamically Modifying the FMXIO Parameter Setting](#)

Effect of ASYTVS Parameter Setting

The meaning of the `FMXIO` parameter is affected by the setting of the `ASYTVS` parameter:

When `ASYTVS=YES` (buffer flushes occur by volume), `FMXIO` specifies the number of I/Os to be started in parallel *on each volume*. The minimum and default number is 1; the maximum number is 16. If you specify a number greater than 16, it is reduced to 16 without returning a message.

When `ASYTVS=NO` (buffer flushes occur in ascending RABN sequence without regard to the distribution of the blocks over volumes), the minimum, default, and maximum values continue to be 1, 60, and 100, respectively.

Dynamically Modifying the FMXIO Parameter Setting

The setting of `FMXIO` can be modified dynamically using the `FMXIO=nn` command from the operator console or the Modify Parameter function of Adabas Online System.

Optimizing Global Cache and Lock Areas

As a user, you must allocate and define sizes that are appropriate to your application needs for the global cache and lock areas.

This section provides guidelines for determining optimal sizes for these areas based on current experience.



Note: There may be sites for which these guidelines are not appropriate.

- [Global Cache Area Size](#)
- [Global Lock Area Size](#)

Global Cache Area Size

The global cache area must be large enough to retain:

- *directory elements* for all blocks that reside in all the buffer pools; and
- enough data elements to keep the changed blocks between buffer flushes (cast-outs).

Directory elements are used to keep track of the cluster members that have a particular block in their buffer pools so that the block can be invalidated should any member modify it.

If the number of directory elements is insufficient, Adabas Parallel Services reuses existing directory elements and invalidates the blocks associated with those directory elements, because they can no longer be tracked. These blocks must then be reread from the database and registered again the next time they are referenced and validated, even though they did not change.

It is generally better to reassign storage for data elements to keep more ASSO and DATA blocks in the global cache area than to define too many directory elements in the global cache area. More data elements than necessary can be used to keep additional blocks to improve the local buffer efficiency.

The number of directory elements need not be greater than the sum of the sizes of all buffer pools divided by the smallest block size in use for ASSO and DATA.

When connecting to the global cache area during startup, the ADAX57 message reports the number of directory elements and data elements. The ADARUN parameters `DIRRATIO` and `ELEMENTRATIO` determine the ratio between the number of directory and data elements.

Global Lock Area Size

All nuclei in a database cluster share the global lock area.

Locks are held for a variety of entities, for example unique descriptor values. These lock types tend to occur with very different frequencies. The amount of lock activity during a session for each lock type is displayed in the shutdown statistics.

It is often the case that ISN locks show the greatest activity. The sum of high-water marks for NH yields an upper limit for the number of ISN locks that were held concurrently during the session.

The global lock manager uses a hash table to allocate and find a specific lock entry.

When the global lock manager receives a lock request (for example, to put an ISN of a file into hold status), it allocates a specific lock entry unless another member of the cluster has already

made a conflicting allocation. A conflicting allocation produces lock contention because another member holds the same lock. Depending on its type, the lock request is then rejected or remains pending, waiting for the associated resource to become available.

The minimum lock structure size can be roughly estimated as:

```
(NU*3 + NH + NT + LDEUQP/15 + MAXFILES*6 + 100) * 400 + LDEUQP*6 + NH*64 + 1,000,000 ↵  
bytes
```

where `MAXFILES` is the maximum number of files in the database (set in `ADADEF` or `ADAORD`) and `NU`, `NH`, `NT`, and `LDEUQP` are the `ADARUN` parameters of the cluster nuclei. The formula in parentheses $(NU*3 + NH + NT + LDEUQP/15 + MAXFILES*6 + 100)$ is used to calculate the minimum number of lock records that the cluster nuclei expect to have available.

Minimizing Communication with Global Areas

Most of the additional processing required for Adabas Parallel Services environments compared to a single Adabas nucleus involves communication with the global areas.

For this reason, optimizing the performance of an Adabas Parallel Services environment means minimizing the need for communication with the global areas. It is also important to keep the time required for each communication as short as possible.

- [Avoiding the Hold Option](#)
- [Reducing Direct Interaction with the Global Cache Area](#)

Avoiding the Hold Option

Lock requests usually depend on application requirements. Under data-sharing, the hold option is more expensive and access with the hold option should be avoided unless records will in fact be updated or must be protected from concurrent updates.

Reducing Direct Interaction with the Global Cache Area

Cache area requests occur when blocks:

- that are referenced do not exist in the local buffer pool;
- exist in the local buffer pool but have become invalid due to concurrent updates by other cluster members or from directory reuse; or
- are updated.

The first and second situation require registering and (re)reading the blocks from the global cache area. This is more expensive than just validating blocks.

The first situation is related to the buffer efficiency in a noncluster environment. In a cluster environment, buffer efficiency represents the combined effect of the local buffer pool and the global cache area. In order to reduce the interaction with the global cache, the local buffer pool (LBP) should not be decreased from what would be used in a noncluster nucleus. A large LBP parameter and the usage of forward index compression are recommended to improve the buffer efficiency in the local buffer pool.

Optimizing Block Sizes

Although earlier versions of Adabas often worked well with large block sizes, the buffer pool manager and forward index compression features introduced with Adabas version 7 make smaller block sizes more attractive, especially in data-sharing mode.

Use the following guidelines when selecting an optimal block size for ASSO and DATA:



Note: Only general recommendations can be given.

1. Avoid 4-byte RABNs

If the database is not extremely large, avoid 4-byte RABNs as this increases the number of AC blocks by 33%. When growth considerations are taken into account, this may require larger block sizes or limit reductions in block size. The same holds true for the maximum compressed record length.

2. Use forward index compression

Forward index compression can significantly reduce the number of index blocks in a database. Apply forward index compression to all frequently accessed files (or to all files, regardless of their frequency of use). Choose the ASSO block size that is as small as possible but large enough to keep the number of index levels down to 3 or 4.

3. Minimize frequently updated descriptors

When files are updated frequently, the number of blocks that are modified and need to be written to the global cache area often depends on the number of descriptors that have been defined and modified during update processing. Support for additional keys whose descriptor values are subject to frequent modifications becomes even more expensive in a data-sharing environment.

12 Adabas SMF Records

▪ Record Structure	70
▪ Record Size Limits	70
▪ Record Subtypes	71
▪ Statistical Recording	72
▪ Record Sections	72
▪ ASMFREC Mapping Macro	89
▪ SMF User Exit	90
▪ IBM Type 89 SMF Records	91

On z/OS, Adabas supports the recording of database activity via SMF records when ADARUN parameter SMF=YES is specified.

SMF recording is supported on z/OS only.

All Adabas SMF records have a common structure, with some sections appearing in all records and others generated according to specific events and parameters specified through ADARUN or operator commands. The ASMFREC macro provides mapping DSECTs for all parts of the SMF record.

Record Structure

Adabas follows the modern convention for SMF record formats. A single record has:

- A standard IBM-type header
- A self-defining section that describes a variable number of detail sections
- A product ID detail section
- User-selected detail sections

Each detail section is described by an eight-byte entry in the self-defining section containing three fields. An entry is also called a *triplet*.

- A 4-byte offset from the beginning of the record to the detail section
- A 2-byte count of the number of instances of the detail section
- A 2-byte length of each detail section instance. If there are no detail section instances of a given type, the triplet is all zeros.

Field ASNumD in the **product ID section** specifies the number of triplets in the self-defining section.

Record Size Limits

SMF records are z/OS V-format records with a system-imposed maximum length of 32,756 bytes. Most Adabas SMF records fit within this limit for most reasonable types of ADARUN nucleus specifications. However, detail sections such as **File Activity** could potentially have several thousand detail section instances.

If the entire set of instances will not fit in the space remaining in the record, Adabas will include only as many as there are room for and write the record. The SMF record is reset by clearing the triplets for all detail sections except the product ID section and then adding as many of the remaining instances as will fit, repeating until all detail sections are processed. Field ASSegNo in the

product ID section will start at 1 and be incremented with each additional record, and field ASSegL will be set to zero for the last (or only) record for an interval or event.

If any detail section is so large that even one instance would cause the record size limit to be exceeded after resetting the SMF record, that detail type is deactivated.

Record Subtypes

The header section field ASSTy identifies a record subtype.

- Subtype 1 (ASStI) - Adabas Nucleus Initialization
- Subtype 2 (ASStT) - Adabas Nucleus Termination
- Subtype 3 (ASStI) - Adabas Interval Statistics
- Subtype 4 (ASStP) - Adabas Parameter Change

Subtype 1 (ASStI) - Adabas Nucleus Initialization

A record of this subtype is generated during nucleus initialization. In addition to the header, self-defining and product ID sections, it contains ADARUN parameter and user sections if these have been selected by SMFDETAIL ADARUN parameter or operator commands.

Subtype 2 (ASStT) - Adabas Nucleus Termination

A record of this subtype is generated during nucleus termination. In addition to the header, self-defining and product ID sections, it contains all detail sections specified by the SMFDETAIL ADARUN parameter or operator commands except for the ADARUN parameter section. Statistics in Adabas detail sections reflect totals for the entire nucleus session.

Subtype 3 (ASStI) - Adabas Interval Statistics

If interval recording has been specified by the SMFINTERVAL ADARUN parameter or operator command, a record of this subtype is generated at the expiration of each interval. In addition to the header, self-defining and product ID sections, it contains all detail sections specified by the SMFDETAIL ADARUN parameter or operator commands except for the ADARUN parameter section. Statistics in Adabas detail sections reflect activity since the previous interval ended, except where noted. This is also called a delta value.

Subtype 4 (ASStP) - Adabas Parameter Change

If the Adabas ADARUN parameter detail section has been specified by the SMFDETAIL ADARUN parameter or operator commands, a record of this subtype is generated whenever an ADARUN parameter value is changed after nucleus initialization. In addition to the header, self-defining, product ID and ADARUN parameter sections, it may also contain a user section if that has been selected by SMFDETAIL ADARUN parameter or operator commands.

Statistical Recording

The nucleus accumulates usage statistics on the resources it uses to accomplish its tasks. These statistics may be recorded at user or system-defined intervals (see ADARUN parameter SMFINTERVAL) and at termination.

Interval recording (Adabas SMF record subtype 3) provides the usage since the last interval ended for each detail section. Adabas SMF record intervals may be synchronized with one of the system-level intervals specified by PARMLIB member SMFPRM_{xx} entries. This allows straightforward analysis of the usage by allowing direct comparison with other record interval data. For example, you can compare the Adabas interval record with RMF data for the same interval to better understand system performance.

Statistics at termination (Adabas SMF record subtype 2) will have cumulative statistics that reflect activity for the entire nucleus session in each specified detail section.

Record Sections

Every Adabas SMF record contains header, self-defining and product ID sections. You can select additional detail sections through the SMFDETAIL ADARUN parameter or operator commands. Each section is mapped by a DSECT generated by the ASMFREC mapping macro.

The following table summarizes the Adabas SMF record sections:

Detail Section Description	ASMFREC Macro or ADARUN Parameter Specification	Self-Defining Section Triplet Label Base	ASMFREC DSECT Name Produced by the ASMFREC Macro
Header and self-defining section	---	---	ASBase
Adabas command activity	CMD	ASTCmd	ASCmd
Adabas global cache activity by block type ¹	CSHB	ASTChB	ASChB
Adabas global cache activity by Adabas file number ¹	CSHF	ASTChF	ASChF

Detail Section Description	ASMFREC Macro or ADARUN Parameter Specification	Self-Defining Section Triplet Label Base	ASMFREC DSECT Name Produced by the ASMFREC Macro
Adabas global cache activity ¹	CSHG	ASTChG	ASChG
Adabas Parallel Services cache activity ²	CSHP	ASTChP	ASChP
Adabas file activity	FILE	ASTFile	ASFile
Adabas global lock activity ¹	LOCK	ASTLok	ASLok
Adabas internucleus messaging control block activity	MSGB	ASTMsgB	ASMsgB
Adabas internucleus messaging counts	MSGC	ASTMsgC	ASMsgC
Adabas internucleus messaging service time histogram	MSGH	ASTMsgH	ASMsgH
ADARUN parameter values	PARM	ASTParm	ASParm
I/O by DD name	IODD	ASTIODD	ASIODD
Product ID ³	ID	ASTPID	ASPID
Session Statistics	SESS	ASTSess	ASSess
Storage pool	STG	ASTStg	ASStg
Thread activity	THRD	ASTThrd	ASThrd
zIIP statistics	ZIIP	ASTzIIP	ASzIIP
User-defined	USER	ASTU _{sr}	user-defined

1. The detail section is available only in cluster environments when either Adabas Cluster Services or Adabas Parallel Services are installed.
2. The detail section is available only in cluster environments when Adabas Parallel Services is installed.
3. The product ID section is always included in every SMF record. It may not be specified in the SMFDETAIL ADARUN parameter or in operator commands.

This section describes the different detail record sections:



Note: The DSECTs provided in the following sections may not be the most current. To see the most current versions of the DSECTs, generate them using the [ASMFREC macro](#).

- [Header Section](#)
- [Self-Defining Section](#)
- [Product ID Section: ID](#)
- [Adabas Command Activity Section: CMD](#)
- [Adabas File Activity Section: FILE](#)
- [Adabas Global Cache Activity by Block Type Section: CSHB](#)
- [Adabas Global Cache Activity by Adabas File Number Section: CSHF](#)
- [Adabas Global Cache Activity Section: CSHG](#)
- [Adabas Global Lock Activity Section: LOCK](#)

- Adabas Internucleus Messaging Control Block Activity Section: MSGB
- Adabas Internucleus Messaging Counts Section: MSGC
- Adabas Internucleus Messaging Service Time Histogram Section: MSGH
- Adabas Parallel Services Cache Activity Section: CSHP
- ADARUN Parameter Value Section: PARM
- I/O by DD Name Section: IODD
- Session Statistics Section: SESS
- Storage Pool Section: STG
- Thread Activity Section: THRD
- zIIP Statistics Section: ZIIP

Header Section

IBM has defined a standard format for the initial part of all SMF records in *z/OS MVS System Management Facilities (SMF)*, IBM document SA22-7630. This section begins every Adabas SMF record.

ASBase	Dsect		Base segment
*			
*			Standard SMF Header
*			
ASRDW	DS	0B14	Record descriptor word
ASLen	DS	B12	Record length
ASSeg	DS	B12	Segment descriptor
ASFlg	DS	B1.8	System indicator flags
ASFStV	Equ	x'40'	Subtypes are valid
ASFV4	Equ	x'10'	MVS/SP V4 and above
ASFV3	Equ	x'08'	MVS/SP V3 and above
ASFV2	Equ	x'04'	MVS/SP V2 and above
ASFVS2	Equ	x'02'	VS2
ASRTy	DS	B11	Adabas record type
ASTme	DS	B14	Time since midnight when record was + moved into SMF buffer in 1/100 sec
ASDte	DS	P14	Date when record was moved into SMF + buffer as 0cyyddF
ASSID	DS	C14	System identifier (SMFPRMxx SID)
ASSSI	DS	C14	Subsystem identifier
ASSty	DS	B12	Subtype
ASStI	Equ	1	Adabas initialization
ASStT	Equ	2	Adabas termination
ASStS	Equ	3	Interval statistics
ASStP	Equ	4	Parameter change
ASStA	Equ	9	Ad hoc record
*			
ASBaseL	Equ	*-ASBase	Length of standard header

Self-Defining Section

The self-defining section follows immediately after the header section. It is part of the header section DSECT.

Each detail section triplet is identified by a base label as shown in the [table at the beginning of this section](#). The base label begins with the prefix specified in the ASMFREC invocation followed by the letter T (for triplet), and then followed by a mnemonic detail section identifier. The base label with suffix O is the offset, with suffix L is the length, and with suffix N is the number of instances.

Here is an example of some triplets.

```

*
*                               Self-Defining Section
*
ASSDS   DS   0B                Self-defining section
*                               Map of typical section triplet
ASSDSO  DS   B14              Offset to section from start of  +
                               record
ASSDSL  DS   B12              Length of section
ASSDSN  DS   B12              Number of section(s)
                               Org   ASSDS
*
ASTID   DS   0B18             ID Section (always present)
ASTIDO  DS   B14              Offset to ID section from start  +
                               of record
ASTIDL  DS   B12              Length of ID section
ASTIDN  DS   B12              Number of ID section(s)
*
ASTUser DS   0B18             User-Defined Section
ASTUser0 DS  B14              Offset to User-Defined section from+
                               start of record
ASTUserL DS  B12              Length of User-Defined section
ASTUserN DS  B12              Number of User-Defined section(s)
*
ASTParm DS   0B18             ADARUN Parameter Section
ASTParm0 DS  B14              Offset to detail section from start+
                               of record
ASTParmL DS  B12              Length of each detail section
ASTParmN DS  B12              Number of detail section(s)
*
. . .
*
ASSDSLn Equ  *-ASSDS          Length of self-defining section
ASSDSNT Equ  ASSDSLn/8        Number of triplets

```

Product ID Section: ID

The product ID section is always present in every Adabas SMF record with one instance. It describes the nucleus generating the SMF record and provides information about the record's contents.

The 2-byte version code consists of a major version and a minor version. A change, such as adding a new triplet or extending a detail section, will increment the minor version. All existing programs should continue to operate as no existing displacements have changed. A more disruptive change will increment the major version and require existing programs to (at least) be reassembled.

ASPID	Dsect		Product ID Detail Section	+
			(always present in SMF record)	
ASSMFV	DS	0B12	SMF record version	
ASSMFVM	DS	B11	SMF record major version	
ASSMFVN	DS	B11	SMF record minor version	
ASSMFVC	Equ	ASSMFV14	Current version: 1.4	
ASSMFV11	Equ	x'0101'	Version 1.1 - Initial release	
ASSMFV12	Equ	x'0102'	Version 1.2 - Added ALS/ASM	
ASSMFV13	Equ	x'0103'	Version 1.3 - ADARUN parms V8.3	
ASSMFV14	Equ	x'0104'	Version 1.4 - ADARUN parms V8.4	
*			in coll. seq., new section SESS	
ASSegNo	DS	B11	Record segment number	
ASSegL	DS	B11	Last segment when = 0	
ASNumD	DS	B12	Number of detail type triplets	
ASPNm	DS	C18	Product name (ADABAS)	
ASVRSC	DS	C18	Product ver/rlse/SM/cum: vvrsscc	
ASSysN	DS	C18	System name	
ASSypN	DS	C18	Sysplex name	
ASVMN	DS	C18	Virtual machine name	
ASJbN	DS	C18	Job name	
ASStN	DS	C116	ProcStep/Step name	
ASJNm	DS	C18	JES job identifier	
ASPgm	DS	C18	Program name	
ASGrp	DS	C18	Cluster messaging group name	
ASST	DS	B18	Nucleus start time in STCK format	
ASIST	DS	B18	Interval start time in STCK format	
ASIET	DS	B18	Interval end time in STCK format	
ASDBID	DS	B14	Database ID	
ASNucX	DS	B12	External nucleus ID	
ASNucI	DS	B11	Internal nucleus ID	
ASSVC	DS	B11	Adabas SVC number	
ASASID	DS	B12	Address space ID	
ASASIDI	DS	B14	Reusable address space ID instance	
ASComp	DS	B12	Completion code	+
			x'0ccc' System ABEND code ccc	+
			x'8ccc' User ABEND code ccc	
ASARC	DS	B14	ABEND reason code	
*				
ASPIDL	Equ	*-ASPID		

Adabas Command Activity Section: CMD

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Adabas command activity data is derived from data presented at nucleus shutdown.

There is one instance for each command group: A1/4, BT, CL, ET, E1/4, L1/4, L2/5, L3/6, L9, LF, N1/2, OP, UC, RC, RE, REST, S1/4, S2, S5, S8, S9, YA, YB, YF, YP, YCAL, V1, V2, V3, V4, U0, U1, U2 and U3. There are 34 possible instances but this is subject to change in future releases.

ASCmd	D Sect		Adabas Command Activity
ASCmdNm	DS	C14	Command name
ASCmdCt	DS	B18	Number of times this command type + was executed
ASCmdTm	DS	B18	Sum of this command type durations + in microseconds
*			
ASCmdL	Equ	*-ASCmd	

Adabas File Activity Section: FILE

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Adabas file activity data is derived from data presented at nucleus shutdown or in response to a DFILUSE operator command. There is one instance for each file possible in the database as specified by ADADEF MAXFILES up to the highest file number with a non-zero use count. The file number is implied by the sequence number of the instance, starting with zero, which reflects commands such as OP that are not associated with a specific file.

ASFile	D Sect		Adabas File Activity
ASFileCt	DS	B18	Number of commands executed against + this file
*			
ASFileL	Equ	*-ASFile	

Adabas Global Cache Activity by Block Type Section: CSHB

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Global cache statistics are available only for Adabas Cluster Services and Adabas Parallel Services nuclei. They are derived from the ones presented at nucleus shutdown or in response to a DX-CACHE operator command. There is one detail section instance for each type of block. Users should examine the block type and not rely on any observed order of the instances. The following block types are reported:

- AC: Address Converter
- DS: Data Storage
- DSST: Data Storage Space Table
- FCB: File Control Block

- NI: Normal Index
- UI: Upper Index
- WORK: Protection Data
- OTHR: Any other block type

ASChB	DSECT	Block	Description
ASChB	DSECT		Global Cache Activity by Block
ASCBCN	DS	B12	Cache Number
ASCBRSV1	DS	B12	Unused
ASCBBT	DS	C14	Block type
ASCBRT	DS	B18	Reads - Total
ASCBRCS	DS	B18	Reads - Completed synchronous
ASCBRCA	DS	B18	Reads - Completed asynchronous
ASCBRIC	DS	B18	Reads - Data in cache
ASCBRNI	DS	B18	Reads - Data not in cache
ASCBRFS	DS	B18	Reads - Failed - Structure
ASCBRO	DS	B18	Reads - For cast-out
ASCBROS	DS	B18	Reads - For cast-out synchronous
ASCBROA	DS	B18	Reads - For cast-out asynchronous
ASCBWT	DS	B18	Writes - Total
ASCBWCS	DS	B18	Writes - Completed synchronous
ASCBWCA	DS	B18	Writes - Completed asynchronous
ASCBWDR	DS	B18	Writes - Data written
ASCBWNR	DS	B18	Writes - Data not written
ASCBWSF	DS	B18	Writes - Structure full
ASCBVI	DS	B18	Validates issued
ASCBVF	DS	B18	Validates failed
ASCBBD	DS	B18	Block deletes issued
ASCBDR	DS	B18	Deletes reissued due to timeout
ASCBUR	DS	B18	Number of times updates redone
*			
ASChBL	Equ	*-ASChB	

Adabas Global Cache Activity by Adabas File Number Section: CSHF

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Global cache statistics are available only for Adabas Cluster Services and Adabas Parallel Services nuclei. They are derived from the ones presented at nucleus shutdown or in response to a DX-CACHE operator command. There is potentially one instance for each file possible in the database as specified by ADADEF MAXFILES. The size of this detail section precludes the ability to generate it for every possible file, so there is one detail section instance for each file that has non-zero usage. Users should examine the file number and not rely on any observed order of the instances.

ASChF	Dsect		Global Cache Activity by File
ASCFCn	DS	B12	Cache Number
ASCFRsv1	DS	B12	Unused
ASCFNum	DS	B14	File number
ASCFRT	DS	B18	Reads - Total
ASCFRCS	DS	B18	Reads - Completed synchronous
ASCFRCA	DS	B18	Reads - Completed a synchronous
ASCFRIC	DS	B18	Reads - Data in cache
ASCFRNI	DS	B18	Reads - Data not in cache
ASCFRFS	DS	B18	Reads - Failed - Structure
ASCFRO	DS	B18	Reads - For cast-out
ASCFROS	DS	B18	Reads - For cast-out synchronous
ASCFROA	DS	B18	Reads - For cast-out asynchronous
ASCFWT	DS	B18	Writes - Total
ASCFWCS	DS	B18	Writes - Completed synchronous
ASCFWCA	DS	B18	Writes - Completed a synchronous
ASCFWDR	DS	B18	Writes - Data written
ASCFWNR	DS	B18	Writes - Data not written
ASCFWSF	DS	B18	Writes - Structure full
ASCFVI	DS	B18	Validates issued
ASCFVF	DS	B18	Validates failed
ASCFBD	DS	B18	Block deletes issued
ASCFDR	DS	B18	Deletes reissued due to timeout
ASCFUR	DS	B18	Number of times updates redone
*			
ASChFL	Equ	*-ASChF	

Adabas Global Cache Activity Section: CSHG

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Global cache statistics are available only for Adabas Cluster Services and Adabas Parallel Services nuclei. They are derived from the ones presented at nucleus shutdown or in response to a DX-CACHE operator command. This detail section appears with one instance.

ASChG	Dsect		Global Cache Activity Section
ASCGCN	DS	B12	Cache Number
ASCGRsv1	DS	B12	Unused
ASCGCOD	DS	B18	Cast-out directory reads issued
ASCGCODA	DS	B18	Cast-out directory - async
ASCGCODS	DS	B18	Cast-out directory - sync
ASCGCOU	DS	B18	Unlock cast-out locks issued
ASCGCOUA	DS	B18	Unlock cast-out locks - async
ASCGCOUS	DS	B18	Unlock cast-out locks - sync
ASCGDR	DS	B18	Directory reads issued
ASCGDRA	DS	B18	Directory reads issued - sync
ASCGDRS	DS	B18	Directory reads issued - async
ASCGPub	DS	(0*9)B18	Publishing requests
ASCGSync	DS	B18	Update sync
ASCGXEnd	DS	B18	BT/CL/ET transaction end
ASCGRedo	DS	B18	Redo threshold
ASCGFull	DS	B18	Full buffer pool

ASCGA11	DS	B18	All blocks
ASCGRABN	DS	B18	Specific RABN
ASCGDS	DS	B18	File DS blocks
ASCGDSST	DS	B18	DSST blocks
ASCGNI	DS	B18	File NI blocks
*			
ASChGL	Equ	*-ASChG	

Adabas Global Lock Activity Section: LOCK

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Global lock statistics are available only for Adabas Cluster Services and Adabas Parallel Services nuclei. They are derived from the ones presented at nucleus shutdown or in response to a DXLOCK operator command. There is one detail section instance for each lock type. The lock type is implied by the sequence number of the instance, starting with one.

ASLok	Dsect		Global Lock Section
*			Lock Types
ASLokGC	Equ	1	GCB
ASLokSE	Equ	2	Security
ASLokFS	Equ	3	FST
ASLokUF	Equ	4	UFT
ASLokSO	Equ	5	Save Online
ASLokFL	Equ	6	Flush
ASLokES	Equ	7	Global ET Synchronization
ASLokRC	Equ	8	Recovery
ASLokUT	Equ	9	UFT-File
ASLokIU	Equ	10	Index Update
ASLokHI	Equ	11	Hold ISN
ASLokUD	Equ	12	Unique DE
ASLokET	Equ	13	ETID
ASLokLT	Equ	14	LOB Tracker
ASLokCM	Equ	15	Command Manager User
ASLokDI	Equ	16	Data Increment
ASLokCP	Equ	17	Checkpoint
ASLokDT	Equ	18	Net-Work DBID Target Assignment
ASLokGU	Equ	19	Global Update Command Sync
ASLokPM	Equ	20	Parameter
ASLokDS	Equ	21	DSF
ASLokRG	Equ	22	RLOG
ASLokSP	Equ	23	SPATS
ASLokCA	Equ	24	Cancel
ASLokWR	Equ	25	TBWK4A/E Table
ASLokWU	Equ	26	PUTUA/E Table
ASLokXI	Equ	27	XIDE
ASLokRH	Equ	28	Replication Handshake
ASLokRI	Equ	29	Read file/ISN
ASLokFA	Equ	30	Format AC/AC1
ASLokOC	DS	B18	Obtains - Conditional
ASLokOG	DS	B18	Obtains - Granted

ASLokOR	DS	B18	Obtains	- Rejected
ASLokOU	DS	B18	Obtains	- Unconditional
ASLokOS	DS	B18	Obtains	- Synchronous
ASLokOA	DS	B18	Obtains	- Asynchronous
ASLokAC	DS	B18	Alters	- Conditional
ASLokAG	DS	B18	Alters	- Granted
ASLokAR	DS	B18	Alters	- Rejected
ASLokAU	DS	B18	Alters	- Unconditional
ASLokAD	DS	B18	Alters	- Deadlock/Rejected
ASLokAS	DS	B18	Alters	- Synchronous
ASLokAA	DS	B18	Alters	- Asynchronous
ASLokRL	DS	B18	Releases	
ASLokRS	DS	B18	Releases	- Synchronous
ASLokRA	DS	B18	Releases	- Asynchronous
*				
ASLokL	Equ	*-ASLok		←

Adabas Internucleus Messaging Control Block Activity Section: MSGB

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Internucleus messaging statistics are available only for Adabas Cluster Services and Adabas Parallel Services nuclei. They are derived from the ones presented at nucleus shutdown or in response to a DXMSG operator command. This detail section appears with one instance. The number of blocks allocated (ASMsgBBA) and the high water mark (ASMsgBBH) reflect the entire nucleus session in interval records.

ASMsgB	Dsect	,	Inter-Nucleus Messaging Counts	
ASMsgBBA	DS	B18	Message control blocks allocated	
ASMsgBBH	DS	B18	Message control blocks used	+
			(high water mark)	
ASMsgBBR	DS	B18	Message control block requests	
*				
ASMsgBL	Equ	*-ASMsgB		

Adabas Internucleus Messaging Counts Section: MSGC

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Internucleus messaging statistics are available only for Adabas Cluster Services and Adabas Parallel Services nuclei. They are derived from the ones presented at nucleus shutdown or in response to a DXMSG operator command. This detail section appears with one instance. Adabas Parallel Services nuclei report only the count of messages sent.

ASMsgC	D	Sect	,	Inter-Nucleus Messaging Counts
ASMsgCMT	DS		C14	Message type
ASMsgCMS	DS		B18	Messages sent
ASMsgCMI	DS		B18	Messages incoming (arrived)
ASMsgCMA	DS		B18	Messages accepted
ASMsgCRS	DS		B18	Replies sent
*				
ASMsgCL	Equ		*-ASMsgC	

Adabas Internucleus Messaging Service Time Histogram Section: MSGH

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Internucleus messaging statistics are available only for Adabas Cluster Services and Adabas Parallel Services nuclei. They are derived from the ones presented at nucleus shutdown or in response to a DXMSG operator command. This detail section appears with two instances:

1. The first represents messages subject to the MXMSG timeout parameter.
2. The second represents certain control messages not subject to MXMSG.

The two instances may be summed for a single representation of all messages. All message times are in microseconds. The minimum and maximum durations (ASMsgHMn and ASMsgHMx) reflect the entire nucleus session in interval records. Field ASMsgMD2 is an extended (16-byte) floating point sum of the squares of all message durations. It may be used to compute a standard deviation.

ASMsgH	D	Sect	,	Inter-Nucleus Messaging Histogram
ASMsgHXP	DS		C14	Transport service
ASMsgHMM	DS		B14	MXMSG or zero for messages not + subject to MXMSG
ASMsgHMC	DS		B18	Message count
ASMsgHMD	DS		B18	Sum of all message durations
ASMsgHMS	DS		B116	Sum of squares, all msg durations + (extended hex floating point)
ASMsgHMn	DS		B14	Minimum duration (us)
ASMsgHMx	DS		B14	Maximum duration (us)
ASMsgHCt	Equ		9	Number of histogram buckets
ASMsgHG	DS		(0*ASMsgHCt)B18	Histogram buckets
ASMsgH10	DS		B18	> 1000 s
ASMsgH09	DS		B18	> 100 s, <= 1000 s
ASMsgH08	DS		B18	> 10 s, <= 100 s
ASMsgH07	DS		B18	> 1 s, <= 10 s
ASMsgH06	DS		B18	> 100 ms, <= 1 s
ASMsgH05	DS		B18	> 10 ms, <= 100 ms
ASMsgH04	DS		B18	> 1 ms, <= 10 ms
ASMsgH03	DS		B18	> 100 us, <= 1 ms
ASMsgH02	DS		B18	<= 100 us
*				
ASMsgHL	Equ		*-ASMsgH	

Adabas Parallel Services Cache Activity Section: CSHP

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Parallel services cache statistics are available only for Adabas Parallel Services nuclei. They are derived from the ones presented at nucleus shutdown or in response to a DXCACHE operator command. This detail section appears with one instance. The directory high water mark `ASCPDHiN` and in-use count `ASCPDirI` reflect the entire nucleus session in interval records.

ASChP	DSect		Parallel Services Cache Activity
ASCPDN	DS	B12	Cache Number
ASCPDsv1	DS	B12	Unused
ASCPNDir	DS	B18	Number of directory elements
ASCPNDiI	DS	B18	Number of directory index elements
*			Directory Statistics
*			General
ASCPDHiN	DS	B18	High-water mark, this nucleus
ASCPDirI	DS	B18	In-use, this nucleus
*			Read
ASCPDRA	DS	B18	Located active
ASCPDRF	DS	B18	Obtained from free pool
ASCPDRC	DS	(0*4)B18	Reclaim criteria categories
ASCPDNN	DS	B18	First choice criteria
ASCPDND	DS	B18	Second choice criteria
ASCPDIN	DS	B18	Third choice criteria
ASCPDID	DS	B18	Fourth choice criteria
ASCPDCF	DS	B18	Unable to obtain (cache full)
ASCPDRT	DS	B18	Tested for reclaim
*			Write
ASCPDWF	DS	B18	Obtained from free pool
*			Space Management Statistics
*			Request Statistics
ASCPDRP	DS	B18	Sufficient preallocated space
ASCPDRF	DS	B18	Free space allocated
ASCPDRN	DS	B18	Reclaim space, first choice
ASCPDRI	DS	B18	Reclaim space, second choice
ASCPDRU	DS	B18	Space unavailable (cache full)
ASCPDRS	DS	B18	Searched part of space chain
ASCPDRF	DS	B18	Searched entire space chain
ASCPDRS	DS	B18	Number of space seqs tested
*			Element Reclaim Statistics
ASCPDRN	DS	B18	First choice criteria
ASCPDRI	DS	B18	Second choice criteria
*			Latch management statistics
*			Cache Space Chain
ASCPDRG	DS	B18	Get Exclusive
ASCPDRW	DS	B18	WaitFor Exclusive
ASCPDRP	DS	B18	Release Exclusive
*			Cache Directory Index
ASCPDRG	DS	B18	Get Exclusive
ASCPDRS	DS	B18	Get Shared
ASCPDRU	DS	B18	Upgrade Exclusive

ASCPDIWE	DS	B18	WaitFor Exclusive
ASCPDIWS	DS	B18	WaitFor Shared
ASCPDIWU	DS	B18	WaitFor Upgrade
ASCPDIWE	DS	B18	Release Exclusive
ASCPDIRS	DS	B18	Release Shared
*			Cache Directory
ASCPDRGE	DS	B18	Get Exclusive
ASCPDRGS	DS	B18	Get Shared
ASCPDRUE	DS	B18	Upgrade Exclusive
ASCPDRWE	DS	B18	WaitFor Exclusive
ASCPDRWS	DS	B18	WaitFor Shared
ASCPDRRE	DS	B18	Release Exclusive
ASCPDRRS	DS	B18	Release Shared
*			Cast-Out Class
ASCPCOGE	DS	B18	Get Exclusive
ASCPCOGS	DS	B18	Get Shared
ASCPCOWE	DS	B18	WaitFor Exclusive
ASCPCOWS	DS	B18	WaitFor Shared
ASCPCORE	DS	B18	Release Exclusive
ASCPCORS	DS	B18	Release Shared
*			
ASChPL	Equ	*-ASChP	

ADARUN Parameter Value Section: PARM

This selectable detail section may appear in initialization records or whenever an ADARUN parameter is changed while the nucleus is running (subtypes 1 and 4 in the header section). It will not be generated for interval or termination records (subtypes 2 and 3). This section has a fixed-length portion containing most parameters, followed by variable-length areas for parameters capable of multiple values or lists of values.

Where possible, the individual field names are formed by prefixing the shortest allowable form of the parameter with ASP. In general, the SMF record will report character parameters in EBCDIC and numeric parameters in binary.

Parameters with limited enumerated values (YES or NO, for example) are reported in 1-byte fields if the possible values are unambiguous. Otherwise, the field length is that used by the nucleus, usually 4 bytes.

Here are some sample entries:

ASParm	Dsect	,	
ASPA0	DS	C11	A0slog
ASPARE	DS	B14	ARExclude
			Offset to file table
ASPARMN	DS	C116	ARMname
ASPASS	DS	C11	ASSocache
ASPASY	DS	C11	ASYtvs
.	.	.	

```

ASPVI   DS    C11    Vista
ASPV64B DS    C11    V64Bit
ASPWO   DS    C11    Workcache
ASParmV DS    0B     Begin variable part
ASParmL Equ   *-ASParm Length of ID section ↵

```

```

ASParm  DSect ,          ADARUN Parameters
*
*          Adabas Nucleus Parameters
*
ASPA0   DS    C11          A0slog
ASPARE  DS    B14          ARExclude      Offset to file table
ASPARMN DS    C116         ARMname
ASPASS  DS    C11          ASSocache
ASPASY  DS    C11          ASYtvs
. . .
ASPVI   DS    C11          V1sta

```

The ADARUN AREXCLUDE parameter is a variable length list of values. The base parameter entry will be an offset from the beginning of the detail section to the table of values. The table is a 4-byte inclusive field followed by 4-byte file numbers. A separate DSECT maps the AREXCLUDE file exclusion table:

```

ASPAFE  DSect ,          ARM File Exclusion Table
ASPAFEN DS    B14          Inclusive length of table
ASPAFEF DS    0B14        First file number entry

```

I/O by DD Name Section: IODD

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). I/O by DD data is derived from data presented at nucleus shutdown. There is one instance for each DD statement administered by the nucleus. You should examine the DD name and not rely on any observed order of the instances. An Adabas nucleus may open and close the same DD name multiple times. Multiple uses of a DD name are summed.

You might see this many DD statement entries in a single nucleus:

Statement Type	Number of DD Statements
ADC	248
ADP	248
ADW	62
ASSO	99
CLOG	8
DATA	99
ECS	1

Statement Type	Number of DD Statements
PLOG	8
RLOG	1
WORK	2

DD names beginning with "AD" represent files that were dynamically allocated, such as during Cluster or Parallel Services recovery, after another nucleus failed.

ADC and ADP represent CLOGs and PLOGs from other nuclei. There may be one set of 8 for each of the 31 possible other nuclei up to a maximum of 248.

Similarly, ADW represents a WORK data set from another nucleus. There may be two for each of the 31 possible other nuclei up to a maximum of 62.

```

ASIODD  DSect ,           I/O Activity by DD
ASIODDn DS   C18         DD Name
ASIODDRd DS   B18         Reads
ASIODDwt DS   B18         Writes
*
ASIODDL  Equ   *-ASIODD
    
```

Session Statistics Section: SESS

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Session statistics data is similar to the data presented at nucleus shutdown or at the DSTAT operator command.

There is one single instance per record. The section has a fixed-length.

Here is a shortened sample of the SESS section:

```

ASSess  DSect ,           Nucleus session statistics
*
ASSDURA DS   BL8         Nucleus Session Duration
ASSWAIT  DS   BL8         Nucleus Wait Time
ASSCPU   DS   BL8         Nucleus CPU Time

ASSASR   DS   BL8         ASSO Reads
ASSASW   DS   BL8         ASSO Writes

...

ASSBFFE  DS   BL4         Buffer efficiency .x precision
ASSessL  Equ   *-ASSess   Length of the session section
    
```

Storage Pool Section: STG

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Storage pool statistics are derived from statistics presented at nucleus shutdown or in response to a DRES operator command. There is one instance for each storage pool with a non-zero size. Be sure to examine the pool name and not rely on any observed order of the instances.

Storage pool statistics are reported two ways: in bytes and also in units such as a user might specify as an ADARUN parameter, for example, NC. When the units are bytes, the two sets of statistics are the same.

Normally an interval record would show the change from the previous interval, but that isn't meaningful for storage pools. Thus the interval and termination record subtypes all reflect total usage for the nucleus session.

ASStg	D Sect		Storage Pool Usage
ASStgNm	DS	C14	Storage pool name
ASStgBSz	DS	B18	Size in bytes
ASStgBHW	DS	B18	High water mark in bytes
ASStgUSz	DS	B18	Size in units from ADARUN parameter
ASStgUHW	DS	B18	High water mark in ADARUN units
*			
ASStgL	Equ	*-ASStg	Length of Storage Pool section

These are the possible storage pools:

Storage Pools		
Pool Name	ADARUN Parameter	Description
AB	NAB	Attached buffers
CQ	NC	Command queue
DUQ	LDEUQP	Unique descriptor
FI	LFP	Internal format buffers
HQ	NH	Hold queue
PLIO	NPLOGBUFFERS	PLOG I/O buffers
REDO	LRDP	Deferred publishing
RPL	LRPL	Replication pool
SC	LCP	Security information
TBI	LI	ISN table
TBS	NQ	Sequential command table
UQ	NU	User queue element
UQF	NU	User queue file elements
WKIO	NWORK1BUFFERS	Work I/O buffers
WORK	LWP	Work

Storage Pools		
Pool Name	ADARUN Parameter	Description
WKP1	LP	Work part 1
WK1B	CLUWORK1CACHE	Work part 1b
WKP2	LWKP2	Work part 2
WKP3	--	Work part 3
XID	NU	Transaction ID

Thread Activity Section: THRD

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). Thread activity data is derived from data presented at nucleus shutdown or in response to a DTH operator command. The ADARUN parameter NTHREAD defines the number of user threads for the nucleus session. There is one instance for each defined user thread. The thread number is implied by the sequence number of the instance.

```

ASThrd  DSect ,          Thread Activity
ASThrdCt DS      B18      Number of commands executed in this +
                          thread
*
ASThrdL  Equ    *-ASThrd
    
```

zIIP Statistics Section: ZIIP

This selectable detail section may appear in interval or termination records (subtypes 2 and 3). zIIP activity data is derived from data presented at nucleus shutdown or in response to a DZSTAT operator command. There is one instance for each zIIP-enabled nucleus product.

```

ASzIIP   DSect ,          Nucleus session statistics
*
ASzProd  DS      C14      Sub-product name
ASzPNuc  Equ    c'ANUC'   Adabas nucleus main task
ASzPRevL Equ    c'REVL'   Review local
ASzRsv1  DS      B14      Unused
*
ASz2SRB  DS      B18      Count of switches to SRB mode
ASz2TCB  DS      B18      Count of switches to TCB mode
ASzPAUT  DS      B18      Count of pause TCB requests
ASzRLST  DS      B18      Count of release TCB requests
ASzPAUS  DS      B18      Count of pause SRB requests
ASzRLSS  DS      B18      Count of release SRB requests
ASzQPRQ  DS      B18      Count of queue parallel requests
ASzXRQE  DS      B18      Count of no free RQE for requests
*
ASzNctr  Equ    64        Number of event counters
ASzCtr   DS      (ASzNctr)B18 Event counters
    
```

```

*
ASzCPUT   DS    B18           CPU time           (STCK format)
ASzZTim   DS    B18           zIIP time          (STCK format)
ASzSCPU   DS    B18           SRB-on-CPU time    (STCK format)
ASzZQCP   DS    B18           zIIP-qualified time (STCK format)
*
ASzIWait  DS    B18           ADAIOR general WAITs
ASzIRlse  DS    B18           ADAIOR releases
*
ASzI IPL   Equ   *-ASzIIP      Length of the zIIP section

```

ASMFREC Mapping Macro

Use the ASMFREC macro to generate the latest SMF record DSECTs. The ASMFREC macro will always generate the header and self-defining section DSECT. Detail section DSECTs will be generated as specified. The header and self-defining sections are mapped by a single DSECT. Each detail section is mapped by its own DSECT. The syntax of the ASMFREC macro is:

```

label    ASMFREC   Prefix={AS | prefix},
                               Detail={All | (type [,type]...)},
                               Title = {'Adabas_SMF_Records' | 'string'}

```

Prefix

Specify a character string to be used as the initial characters for all DSECT and field names. The default is Prefix=AS.

Detail

Identify which detail section DSECTs are to be included in the expansion. "All" is the default and will include all detail sections. Alternatively, a comma-delimited list of types (enclosed in parentheses) can be specified; only the types specified will be included. The valid types are shown in the following table. A null value (Detail=) will inhibit all detail section DSECTs.

ASMFREC Macro Specification	Detail Section Description	ASMFREC DSECT Name Produced by the ASMFREC Macro ¹
CMD	Adabas command activity	xxCmd
CSHB	Adabas global cache activity by block type	xxChB
CSHF	Adabas global cache activity by Adabas file number	xxChF
CSHG	Adabas global cache activity	xxChG
CSHP	Adabas Parallel Services cache activity	xxChP
FILE	Adabas file activity	xxFile

ASMFREC Macro Specification	Detail Section Description	ASMFREC DSECT Name Produced by the ASMFREC Macro ¹
LOCK	Adabas global lock activity	xxLok
MSGB	Adabas internucleus messaging control block activity	xxMsgB
MSGC	Adabas internucleus messaging counts	xxMsgC
MSGH	Adabas internucleus messaging service time histogram	xxMsgH
PARAM	ADARUN parameter values	xxParm
IODD	I/O by DD name	xxIODD
SESS	Nucleus session statistics	xxSess
STG	Storage pool	xxStg
THRD	Thread activity	xxThrd
ZIIP	zIIP statistics	xxzIIP
USER	User-defined	user-defined

1. Where *xx* is the prefix specified in the ASMFREC macro.

Title

If the Title default "Adabas SMF Record " or another quoted string is specified, an assembler Title statement is generated before the header section DSECT. A null value (Title=) for this operand will inhibit a title in the DSECT.

SMF User Exit

You can provide a user exit if you want to add a detail section to the Adabas SMF record. The user exit is a separate load module whose name must be provided in the ADARUN UEXSMF parameter. For complete information about the user exit, read *SMF User Exit*, in the *Adabas User, Hyperdescriptor, Collation Descriptor, and SMF Exits Manual*.

IBM Type 89 SMF Records

An Adabas nucleus can register with z/OS to have CPU usage statistics included in IBM type 89 SMF records. These records are described in *z/OS MVS System Management Facilities (SMF)*, IBM document SA22-7630.

To activate type 89 recording for Adabas, specify ADARUN parameters SMF=YES and SMF89=YES. During initialization Adabas will register the nucleus address space with z/OS SMF and have its CPU statistics included in subtype 1 of the type 89 records. The address space is deregistered at nucleus termination. Each Adabas nucleus appears as a separate type 89 entry.

The type 89 entries include CPU usage and a number of descriptive registration parameters. Adabas nuclei use these descriptive fields in type 89 entries as follows:

SMF Type 89 Descriptive Fields				
Name	Length	Format	Description	Value
SMF89UPO	16	EBCDIC	Product owner or vendor name	SOFTWARE AG
SMF89UPN	16	EBCDIC	Product name	ADABAS
SMF89UPV	8	EBCDIC	Product version	The eight-byte product version has two-byte numeric values for the Adabas version, release, SM level, and cumulative level.
SMF89UPQ	8	Binary	Product qualifier	The product qualifier is a seven-byte string that may be used to distinguish among several nucleus instances. It contains a series of binary fields: SVC (1 byte) DBID (4 bytes) NucID (2 bytes) Use both the SVC and DBID to identify instances of Adabas Cluster or Parallel Service nuclei for the same database on any one system.
SMF89UPI	8	EBCDIC	Product ID	The product ID is a string of up to eight single characters to show what add-on products are being used. The characters may appear in any order: C (Adabas Cluster Services) D (Adabas Delta Save) F (Adabas Fastpath) M (Adabas Review) P (Adabas Parallel Services) R (Event Replicator for Adabas) S (Adabas Cache Facility) T (Adabas Transaction Manager)

SMF Type 89 Descriptive Fields				
Name	Length	Format	Description	Value
				<p>U (Adabas Security) V (Adabas Vista)</p> <p>The following applies:</p> <p>C, D, P, and R are mutually exclusive. D, F, M, T and V are exclusive with A.</p>

13

Adabas Online System Cluster Environment Screens

■ Scrolling through Nucleus IDs in a Cluster	94
■ Displaying Cluster Members	94
■ Nucleus File Status	96
■ Nucleus Status Flags	97
■ Cluster Usage	100
■ Estimating Sizes for the Cache Structure in a Cluster Environment	107
■ Estimating Sizes for the Lock Structure in a Cluster Environment	111
■ Maintain the User Table	114

This chapter describes the Adabas Online System screens that apply to a cluster environment.

Scrolling through Nucleus IDs in a Cluster

You can use the PF2 key (function key 2) on some cluster environment screens in Adabas Online System (AOS) to scroll through the nuclei in a cluster. This allows you to review the same sets of statistics for each of the nuclei in a cluster.

When you press PF2 on such an AOS screen (or enter the `NextNucid` command), the same screen appears, but with data pertaining to the nucleus with the next highest nucleus ID in the cluster. If there is no higher nucleus ID, AOS cycles back to the beginning and displays the same screen, with data pertaining to the nucleus with the lowest ID in the cluster.

The screens on which this functionality is provided are:

- **Session Monitoring** menu
- **Display Parameters** screen
- **Modify Parameters** screen
- **Queue Displays** menu
- **Display User Queue** screen
- **Resource Utilization** menu

Once the nucleus ID is incremented on any of these screens, it remains in use on all AOS screens until it is changed.

Displaying Cluster Members

From the **Session Monitoring** menu, a new function *Display cluster members* (option A) produces the following screen:


```
14:04:00          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
DBID 1955          - Display PPT Entry -          MACA012

NucID ... 1021    Active Nucleus, PLOG(s) not copied, CLOG(s) not copied

Name      Dataset Status          DataSet Name
-----
WORK1
PLOGR1
PLOGR2
CLOGR1
CLOGR2
RD.USAXXX.DB1955.WORKR1
RD.USAXXX.DB1955.PLOGR1
RD.USAXXX.DB1955.PLOGR2
RD.USAXXX.DB1955.CLOGR1
RD.USAXXX.DB1955.CLOGR2

Press 'ENTER', PF3 or PF12 to continue
PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help                Exit                                Menu                ↵
```

Nucleus File Status

From the **Interval Utilization** menu, the *Nucleus file status* (option N) can be used to determine the status of nucleus files. This is the equivalent of the DNFV operator command.

```

14:05:09          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
DBID 1955          - Nucleus File Status -          PACUN02
NucID 1021
    
```

File	Locking NucID	Access count	Update count	State
29		0	0	Access
39		0	0	Access
40		0	0	Access
41		0	0	Access
90		0	0	Update

```

Last page
PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help      Repos      Exit      Refresh      -      +      Menu      ↵
    
```

In an Adabas cluster environment, the file may be locked for exclusive use by another cluster nucleus. If this is the case and the file is in the nucleus file status table, the Locking NucID column for the file shows the ID of the nucleus that has exclusive control.

The Access count / Update count fields display the number of access or update users, respectively, that refer to the specified file in their user queue elements (UQEs). These users either have specified the file in an OP command with R-option or are using the file in an as yet incomplete transaction.

A State field indicates when the file is used for access only or for access and update. The State field indicates to what extent a nucleus can use a file on its own. If the requested use exceeds the given state, the nucleus must first communicate with the other nuclei in the cluster in order to upgrade the state.

Nucleus Status Flags

From the **Interval Utilization** menu, a second screen, *System status* (option S), displays I/O counts for the ASSO, DATA, WORK, and PLOG data sets; remote and local call distribution; and other current session status information.

```

14:07:19          ***** A D A B A S BASIC SERVICES *****          2014-05-02
DBID 1955          - System Status -          PACUS12
NucID: 1021

Physical      Reads          Writes          Call Distribution
-----
ASSO          7,529          280          No. of HQEs active .....          0
DATA          4,946          90          No. of UQEs in User Queue ..          1
WORK          2          379          No. of CQEs waiting in CQ ..          0
PLOG          15          274          No. of PLOG switches .....          0
CLOG          29          10          No. of CLOGs .....          2
              No. of PLOGs .....          2

PLOG protection blks .          273
PLOG protection I/Os .          273          Format translations .....          577
WORK1 protection blks.          379          Format overwrites .....          0
WORK1 protection I/Os.          379          Auto-restarts .....          13
              Throw-backs for ISN .....          0
Logical reads .....          35,163          Throw-backs for Space....          0
Buffer efficiency ....          2.8

                                                    page 1 of 3

PF1---- PF2---- PF3----- PF4----- PF6----- PF7----- PF8----- PF9----- PF12----
Help          Exit          Refresh          +          FullView Menu ←
    
```

Pressing PF8 displays another screen of system status information.

```

14:07:19          ***** A D A B A S BASIC SERVICES *****          2014-05-02
DBID 1955          - System Status -          PACUS12
NucID: 1021

          Call Distribution          Commands
-----
User sessions .....          67          Remote .....          0
Buffer flushes .....          91          Local .....          8,573
Flush phases .....          91          Internal .....          6,122
Blocks flushed .....          360          Operator .....          0
Flush I/Os .....          360          Total .....          14,695

          Flush Requests Returned          Time Elapsed
-----
Immediately .....          1          Duration .... 3 days, 17:33:36
After logical flush ..          0          Wait-time ... 3 days, 17:33:25
After entire flush ...          98          CPU time .... 0 day, 00:00:39

                                                    page 2 of 3

PF1---- PF2---- PF3----- PF4----- PF6----- PF7----- PF8----- PF9----- PF12----
Help          Exit          Refresh          -          +          FullView Menu ←
    
```

Pressing PF8 again displays an additional screen that indicates if one or more of the following are in progress:

- Online database save running;
- ADAEND in progress;
- Online file save running;
- READONLY/UTIONLY transition;
- READONLY status;
- Update processing suspended;
- ET-sync in progress;
- UTIONLY status; and
- Exclusive-DB-control utility running.

Otherwise, "Adabas operation normal" is displayed.

```

14:07:19          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
DBID 1955          - System Status -          PACUS12
NucID: 1021

                                Nucleus Status Flags
                                -----
                                Adabas operation normal

                                                                page 3 of 3

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF9----- PF12-----
Help          Exit      Refresh          -          FullView Menu  ←
    
```

Cluster Usage

From the **Session Monitoring** menu, *Cluster usage* (option X) displays nucleus cluster statistics that are equivalent of those displayed using the DXCACHE, DXLOCK, and DXFILE operator commands.

The equivalent direct command is:

```
DISPLAY CLUSTERSTATUS
```

```

14:09:48          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
                   - Cluster Usage -                               PACUX02

                Code      Service
                ----      -
                C          Cache statistics
                F          File statistics
                L          Lock statistics
                ?          Help
                .          Exit
                ----      -

Code ..... _
File Number .. 29
Database ID .. 1955 (WIS1955)          NucID .. 1021

Command ==>
PF1----- PF2----- PF3----- PF4----- PF6----- PF9----- PF10----- PF12-----
Help          Exit          Fuse          Flist          Menu          ↵
    
```

This section covers the following topics:

- [Cache Statistics](#)
- [File Statistics](#)

- [Lock Statistics](#)

Cache Statistics

Choosing *cache statistics* (option C) from the Cluster Usage menu displays the following menu:

```

14:10:10          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
                   - Cache Statistics -                               PACUX12

      Code      Service
      ----      -
      K      Cast-out / Directory
      P      Publishing requests
      X      Individual cache blocks
      .      Exit
      ?      Help
      ----      -

Code ..... _
Database ID .. 1955   (WIS1955)                NucID .. 1021

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help          Exit      Refresh          Menu          ↵
    
```

The rest of this section describes each of the options on this screen.

- [Cast-out / Directory](#)
- [Publishing Requests](#)
- [All Cache Blocks](#)

Cast-out / Directory

Choosing *cast-out / directory* (option K) from the Cache Statistics menu display the following:

```

14:10:10          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
DBID 1955          - Cast-out / Directory -          PACUX12
NucID 1021

      Cast-out Directory Reads          Directory Reads
      -----
Total .....          182          Total .....          10
  Sync .....          107          Sync .....          0
  Async ....          75          Async ....          10

Unlock Cast-out Calls
-----
Total .....          91
  Sync .....          0
  Async ....          91

PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help          Exit          Refresh          Detail          Menu          ↵
    
```

Counters have a multiplier column with the following values:

Value	The total shown is in ...
blank	(factor of 1)
K	kilo (factor of 1,000)
M	mega (factor of 1,000,000)
G	giga (factor of 1,000,000,000)

If a number has a multiplier shown, it has been divided by the multiplier, showing the significant digits to 9 places with no decimal point.

Press PF9 to see the entire value. This value is the exact count up to 20 digits in length.

Publishing Requests

Choosing *publishing requests* (option P) from the Cache Statistics menu display the following:

```

14:11:14          ***** A D A B A S BASIC SERVICES *****          2014-05-02
DBID 1955          - Publishing Requests -          PACUX12
NucID 1021

          Publishing Request Category
          -----
          Update sync .....          91
          BT or CL or ET ....          3
          Redo threshold ....          0
          Full bufferpool ...          0
          All blocks .....          97
          Specific RABN .....          0
          File DS blocks ....          0
          All DSST blocks ...          90
          File NI blocks ....          0

PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help       Exit       Refresh      Detail      Menu       ↵
    
```

All Cache Blocks

Choosing *all cache blocks* (option X) from the Cache Statistics menu display the following:

```

14:11:54          ***** A D A B A S BASIC SERVICES *****          2014-05-02
DBID 1955          - All Cache Blocks -          PACUX12
NucID 1021

          Reads                               Writes
          -----                               -----
          Total .....          7,460          Total .....          373
          Sync .....          1,241          Sync .....          179
          Async .....          6,219          Async .....          194

          In cache .....          105          Written .....          373
          Not in cache ..          7,355          Not written .....          0
          Struc. full ...          0          Struc. full .....          0

          Cast-out Reads                       Other
          -----                               -----
          Total .....          360          Validates .....          44,877
          Sync .....          360          Invalid .....          0
          Async .....          0          Deletes .....          0
                                   Timeouts .....          0
                                   Redo processes .....          0

PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help       Repos      Exit       Refresh    PrevBlk   NxtBlk    Detail      Menu       ↵
    
```

Use PF7 and PF8 to scroll through the cache blocks; use PF2 to reposition.

Statistics are displayed for the following:

- All cache blocks
- Address converter (AC) cache blocks
- Data Storage (DS) cache blocks
- Data Storage space table (DSST) cache blocks
- File control block (FCB) cache blocks
- Normal index (NI) cache blocks
- Upper index (UI) cache blocks

Press PF9 from the above screen to display the following detail screen:

```

14:11:54          ***** A D A B A S BASIC SERVICES *****          2014-05-02
DBID 1955          - All Cache Blocks -          PACUX12
NucID 1021
Reads                      Writes
-----
Total .....              7,460 Total .....              373
  Sync .....              1,241  Sync .....              179
  Async .....              6,219  Async .....              194

  In cache..              105   Written ...              373
  Not in ...              7,355  Not writ ..              0
  Stru.full.              0     Stru.full .              0

Cast-out Reads            Other
-----
Total .....              360  Validates ...           44,877
  Sync .....              360  Invalid ...              0
  Async .....              0   Deletes .....           0
                                   Timeouts ..              0
                                   Redo procs ..            0

                                   Press Enter to continue  ↵
    
```

File Statistics

Choosing *file statistics* (option F) from the Cluster Usage menu for file 25 displays the following menu:

```

14:12:42          ***** A D A B A S BASIC SERVICES *****          2014-05-02
DBID 1955          - File 29 Statistics -          PACUX22
NucID 1021
  Reads                                Writes
  -----                                -----
  Total .....                          5          Total .....                          0
  Sync .....                            0          Sync .....                          0
  Async .....                           5          Async .....                          0

  In cache .....                        0          Written .....                       0
  Not in cache ..                        5          Not written .....                   0
  Struc. full ...                        0          Struc. full .....                   0

  Cast-out Reads                        Other
  -----                                -----
  Total .....                          0          Validates .....                     12
  Sync .....                            0          Invalid .....                      0
  Async .....                           0          Deletes .....                      0
                                          Timeouts .....                     0
                                          Redo processes .....                0

PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9----- PF12-----
Help      Repos      Exit      Refresh      Detail      Menu      ↵
    
```

Lock Statistics

Choosing *lock statistics* (option L) from the Cluster Usage menu displays the following menu:

```

14:13:03          ***** A D A B A S BASIC SERVICES *****          2014-05-02
                    - Lock Statistics -                               PACUX39

Code  Service                Code  Service                Code  Service
-----
A  Buffer flush                L  Global ET sync        W  Replicat handshake
B  Cancel                      M  Glbl upd cmd sync     X  Security
C  Checkpoint                  N  Hold ISN               Y  Spats
D  Container                    O  LOB tracker            Z  Unique descriptor
E  Delta Save (DSF)            P  Net-wrk DBID asgmt    1  User
F  Distributed tranID          Q  New-Data-RABN         2  Wkpt4 RABN index
G  ETID                         R  Online save           3  Wkpt4 PET user tbl
H  File-lock-table             S  Parameter              .  Exit
I  Format-addr-conver          T  Record read           ?  Help
J  Free space table            U  Recovery (online)
K  GCB                          V  Recovery log(RLOG)

-----
Code ..... _
Database ID .. 1955 (WIS1955)          NucID .. 1021

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help          Exit      Refresh          Menu
0             19,031      ←
    
```

Each of the options on the Lock Statistics menu displays statistics for a particular lock. For each lock, the screen displays obtain and release information about the various types of that lock that are currently in use by a cluster nucleus:

- The system may obtain locks conditionally or unconditionally, synchronously or asynchronously. A conditional request for a lock may be granted or rejected.
- Releases may be performed synchronously or asynchronously.

Hold ISN Lock

Choosing *hold ISN lock* (option J) from the Lock Statistics menu displays the following:

```

14:13:03          ***** A D A B A S BASIC SERVICES *****          2014-05-02
DBID 1955          - Hold ISN Lock -          PACUX39
NucID 1021

  Obtains          Alters
-----          -----
Conditional ....          91          Conditional .....          0
  Granted .....          91          Granted .....          0
  Rejected ....          0          Rejected .....          0
Unconditional ..          0          Unconditional .....          0
          Rejected-deadlock ..          0

Sync .....          3          Sync .....          0
Async .....          88          Async .....          0

Releases
-----
Issued .....          91
  Sync .....          8
  Async .....          83

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help      Repos      Exit      Refresh          PrevLok  NxtLok  Menu      ↵
    
```

Use PF7 and PF8 to scroll through the locks; use PF2 to reposition.

Estimating Sizes for the Cache Structure in a Cluster Environment

These instructions describe how to use the structure size calculator to estimate the size for the cache structure in a cluster environment. The cache structure should be made large enough to provide sufficient space for:

- Tracking all blocks kept in the buffer pools of all connected cluster nuclei (directory elements) and
- Keeping all changed blocks until they are written to the database (data elements).

The assignment of total cache space into directory and data elements is done via the `DIRRATIO` and `ELEMENRATIO ADARUN` parameters.

The actual cache structure size value is dependent on coupling facility internals and may vary across different coupling facility levels. If the estimates by this structure size calculator are too far off the real allocations reported by cluster nuclei, you can tweak the directory element size value to bring the calculator more in line with the actual coupling facility being used.

> To access and use the space calculator for the cache structure in a cluster environment

- 1 Select option **S** on the **Basic Services Main Menu** in AOS.

The **Space Calculation** menu appears.

```

14:15:27          ***** A D A B A S BASIC SERVICES *****          2014-05-02
                  - Space Calculation -                               PSP0002

                  Code      Service
                  -----
                  A        ASSO
                  C        Cluster-Cache/Lock
                  D        DATA
                  F        DDFILEA
                  S        SORT
                  T        TEMP
                  W        WORK
                  ?        Help
                  .        Exit
                  -----
Code ..... _
Database ID ... 1955 (WIS1955)

Command ==>
PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help                Exit                                Menu
    
```

- 2 Select option **C** on the **Space Calculation** menu.

The **Cache Structure Calculator** panel appears.

```

14:17:34          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
DBID 1955          -  Cache Structure Calculator  -                      PSPC002

Smallest block size in DB ..... 2544
Largest block size in DB ..... 5724
Buffer pool size (LBP) ..... 80896_____
Size proper for caching blocks .. 100000_____
Max nuclei in cluster ..... 3
Directory element size ..... 400
Cache structure size (in KB) .... _____

For minimum calculation, leave cache structure size field empty.
Modify values, press Enter to provide estimates below.

Cache CFRM SIZE/INITSIZE ..... 2650          ( 2.5          MB)
ADARUN DIRRATIO ..... 67
ADARUN ELEMENTRATIO ..... 48
Cache directory elements ..... 135
Cache data elements ..... 97
Cache data element size ..... 1024

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help                Exit          Lock                Menu                ↵
    
```

3 Specify values for the following fields on the **Cache Structure Calculator** panel.

Field	Description	Default
Smallest block size	Specify a value between 1024 and 32768 bytes.	The smallest block size of the current Adabas Online System database ID.
Largest block size	Specify a value between 1024 and 32768 bytes. If the value of the Smallest block size field exceeds this value, then the Smallest block size value is swapped in.	The largest block size of the current Adabas Online System database ID.
Buffer pool size	Specify a value between 80,000 and 999,999,999,999 bytes.	The value of the LBP parameter setting for the current Adabas Online System database ID.
Size proper for caching blocks	Specify a value between 100000 - 999,999,999,999 bytes.	The LBP parameter of the current Adabas Online System database ID, rounded down to nearest 100000. "Size proper" means that this does not include the overhead in the cache structure required for administering these blocks. Thus, this value specifies how much space should be available in the cache structure for keeping changed blocks between buffer flushes and for buffering blocks so that the cluster nuclei do not have to read them from the database.

Field	Description	Default
Max Nuclei in cluster	Specify a value between 2 and 32.	3
Directory element size	Specify a value between 100 and 999 bytes. This value specifies how much space (including the overhead for the access paths) will be used in the cache structure by each directory element.	400
Cache Structure size	Specify a blank for the minimum calculation or specify a value between 100 and 999,999,999 (KB). Although this value is given as an output field, you may want to propose a cache structure size, to see how to allocate the cache space (directory and data elements).	blank

- 4 Press Enter after all values in the previous step are specified.

The following output fields on the screen are filled.

Field	Description
Cache CFRM SIZE/INITSIZE	The recommended cache structure SIZE or INITSIZE specification in the coupling facility resource management policy.
ADARUN DIRRATIO	The recommended ADARUN DIRRATIO parameter settings for the cluster nuclei.
ADARUN ELEMENTRATIO	The recommended ADARUN ELEMENTRATIO parameter settings for the cluster nuclei.
Cache directory	The estimated directory and data element counts resulting from the SIZE/INITSIZE and DIRRATIO settings.
Cache data elements	The estimated directory and data element counts resulting from the SIZE/INITSIZE and ELEMENTRATIO settings.
Cache data element size	This (accurate) value depends only on the largest Asso/Data/Work block size in the database.

Estimating Sizes for the Lock Structure in a Cluster Environment

These instructions describe how to use the structure size calculator to estimate the size for the lock structure in a cluster environment. The lock structure should be made large enough to provide sufficient space for:

- Keeping the lock record elements for all locks held at the same time, and
- Avoiding too much false contention between locks on different resources.

The number of lock table entries and record elements are shown in the results for comparison with the related cluster nucleus message (ADAX70) and to aid your own calculations.

The actual lock structure size value is dependent on coupling facility internals and may vary across different coupling facility levels. If the estimates by this structure size calculator are too far off the real allocations reported by cluster nuclei, you can tweak the lock record element size value to bring the calculator more in line with the actual coupling facility being used.

➤ To access and use the space calculator for the lock structure in a cluster environment

- 1 Select option **S** on the **Basic Services Main Menu** in AOS.

The **Space Calculation** menu appears.

- 2 Select option **C** on the **Space Calculation** menu.

The **Cache Structure Calculator** panel appears.

```

14:17:34          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
DBID 1955          -  Cache Structure Calculator  -          PSPC002

Smallest block size in DB ..... 2544
Largest block size in DB ..... 5724
Buffer pool size (LBP) ..... 80896_____
Size proper for caching blocks .. 100000_____
Max nuclei in cluster ..... 3
Directory element size ..... 400
Cache structure size (in KB) .... _____

For minimum calculation, leave cache structure size field empty.
Modify values, press Enter to provide estimates below.

Cache CFRM SIZE/INITSIZE ..... 2650          ( 2.5          MB)
ADARUN DIRRATIO ..... 67
ADARUN ELEMENTRATIO ..... 48
Cache directory elements ..... 135
Cache data elements ..... 97
Cache data element size ..... 1024

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help          Exit          Lock          Menu          ↵
    
```

3 Press PF4 to access the **Lock Structure Calculator** panel:

```

14:16:57          ***** A D A B A S  BASIC  SERVICES *****          2014-05-02
DBID 1955          -  Lock Structure Calculator  -          PSPL002

Max files in database (MAXFILES) ..... 1000
Max number of parallel users (NU) ..... 200_____
Number of hold queue elements (NH) .... 800
Number of threads (NT) ..... 25
Unique descriptor pool size (LDEUQP) .. 50000
Lock record element size ..... 260
Lock structure size (in KB) .....

For minimum calculation, leave lock structure size field empty.
Modify values, press Enter to provide estimates below.

Lock CFRM SIZE/INITSIZE ..... 1478786      ( 1444.1      MB)
Number of lock table entries ..... 16777216
Number of lock record elements ..... 5305996      Required min .. 5240000

PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help          Exit          Cache          Menu          ↵
    
```

4 Specify values for the following fields on the **Lock Structure Calculator** panel.

Field	Description	Default
Max files in database	Specify a value between 3 and 5000. This is the same as the MAXFILES parameter of the ADADEF and ADAORD utilities.	The MAXFILES parameter setting of the current AOS database ID.
Max number of parallel users	Specify a value between 20 and 16,777,215.	The NU parameter setting of the current AOS database ID.
Number of hold queue elements	Specify a value between 20 and 16,777,215. .	The NH parameter setting of the current AOS database ID.
Unique descriptor pool size	Specify a value between 1 and 999,999,999.	The LDEUQP parameter setting of the current AOS database ID.
Lock record element size	Specify a value between 100 and 999. This parameter specifies how much space (including the overhead for the access paths) will be used by each lock record element in the lock structure.	260
Lock structure size	Specify a blank for the minimum calculation or specify a value between 100 and 999,999,999 (KB). Although this value is given as an output field, you may want to propose a lock structure size, to see how to allocate the lock table entries and lock table elements.	blank

- 5 Press Enter after all values in the previous step are specified.

The following output fields on the screen are filled.

Field	Description
Lock CFRM SIZE/INITSIZE	The recommended lock structure SIZE or INITSIZE specification in the coupling facility resource management policy.
Number of lock table entries	The calculated count of lock table entries resulting from the SIZE/INITSIZE setting.
Number of lock record elements	The estimated count of lock record elements resulting from the SIZE/INITSIZE setting. You must actually start a cluster nucleus with the specified parameters to see how many lock record elements it gets from the lock structure. The number on the right side is the minimum number of lock record elements required to be available by the starting cluster nuclei.

Maintain the User Table



Note: This option is available in Adabas nucleus cluster environments only.

A new function has been added to the **Session Opercoms** menu to support the CLUFREEUSER command. When option V (User Table Maintenance) is selected, the following screen is displayed:

```

15:10:47          ***** A D A B A S BASIC SERVICES *****          2014-05-02
                  - User Table Maintenance -                          PACIV02

                  Code      Service
                  ----      -
                  C         Begin CLUFREEUSER process
                  ?         Help
                  .         Exit
                  ----      -

Code ..... _
TNA ..... 0_____
UID ..... _____
Force ..... _
Global ..... _

Database ID .. 1955 (WIS1955)          NucID .. 1021

Command ==>
PF1----- PF2----- PF3----- PF4----- PF6----- PF7----- PF8----- PF12-----
Help           Exit           Menu           ↵
    
```

The CLUFREEUSER command is only valid in cluster environments. It can be issued against the local nucleus only or, with the Global option, against all active and inactive nuclei in the cluster.

The command is used to delete leftover user table elements (UTES) in common storage that are no longer associated with user queue elements (UQEs) in a nucleus where

TNA	is a decimal number specifying the timeout value in seconds. UTEs that are not used during the time specified may be deleted if other conditions are fulfilled. If TNA is not specified, UTEs may be deleted without regard to their recent use.	
UID	is a character string or hexadecimal byte string as follows:	
	ccccccc	where the argument is 1-8 letters, digits, or embedded '-' signs without surrounding apostrophes.
	'ccccccc'	where the argument is 1-8 characters with surrounding apostrophes.

	X'xxxxxxxxxxxxxxxx'	where the argument is an even number of 2-16 hexadecimal digits enclosed by 'X'.
	<p>A character string must be enclosed in apostrophes if it contains characters other than letter, digits, or embedded '-' signs. If a specified character string is less than 8 characters long, it is implicitly padded with blanks. If a specified hexadecimal string is shorter than 16 hexadecimal digits, it is implicitly padded with binary zeros.</p> <p>If the last 8 bytes of a user's 28-byte communication ID match a specific user ID or user ID prefix, that user's UTE may be deleted if other conditions are fulfilled.</p> <p>If UID not specified, UTEs may be deleted regardless of their user IDs.</p>	
FORCE	Delete leftover UTEs even if the users are due a response code 9 (ADARSP009), subcode 20. If FORCE is not specified, such UTEs are not deleted. Before using the FORCE parameter, ensure that the users owning the UTEs to be deleted will not expect any of their transactions to remain open. Specify FORCE on this screen by marking the Force field with any character.	
GLOBAL	Delete leftover UTEs throughout the Adabas cluster if they are no longer associated with UQEs and are eligible according to the other specified parameters. Additionally and subject to the other rules, delete leftover UTEs if their assigned nuclei have terminated since their last use. If GLOBAL is not specified, only UTEs assigned to the local nucleus and used since the nucleus start are eligible for deletion. Specify GLOBAL on this screen by marking the Global field with any character.	

Index

A

- Adabas Online System (AOS) screens
 - cluster usage, 100
 - displaying cluster members, 94
 - estimating cache structure sizes, 107
 - estimating lock structure sizes, 111
 - in a cluster environment, 93
 - maintaining the user table, 114
 - nucleus file status, 96
 - nucleus state flags, 97
 - scrolling through nucleus IDs in a cluster, 94
- Adabas Parallel Services
 - starting, 7
 - stopping, 13
- Adabas Recovery Aid (ADARAI), 36
- Adabas SMF records, 69
- ADACHK utility
 - PPTPRINT function, 34
- ADACOM
 - abnormal termination, 15
 - initialization process, 8
 - normal termination, 14
 - starting, 8
- ADADBS OPERCOM commands, 34
- ADADBS utility
 - REFRESHSTATS function, 35
- ADAPLP utility
 - IPLOGPRI function, 36
- ADAREP utility, 36
- ADARES BACKOUT utility function, 45
- ADARES REGENERATE utility function, 45
- ADARES utility
 - CLCOPY function, 36
 - MERGE CLOG function, 44
 - PLCOPY function, 38
- ADARUN parameters
 - determining correct settings, 62
- ADARUN settings, 62
- ADASAV RESTPLOG utility function, 47
- ADASAV utility, 46
- alert settings
 - cancel alerts, 18
 - messaging performance statistics, 21
 - overview, 17
 - self-termination alerts, 19
 - self-termination query alerts, 20
- archive recovery, 31
- ARM

- see automatic restart management (ARM), 31
- ASMFREC mapping macro, 89
- ASStI subtype, 71
- ASStP subtype, 72
- ASStT subtype, 71
- automatic restart management (ARM), 31
- autorestart
 - session, 30

B

- backout processing, 27

C

- cache and lock size, estimating, 66
- cache statistics, 55
- cache structure sizes, 107
- cancel alerts, 18
- checkpoint information extended, 36
- checkpoints
 - uniquely identifying, 45, 47
- CLnn command
 - close a nucleus to new users, 62
- close (CL) commands, 62
- Cluster
 - display usage statistics
 - using Basic Services, 100
- cluster
 - maintain user table, 114
- Cluster nuclei
 - collaboration between, 23
- Collaboration
 - between cluster nuclei, 23
- command logs
 - merging multiple, 36, 44
- command statistics, 52
- commands
 - routing, 35
- COMPRINT
 - messages written to, 8

D

- data set activity statistics, 59
- deferred publishing, 63

E

efficiency statistics, 54
 estimating cache structure sizes, 107
 estimating lock structure sizes, 111

G

global cache and lock areas, 64
 global commands, 34
 global lock statistics, 57

I

IBM type 89 SMF records, 91
 initialization, 7
 ADACOM, 8
 nucleus, 9
 PPT processing, 10
 sequence, 8

L

lock structure sizes, 111

M

messaging performance statistics, 21
 messaging statistics, 55
 minimizing communication with global areas, 66

N

NUCID
 ADADBS utility OPERCOM parameter
 for command routing, 35
 Nucleus
 routing utility-issued operator commands, 35
 nucleus
 abnormal termination, 15
 initialization process, 9
 normal termination, 14
 serialization during initialization, 9
 starting, 8
 Nucleus file
 display status
 using Basic Services, 96

O

offline recovery, 30
 operator commands
 self-termination query alerts, 20
 optimizing block sizes, 67
 optimizing global cache and lock areas, 64

P

parallel participant table
 print/dump using utility, 34
 Parameters
 ADADBS OPERCOM
 NUCID, 35

performance and tuning, 61
 ADARUN settings, 62
 allocation Work data set space, 62
 deferred publishing, 63
 minimizing communication with global areas, 66
 optimizing block sizes, 67
 optimizing global cache and lock areas, 64
 tuning buffer flushes, 64
 using close (CL) commands, 62
 PPT processing, 10
 protection logs
 merging multiple, 38

R

recovery
 archive, 31
 automatic restart management (ARM), 31
 offline, 30
 online, 30
 session autorestart, 30
 restart/recovery processing, 29
 routing commands
 to a specific nucleus, 35
 to all cluster nuclei, 35

S

self-termination alerts, 19
 self-termination operator query, 20
 serialization
 of nucleus, 9
 Session
 display status
 using Basic Services, 97
 reset statistical values for
 using utility, 35
 session end statistics, 49
 SMF records, 69
 Adabas command activity section, 77
 Adabas global cache activity by Adabas file number section,
 78
 Adabas global cache activity by block type section, 77
 Adabas global cache activity section, 79
 Adabas global lock activity section, 80
 Adabas internucleus messaging control block activity sec-
 tion, 81
 Adabas internucleus messaging counts section, 81
 Adabas internucleus messaging service time histogram
 section, 82
 Adabas Parallel Services cache activity section, 83
 Adabasa file activity section, 77
 ADARUN parameter value section, 84
 ASMFREC mapping macro, 89
 header section, 74
 I/O by DD name section, 85
 IBM type 89, 91
 product ID section, 76
 record sections, 72
 record size limites, 70
 record structure, 70
 record subtypes, 71
 self-defining section, 75
 session statistics section, 86

- SMF user exit, 90
- statistics recording, 72
- storage pool section, 87
- subtype 1, Adabas nucleus initialization, 71
- subtype 2, Adabas nucleus termination, 71
- subtype 3, Adabas interval statistics, 71
- subtype 4, Adabas parameter change, 72
- thread activity section, 88
- zIIP statistics section, 88
- SMF user exit, 90
- space allocation, 62
- statistics, 49
 - ASSO, DATA, I/O distribution, 51
 - buffer flush information, 54
 - cache, 55
 - command, 52
 - command distribution by file, 53
 - command distribution by source, 52
 - command distribution by thread, 52
 - command distribution by type, 53
 - data set activity, 59
 - efficiency, 54
 - general nucleus information, 50
 - global lock, 57
 - high-water marks, 54
 - I/O counts, 50
 - input/output, 50
 - log reads and buffer efficiency, 51
 - messaging, 55
 - user sessions, 53
- System
 - display status
 - using Basic Services, 97

- WORK data set I/O statistics, 58

T

- termination, 13
 - abnormal, 14
 - ADACOM, 14
 - normal, 14
 - nucleus, 14
- timeout settings
 - cancel alerts, 18
 - messaging performance statistics, 21
 - overview, 17
 - self-termination alerts, 19
 - self-termination query alerts, 20
- tuning buffer flushes, 64

U

- user sessions, 53
- user table
 - maintain, 114
- Utilities
 - ADADBS OPERCOM
 - NUCID parameter, 35
- utilities
 - cluster environment, 34
- utility processing, 33

W

- Work data set allocation, 62

