Adabas

Operations

Version 8.6.2

October 2026

ADABAS & NATURAL

This document applies to Adabas Version 8.6.2 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 © 1971-2026 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: ADAMF-AADAMFOPERATOR-862-20261013

Table of Contents

Operations xi
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
I 5
2 Conventions 7
Syntax Conventions 8
Version 10
3 Controlling the Adabas Environment 11
II Adabas Initialization (ADARUN Statement) 13
4 ADARUN Parameter Syntax 15
5 ADARUN Parameter Directory 17
ALOGDEV Parameter: Multiple Audit Logs Device Type 49
ALOGSIZE Parameter: Multiple Audit Log Size 50
AOSLOG Parameter: Log Active Nucleus Update Commands 50
AREXCLUDE Parameter: Exclude Files from Autorestart Processing 51
ARMNAME Parameter: Activate Automatic Restart Management (ARM) Program 51
ARNWORKBUFFERS Parameter: Autorestart WORK I/O Buffer Controls 52
ASSOCACHE Parameter: Associator Controller Caching 53
ASSOSPACEWARN Parameter: Associator Space Notifications 54
ASYTVS Parameter: Asynchronous Buffer Flush 55
AUDITING Parameter: Enable Nucleus Auditing 55
AUDITLOG Parameter: Audit of AOS and Utility Functions 56
AUTOCQENV Parameter: Environment Affected by Command Return from the Command Queue 56
AUTOCQTIME Parameter: Wait Time Threshold for Commands in Command Queue during Online Recovery 57
AUTOINCASSOSIZE Parameter: Associator Automatic Database Increase Size 58
AUTOINCASSOTHRESHOLD Parameter: Associator Space Threshold 59
AUTOINCASSOTOTAL Parameter: Associator Space Maximum Size 60
AUTOINCDATASIZE Parameter: Data Storage Automatic Database Increase Size 61
AUTOINCDATATHRESHOLD Parameter: Data Storage Space Threshold 63
AUTOINCDATATOTAL Parameter: Data Storage Space Maximum Size 64
CACHE Parameter: Adabas Caching Facility Control 65
CACTIVATE Parameter: Adabas Caching Facility RABN Activation 65

CASSODSP / CASSOEXT / CASSOHSP / CASSOG64 / CASSOL64 / CASSOP64 /CASSOV64 Parameters: Adabas Caching Facility Associator RABN Cache Space Control 66
CASSOMAXS Parameter: Associator Cache Space Unit Size 68
CBUFNO Parameter: Number of Read-Ahead Buffers for Concurrent I/O Processing 69
CCTIMEOUT Parameter: Cache Space Area or RABN Inactivity Time 70
CDATADSP / CDATAEXT / CDATAHSP / CDATAG64 / CDATAL64 / CDATAP64 / CDATAV64 Parameter: Adabas Caching Facility Data Storage RABN Cache Space Control 72
CDATAMAXS Parameter: Data Storage Cache Space Unit Size 74
CDEMAND Parameter: Lowest Adabas Buffer Efficiency for Caching 75
CDISPSTAT Parameter: Adabas Caching Facility RABN Range Statistics Display Control 76
CDXnn Parameter: Collation Descriptor User Exit 77
CEXCLUDE Parameter: Adabas Caching Facility Excluded Command Types 78
CFILE Parameter: Adabas Caching Facility File Range 79
CLOGBMAX Parameter: Logged Buffer Size Limit 81
CLOGDEV/CLOGSIZE Parameter: Multiple Command Log Device Type / Log Size 81
CLOGLAYOUT Parameter: Command Logging Format 82
CLOGMAX Parameter: Total Logged Buffer Size Limit for a Command 83
CLOGMRG Parameter: Cluster Command Log Merge 83
CLUCACHEEXTRA Parameter: Adjust Shared Cache Vector Size 85
CLUCACHENAME Parameter: Cluster Cache Structure/Area Name 85
CLUCACHESIZE Parameter: Size of the Global Cache Area 86
CLUCACHETYPE Parameter: Global Cache Area Storage Type Option 87
CLUCACHEUNCHANGED Parameter: Global Cache Unchanged Block Option 89
CLUGROUPNAME Parameter: Cluster Group Name 90
CLULOCKNAME Parameter: Cluster Lock Structure/Area Name 91
CLULOCKSIZE Parameter: Size of the Global Lock Area 92
CLUPUBLPROT Parameter: Publishing of Protection Data 92
CLUSTER Parameter: Adabas Nucleus Cluster Session Control 93
CLUWORK1CACHE Parameter: Number of WORK Blocks in Global Cache 94
CMAXCSPS Parameter: Adabas Caching Facility Storage Area Quantities 95
CRETRY Parameter: Adabas Caching Facility Retry Time 96
CSTORAGE Parameter: Adabas Caching Facility Override Cache Type 97
CT Parameter: Command Timeout Limit 98
CWORKSTORAGE Parameter: Adabas Caching Facility Work Area Cache Type 98

CWORK2FAC Parameter: Adabas Caching Facility Work Part 2 Percentage 99
CWORK3FAC Parameter: Adabas Caching Facility Work Part 3 Percentage 100
CXFILE Parameter: Adabas Caching Facility Excluded Files 101
DATACACHE Parameter: Data Storage Controller Caching 102
DATASPACEWARN Parameter: Data Storage Space Notifications 103
DBID Parameter: Database Identification 105
DEVICE Parameter: Device Type 105
DIRRATIO / ELEMENTRATIO Parameters: Cluster Cache Directory-to-Data Elements Ratio 105
DSF Parameter: Adabas Delta Save Facility Control 106
DSFEX1 Parameter: Delta Save Facility Exit 107
DTP Parameter: Distributed Transaction Processing Control 108
DUALALD Parameter: Dual Audit Log Device Type 109
DUALALS Parameter: Dual Audit Log Size 109
DUALCLD/DUALCLS Parameter: Dual Command Log Device Type / Log Size 110
DUALPLD/DUALPLS Parameter: Dual Protection Log Device Type / Log Size 111
ENCRYPTION Parameter: Activate Adabas Encryption 112
EXCPVR Parameter: Control EXCP or EXCPVR Use 112
FASTPATH Parameter: Adabas Fastpath Control 113
FMXIO Parameter: Parallel LFIOP I/O Operations 113
FORCE Parameter: Allow Nucleus Database ID or Review Hub Table Entry Overwrite 114
HEXnn Parameter: Hyperdescriptor User Exit 115
IGNDIB Parameter: Ignore DIB Entry for Adabas Nucleus 116
IGNDTP Parameter: Ignore Two-Phase Commit Area (Work Part 4) 117
INDEXCROSSCHECK Parameter: Data Storage Record Validation 118
INDEXUPDATE Parameter: Index Update Processing Control 119
INFOBUFFERSIZE Parameter: Information Buffer Pool (Adabas Event Log) Size 119
INTAUTO Parameter: Time Interval Between Autorestart Progress Messages 120
INTNAS Parameter: Time Interval Between SYNS 60 Checkpoints 121
LAP Parameter: Adabas Auditing Pool Size 121
LARGEPAGE Parameter: Large Page Support Option 122
LBP Parameter: Length of Buffer Pool 123
LCP Parameter: Length of Security Pool 125
LDEUQP Parameter: Length of Unique Descriptor Pool 125
LDTP Parameter: Length of Distributed Transaction Processing Area (Work Part 4) 126
LFIOP Parameter: Asynchronous Buffer Pool Control 127
LFP Parameter: Length of Internal Format Buffer Pool 128

LI Parameter: Length of Table of ISNs (TBI) 129
LNKGNAME Parameter: Link Globals Table Name 130
LOCAL Parameter: Local Adabas Review Hub 130
LOGGING / LOGxxxx Parameters: Command Logging Control 131
LOGSIZE Parameter: Maximum Command Log Size 132
LOGWARN Parameter: CLOG and PLOG Alert Message Interval 133
LP Parameter: Length of Data Protection Area (Work Part 1) 134
LQ Parameter: Length of Sequential Command Table 135
LRDP Parameter: Length of the Redo Pool 135
LRPL Parameter: Adabas or Event Replicator Replication Pool Size 137
LS Parameter: Length of Internal Sort Area 138
LSF Parameter: Logical Save Facility 139
LTZ Parameter: Time Zone Pool Size 139
LU Parameter: Length of Intermediate User Buffer Area 140
LWKP2 Parameter: Length of ISN List Processing Area (Work Part 2) 141
LWP Parameter: Length of Adabas Work Pool 142
MLWTO Parameter: Multi-Line Console Message Control 143
MODE Parameter: Mode of Operation 143
MSGBUF Parameter: Size of Message Buffer 144
MSGCONSL/MSGDRUCK/MSGPRINT Parameter: Case of Messages Sent to the Console/DDDRUCK /DDPRINT 145
MXCANCEL Parameter: Timeout Threshold for Cancelled Peer Nucleus 146
MXCANCELWARN Parameter: Timeout Threshold for Cancelled Peer Nucleus Warning 147
MXMSG Parameter: Timeout Threshold for Internucleus Command Processing 148
MXMSGWARN Parameter: Timeout Threshold for Internucleus Command Processing Warning 149
MXSTATUS Parameter: Member-Level XCF Status Monitoring Heartbeat Interval 150
MXTNA Parameter: Maximum Non-activity Time 151
MXTSX Parameter: Maximum Sx Command Execution Time 152
MXTT Parameter: Maximum Transaction Time 153
MXWTOR Parameter: Self-Termination Operator Query Interval 154
NAB Parameter: Number of Attached Buffers 155
NALOG Parameter: Number of Audit Log Data Sets 156
NC Parameter: Number of Command Queue Elements 157
NCLOG Parameter: Number of Command Log Data Sets 158
NH Parameter: Number of Hold Queue Elements 159
NISNHQ Parameter: Number of ISNs in Hold Queue per User 160
NONDES Parameter: Nondescriptor Searches 161
NPLOG Parameter: Number of Protection Log Data Sets 161
NPLOGBUFFERS Parameter: PLOG Protection I/O Buffer Control 162
NQCID Parameter: Number of Active Command IDs per User 163

NSISN Parameter: Number of ISNs per ISN Table Element 164
NT Parameter: Number of Threads 165
NU Parameter: Number of User Queue Elements 166
NUCID Parameter: Adabas Cluster Nucleus ID 167
NWORK1BUFFERS Parameter: Work Part 1 Protection I/O Buffer Control 168
OPENRQ Parameter: Open Command Required 168
PGFIX Parameter: EXCPVR Page Fixing Control 169
PLOGDEV/PLOGSIZE Parameter: Multiple Protection Log Device Type / Size 169
PLOGRQ Parameter: Protection Log Required 170
PREFETCH Parameter: Activate Prefetch Feature 171
PREFICMD/PREFIFIL Parameter: Include Commands/Files in Prefetch/Multifetch Processing 172
PREFNREC Parameter: Multifetch Record Count 173
PREFSBL Parameter: Prefetch Buffer Length per Call 174
PREFSTDD Parameter: Multifetch Statistics DD Name 175
PREFTBL Parameter: Total Size of Prefetch Buffer 178
PREFXCMD (PREFXFIL) Parameters: Exclude Commands (Files) from Prefetch/Multifetch Processing 179
PROGRAM Parameter: Program to Run 180
QBLKSIZE Parameter: Block Size for Sequential Data Sets 182
READONLY Parameter: Read-Only Session Control 183
REFSTPRT Parameter: Print Statistics before Refresh Statistics 184
REPLICATION Parameter: Replication Control 184
REVBUFMB Parameter: Buffer Pool Size in Megabytes 185
REVFILTER Parameter: Review Record Filtering Control 186
REVIEW Parameter: Adabas Review Control 186
REVLOGBMAX Parameter: Logged Buffer Size Limit for Review 188
REVLOGMAX Parameter: Total Logged Buffer Size Limit for a Review Command 188
REVLOGNR Parameter: Maximum Number of Records per Buffer 189
REVSTHMV Parameter: Send to HUB Mode Version 190
REVTIMER Parameter: Send Interval in Seconds 190
RIAFTERUPDATE Parameter: Controlling RI Command Behavior 191
RPLCONNECTCOUNT Parameter: Replication Connection Attempt Count 192
RPLCONNECTINTERVAL Parameter: Replication Connection Attempt Interval 192
RPLPARMS Parameter: Replication Parameter Location 192
RPLSORT Parameter: Event Replicator for Adabas Transaction Data Sorting Control 193
RPWARNINCREMENT Parameter: Replication Pool Warning Message Interval 194
RPWARNINTERVAL Parameter: Replication Pool Usage Interval 195

RPWARNMESSAGELIMIT Parameter: Replication Pool Usage Warning Message Count 195
RPWARNPERCENT Parameter: Replication Pool Usage Threshold 196
RVCLIENT Parameter: Adabas Review Client Reporting Activation 196
SECUID Parameter: Security System User ID Requirement Level 197
SMF Parameter: Enable Adabas SMF Recording 198
SMF89 Parameter: Enable SMF Type 89 Records 198
SMFDETAIL Parameter: Selecting Detail SMF Record Sections 199
SMFINTERVAL Parameter: Enabling Adabas SMF Interval Records 201
SMFRECNO Parameter: Selecting User-Defined Adabas SMF Record Number 202
SMFSUBSYS Parameter: Specifying SMF Subsystem Name 202
SMGT Parameter: Error Handling and Message Buffering Facility 203
SORTCACHE Parameter: Sort Area Controller Caching 203
SPT Parameter: Activate Adabas Triggers and Stored Procedures 204
SRLOG Parameter: Controlling Spanned Record Logging 205
SVC Parameter: SVC Number 205
TARGETID Parameter: Entire Net-Work Target ID 206
TCPIP Parameter: TCP/IP Access Control 206
TCPURL Parameter: TCP/IP Universal Resource Locator 207
TEMPCACHE Parameter: Temp Area Controller Caching 208
TFLUSH Parameter: Synchronous Buffer Flush Duration 209
TLSCMD Parameter: Time Limit for S1, S2, and S4 Complex Searches 209
TMDRQ Parameter: Size of ATM Request Queue 210
TMETDATA Parameter: ET Data Control 211
TMGTT Parameter: Global Transaction Time Limit 212
TMLOG Parameter: ATM Manager Logging Control 213
TMMSGSEV Parameter: Severity Threshold for Warning Messages 214
TMRESTART Parameter: ATM Restart Control 214
TMSYNCMGR Parameter: Support for External Transaction Coordinators 215
TMTCIDPREF Parameter: Dynamic Client ID Prefix 215
TNAA Parameter: Nonactivity Time Limit for Access-Only Users 216
TNAE Parameter: Nonactivity Time Limit for ET Logic Users 217
TNAX Parameter: Nonactivity Time Limit for Exclusive Control Users 217
TT Parameter: Transaction Time Limit 218
UEXn Parameter: User Exit 219
UEXSMF Parameter: Specifying SMF User Exit Module Name 220
UPDATECONTROL Parameter: Control Scheduling of Update Commands During Buffer Flush 220
UTIONLY Parameter: Utilities-Only Session Control 221
V64BIT Parameter: 64-Bit Virtual Storage Option 222
VISTA Parameter: Adabas Vista Control 223
WORKCACHE Parameter: Work Area Controller Caching 223
ZIIP Parameter: Activate Usage of Adabas for zIIP 224

6 Setting Adabas Session Parameters 227
General Guidelines 228
Using Session Statistics to Determine Parameter Settings 232
Parameters for Operating Systems 232
7 ADARUN Parameters Pertinent to Adabas Add-on Products 233
Adabas Caching Facility ADARUN Parameters 234
Adabas Fastpath ADARUN Parameters 235
Adabas Cluster Services ADARUN Parameters 235
Adabas Delta Save Facility ADARUN Parameters 236
Adabas Parallel Services ADARUN Parameters 236
Adabas Review ADARUN Parameters 237
Adabas Transaction Manager ADARUN Parameters 238
Adabas Vista ADARUN Parameters 239
Event Replicator for Adabas ADARUN Parameters 239
Adabas Auditing ADARUN Parameters 240
III Other Operations 241
8 Operator Commands 243
Entering Operator Commands on z/OS Systems 244
Operator Command Groupings 245
Nucleus Commands 245
Adabas Utility Operator Command DSTAT 317
SMGT Operator Command 318
SMGT Display Output Samples 323
9 Adabas Session Execution 327
Program Synchronization Using z/OS Session System Services 328
Single-User Mode 335
User Profile 335
10 Adabas Utility Execution 339
Condition and Termination Codes 340
Utility Usage of Files and Databases 341
Disk Accesses by Utility and Function 344
Possible Resource Conflicts 346
11 Linking Applications to Adabas 349
How the Adabas API Works 350
Available Link Routines 352
Required Work Area 353
Required Application Reentrancy Properties 354
Adabas Control Block (ACB) Options 356
Extended Adabas Control Block (ACBX) Options 358
Programming Conventions for Issuing Direct Calls 359
Using the CICS COMMAREA or TWA with COBOL Programs 365
Using the Adabas API in Batch Mode 368
12 Adabas Restart and Recovery 371
Work Data Set 372
Protection Log 373

Restart Operations 375
Database Recovery 377
13 Local Time Zone Change Support 383
Time Stamp Overview 384
Activating the Local Time Zone Change 384
14 Adabas Timeout Feature 385
Timeout Parameters 386
Action Taken If Time Limit Exceeded 387
Index 389

Operations

This documentation contains information needed to operate the Adabas nucleus, Adabas utilities and user programs that call Adabas.

The following summarizes the topics covered by this documentation:

Conventions	Explains the syntax conventions and notation for the version.
Controlling the Adabas Environment	Provides an overview of the Adabas operating structure.
Securing the Adabas Environment	Provides a description of the security measures that are available to prevent unauthorized access to the Adabas environment.
Software AG Mainframe Product Licensing	Describes Software GmbH's mainframe product license support and processing.
Adabas Initialization (ADARUN)	Describes the use of ADARUN control parameters. ADARUN is the Adabas control module and the control statement name. The ADARUN control statement invokes the Adabas nucleus and the Adabas utilities, and defines the Adabas operating environment. User programs are executed according to normal operating system conventions, and communicate with the nucleus in various ways depending on the operating system environment. Batch and TSO user programs normally communicate with Adabas through ADARUN.
Operator Commands	Describes Adabas operator commands that can be entered on the console during an Adabas session or during utility operation to terminate an Adabas or user session; display nucleus or utility information; log commands into CLOG; or change Adabas operating parameters or conditions. Commands are listed alphabetically. One command, DSTAT, is listed twice: once as a command for displaying nucleus status, and separately as a command to display current Adabas utility operating status. For Adabas Caching Facility operator commands, see the Adabas Caching Facility documentation. For Adabas Parallel Services operator commands, see the Adabas Parallel Services documentation. For Adabas Cluster Services operator commands, see the Adabas Cluster Services documentation.
Adabas Session Execution	Lists the job control statements required to execute a session under z/OS, and shows examples of the job streams. For examples of utilities jobs, see the Adabas Utilities documentation.
Adabas Utility Execution	Describes conditions and requirements for running the Adabas utilities, which are described in detail in the Adabas Utilities documentation .
Linking Applications to Adabas	Describes the Adabas link/router APIs and their use, the requirements for single- and multiuser program execution, and the job structure requirements.

	Since most systems do not allow a standard call to Adabas, Software GmbH provides an application program interface (API) to translate calls issued by an application program into a form that can be handled by Adabas. Batch applications are supported in both single- and multiuser mode; online operations are controlled by teleprocessing (TP) monitors. The Adabas API is available across all supported mainframe platforms; versions of the API that are specific to particular TP monitors are provided. *Adalink* is a generic term that refers to the portion of the API that is specific to a particular TP monitor.
Adabas Restart and Recovery	Provides information and recommended procedures related to restart and recovery.
Local Time Zone Change Support	Provides information on changing the local time zone.
Adabas Timeout Feature	Describes the Adabas time-out feature, a control mechanism for monitoring and restoring unavailable or locked system resources.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	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.softwareag.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.

I

- 2 Conventions 7
- 3 Controlling the Adabas Environment 11

2 Conventions

- Syntax Conventions 8
- Version 10

Syntax Conventions

The following table describes the conventions used in syntax diagrams of ADARUN parameters and operator commands.

Convention	Description	Example
uppercase, bold	Syntax elements appearing in uppercase and bold font are Adabas keywords. When specified, these keywords must be entered exactly as shown.	**ADADBS CHANGE FILE =** *file-number* The syntax elements ADADBS, CHANGE, and FILE are Adabas keywords.
lowercase, italic, normal font	Syntax elements appearing in lowercase and normal, italic font identify items that you must supply.	**ADADBS CHANGE FILE =** *file-number* The syntax element *file-number* identifies and describes the kind of value you must supply. In this instance, you must supply the number of the file affected by the ADADBS CHANGE operation.
mixed case, normal font	Syntax elements appearing in mixed case and normal font (not bold or italic) identify items established by other Adabas control statements. This notation is usually used to identify how default values are determined for some parameters in Adabas syntax.	**[SORTDEV = {** *device-type* | ADARUN-device **}]** The syntax element "ADARUN-device" indicates that the device type identified by the ADARUN DEVICE parameter will be used if a different device type is not specified. The literal "ADARUN-device" should *not* be specified for the SORTDEV parameter.
underlining	Underlining is used for two purposes: 1. To identify default values, wherever appropriate. Otherwise, the defaults are explained in the accompanying parameter descriptions. 2. To identify the short form of a keyword.	**[LRECL = {***record-buffer-length* | **4000 }]** In the example above, 4000 is the default that will be used for the LRECL parameter if no other record buffer length is specified. **DEVICE** In the example above, the short version of the DEVICE parameter is DE.

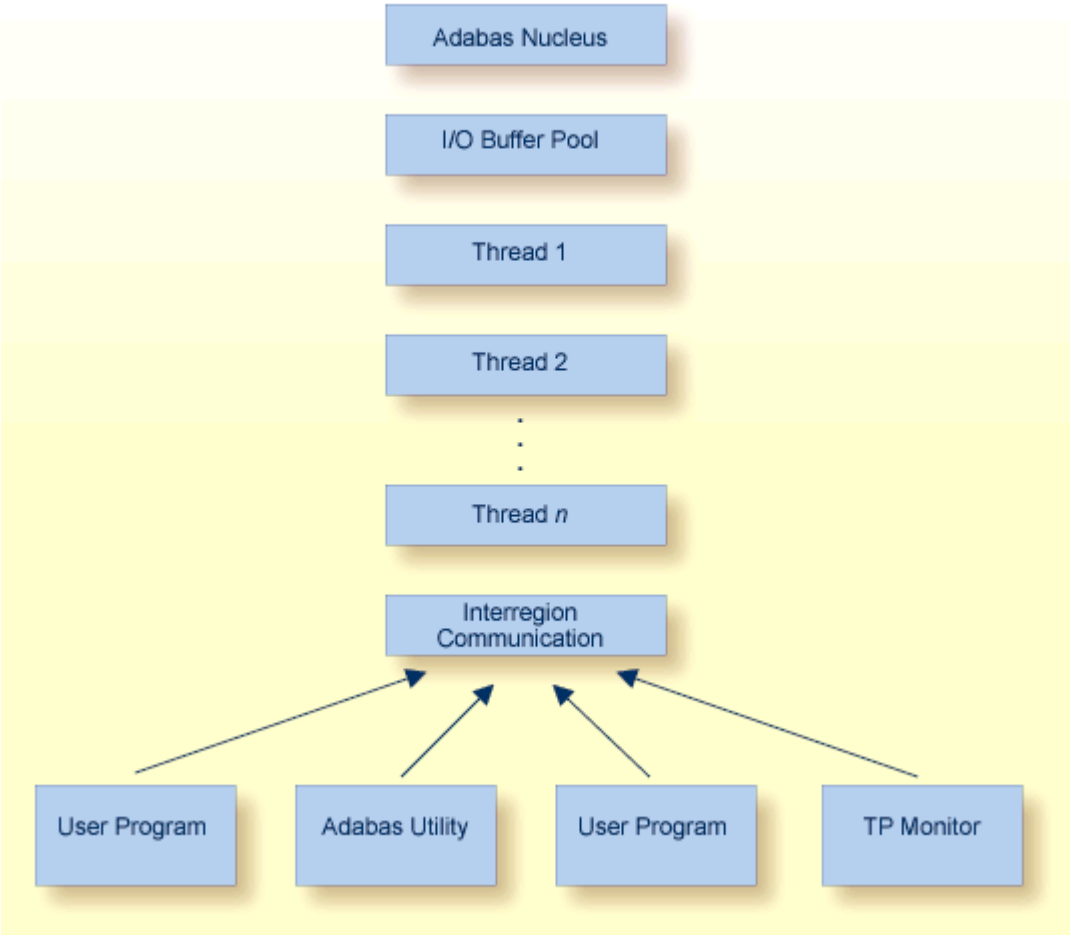
Convention	Description	Example
vertical bars (|)	Vertical bars are used to separate mutually exclusive choices. **Note:** In more complex syntax involving the use of large brackets or braces, mutually exclusive choices are stacked instead.	ADAORD { RESTRUCTUREF | REF } In the example above, you must select RESTRUCTUREF or REF for this ADAORD function. There are no defaults.
brackets ([ ])	Brackets are used to identify optional elements. When multiple elements are stacked or separated by vertical bars within brackets, only one of the elements may be supplied.	[SORTSEQ = { *descriptor* [,MU] [,NU] ISN [, STARTISN = *value*] physical-sequence }] In this example, the SORTSEQ parameter and the MU, NU, and STARTISN subparameters are optional. **Note:** Note that the mutually exclusive choices for the SORTSEQ parameter are stacked.
braces ({ })	Braces are used to identify required elements. When multiple elements are stacked or separated by vertical bars within braces, one and only one of the elements must be supplied.	{ SUSPEND [TTSYN = { *time-available-to-sync* | ADARUN-TT }] [TRESUME = { *time-until-resume* | 120 }] RESUME } In this example, either the SUSPEND or RESUME parameter is required.
indentation	Indentation is used to identify subparameters of a parameter.	{ SUSPEND [TTSYN = { *time-available-to-sync* | ADARUN-TT }] [TRESUME = { *time-until-resume* | 120 }] RESUME } In this example, TTSYN and TRESUME are subparameters of the SUSPEND parameter.
ellipsis (...)	Ellipses are used to identify elements that can be repeated. If the term preceding the ellipsis is an expression enclosed in square brackets or braces, the ellipsis applies to the entire bracketed expression.	[FIELD = ' *field-name* [, *option*]... '] ... In this example, the FIELD parameter can be repeated. In addition, more than one option can be associated with a field.

Convention	Description	Example
other punctuation and symbols	All other punctuation and symbols must be entered exactly as shown.	[FIELD = ' *field-name* [, *option*]... '] ... In this example, the single quotation marks must be specified around the field definitions and their associated options. In addition, options must be separated by commas.

Version

Notation `vrs`, `vr`, or `v`: When used in this documentation, the notation `vrs` or `vr` stands for the relevant version of a product. For further information on product versions, see *version* in the *Glossary*.

3 Controlling the Adabas Environment



Adabas Operating Structure

The Adabas *nucleus* manages all database activities; for example, data access/update.

The Adabas *utilities* are used to perform database service functions; for example, loading/deleting files.

A *user program* calls the Adabas nucleus in order to perform database access/update.

For controlling the Adabas environment, the database administrator can use either the optional Adabas Online System or the ADADBS utility.

The Adabas nucleus and each user program or Adabas utility are executed in a separate storage area defined by the operating system. On z/OS, this is an address space, dataspace, or hiperspace.

For consistency and simplification, this manual refers to these z/OS areas as *regions*.

II Adabas Initialization (ADARUN Statement)

The ADARUN control statement defines and starts the Adabas operating environment. The ADARUN control statement also starts Adabas utilities.

ADARUN:

- loads the ADAIOR module, which performs database I/O and other functions that depend on the operating system;
- interprets the ADARUN parameter statements, then loads and modifies the appropriate Adabas nucleus or utility modules according to the ADARUN parameter settings; and
- transfers control to Adabas.

The ADARUN statement, normally a series of entries each specifying one or more ADARUN parameter settings, is specified in the DDDDCARD data set. For more specific job information, refer to the appropriate location in this documentation.

Note: The ADARUN parameters described in this chapter apply to the standard Adabas configuration and to Adabas cluster configurations. Other ADARUN parameters must also be specified for such features as: Adabas Caching Facility, Adabas Delta Save Facility Facility, Adabas Transaction Manager. Refer to the related documentation for more information about the ADARUN parameters that may be required.

The ADARUN documentation is organized under the following headings:

ADARUN Parameter Syntax
ADARUN Parameter Directory
Setting Adabas Session Parameters
ADARUN Parameters Pertinent to Adabas Add-on Products

The ADARUN parameter directory provides a tabular overview of ADARUN parameters.

4 ADARUN Parameter Syntax

The syntax for the ADARUN statement and parameters is:

```
ADARUN  parameter = value,...
```

where *parameter*=*value* is one of the ADARUN parameters described in this section. Multiple *parameter*=*value* entries separated by commas may be included in a statement. The comma following the last *parameter*=*value* entry of a statement is ignored.

All *parameter*=*value* entries must end before position 73. Any *parameter*=*value* entries that would extend beyond position 72 must be coded on a new statement. Positions 73-80 are ignored.

The literal "ADARUN" must be entered in positions 1-6 of each ADARUN statement. Any number of blanks is permitted between "ADARUN" and the first parameter, but no blanks are permitted within the *parameter*=*value* string. A blank following a *parameter*=*value* entry indicates the end of the statement.

An asterisk (*) in position 1 indicates a user comment line.

The first ADARUN statement cannot continue beyond position 72. The second statement represents a continuation of the first statement. All ADARUN continuation statements have the same format and restrictions as the first statement.

The ADARUN parameters are summarized in the following table, and described in detail in the sections following the table.

Unless noted otherwise, each parameter has a default value that ADARUN uses if the parameter is not specified.

Some parameter names can be abbreviated. The minimum acceptable abbreviation for a given parameter is shown by the underlined part of the parameter name. Parameter names shown without underlining cannot be abbreviated.

5 ADARUN Parameter Directory

- ALOGDEV Parameter: Multiple Audit Logs Device Type 49
- ALOGSIZE Parameter: Multiple Audit Log Size 50
- AOSLOG Parameter: Log Active Nucleus Update Commands 50
- AREXCLUDE Parameter: Exclude Files from Autorestart Processing 51
- ARMNAME Parameter: Activate Automatic Restart Management (ARM) Program 51
- ARNWORKBUFFERS Parameter: Autorestart WORK I/O Buffer Controls 52
- ASSOCACHE Parameter: Associator Controller Caching 53
- ASSOSPACEWARN Parameter: Associator Space Notifications 54
- ASYTVS Parameter: Asynchronous Buffer Flush 55
- AUDITING Parameter: Enable Nucleus Auditing 55
- AUDITLOG Parameter: Audit of AOS and Utility Functions 56
- AUTOCQENV Parameter: Environment Affected by Command Return from the Command Queue 56
- AUTOCQTIME Parameter: Wait Time Threshold for Commands in Command Queue during Online Recovery 57
- AUTOINCASSOSIZE Parameter: Associator Automatic Database Increase Size 58
- AUTOINCASSOTHRESHOLD Parameter: Associator Space Threshold 59
- AUTOINCASSOTOTAL Parameter: Associator Space Maximum Size 60
- AUTOINCDATASIZE Parameter: Data Storage Automatic Database Increase Size 61
- AUTOINCDATATHRESHOLD Parameter: Data Storage Space Threshold 63
- AUTOINCDATATOTAL Parameter: Data Storage Space Maximum Size 64
- CACHE Parameter: Adabas Caching Facility Control 65
- CACTIVATE Parameter: Adabas Caching Facility RABN Activation 65
- CASSODSP / CASSOEXT / CASSOHSP / CASSOG64 / CASSOL64 / CASSOP64 /CASSOV64 Parameters: Adabas Caching Facility Associator RABN Cache Space Control 66
- CASSOMAXS Parameter: Associator Cache Space Unit Size 68
- CBUFNO Parameter: Number of Read-Ahead Buffers for Concurrent I/O Processing 69
- CCTIMEOUT Parameter: Cache Space Area or RABN Inactivity Time 70
- CDATADSP / CDATAEXT / CDATAHSP / CDATAG64 / CDATAL64 / CDATAP64 / CDATAV64 Parameter: Adabas Caching Facility Data Storage RABN Cache Space Control 72
- CDATAMAXS Parameter: Data Storage Cache Space Unit Size 74
- CDEMAND Parameter: Lowest Adabas Buffer Efficiency for Caching 75
- CDISPSTAT Parameter: Adabas Caching Facility RABN Range Statistics Display Control 76

- CDXnn Parameter: Collation Descriptor User Exit 77
- CEXCLUDE Parameter: Adabas Caching Facility Excluded Command Types 78
- CFILE Parameter: Adabas Caching Facility File Range 79
- CLOGBMAX Parameter: Logged Buffer Size Limit 81
- CLOGDEV/CLOGSIZE Parameter: Multiple Command Log Device Type / Log Size 81
- CLOGLAYOUT Parameter: Command Logging Format 82
- CLOGMAX Parameter: Total Logged Buffer Size Limit for a Command 83
- CLOGMRG Parameter: Cluster Command Log Merge 83
- CLUCACHEEXTRA Parameter: Adjust Shared Cache Vector Size 85
- CLUCACHENAME Parameter: Cluster Cache Structure/Area Name 85
- CLUCACHESIZE Parameter: Size of the Global Cache Area 86
- CLUCACHETYPE Parameter: Global Cache Area Storage Type Option 87
- CLUCACHEUNCHANGED Parameter: Global Cache Unchanged Block Option 89
- CLUGROUPNAME Parameter: Cluster Group Name 90
- CLULOCKNAME Parameter: Cluster Lock Structure/Area Name 91
- CLULOCKSIZE Parameter: Size of the Global Lock Area 92
- CLUPUBLPROT Parameter: Publishing of Protection Data 92
- CLUSTER Parameter: Adabas Nucleus Cluster Session Control 93
- CLUWORK1CACHE Parameter: Number of WORK Blocks in Global Cache 94
- CMAXCSPS Parameter: Adabas Caching Facility Storage Area Quantities 95
- CRETRY Parameter: Adabas Caching Facility Retry Time 96
- CSTORAGE Parameter: Adabas Caching Facility Override Cache Type 97
- CT Parameter: Command Timeout Limit 98
- CWORKSTORAGE Parameter: Adabas Caching Facility Work Area Cache Type 98
- CWORK2FAC Parameter: Adabas Caching Facility Work Part 2 Percentage 99
- CWORK3FAC Parameter: Adabas Caching Facility Work Part 3 Percentage 100
- CXFILE Parameter: Adabas Caching Facility Excluded Files 101
- DATACACHE Parameter: Data Storage Controller Caching 102
- DATASPACEWARN Parameter: Data Storage Space Notifications 103
- DBID Parameter: Database Identification 105
- DEVICE Parameter: Device Type 105
- DIRRATIO / ELEMENTRATIO Parameters: Cluster Cache Directory-to-Data Elements Ratio 105
- DSF Parameter: Adabas Delta Save Facility Control 106
- DSFEX1 Parameter: Delta Save Facility Exit 107
- DTP Parameter: Distributed Transaction Processing Control 108
- DUALALD Parameter: Dual Audit Log Device Type 109
- DUALALS Parameter: Dual Audit Log Size 109
- DUALCLD/DUALCLS Parameter: Dual Command Log Device Type / Log Size 110
- DUALPLD/DUALPLS Parameter: Dual Protection Log Device Type / Log Size 111
- ENCRYPTION Parameter: Activate Adabas Encryption 112
- EXCPVR Parameter: Control EXCP or EXCPVR Use 112
- FASTPATH Parameter: Adabas Fastpath Control 113
- FMXIO Parameter: Parallel LFIOP I/O Operations 113
- FORCE Parameter: Allow Nucleus Database ID or Review Hub Table Entry Overwrite 114
- HEXnn Parameter: Hyperdescriptor User Exit 115

- IGNDIB Parameter: Ignore DIB Entry for Adabas Nucleus 116
- IGNDTP Parameter: Ignore Two-Phase Commit Area (Work Part 4) 117
- INDEXCROSSCHECK Parameter: Data Storage Record Validation 118
- INDEXUPDATE Parameter: Index Update Processing Control 119
- INFOBUFFERSIZE Parameter: Information Buffer Pool (Adabas Event Log) Size 119
- INTAUTO Parameter: Time Interval Between Autorestart Progress Messages 120
- INTNAS Parameter: Time Interval Between SYNS 60 Checkpoints 121
- LAP Parameter: Adabas Auditing Pool Size 121
- LARGEPAGE Parameter: Large Page Support Option 122
- LBP Parameter: Length of Buffer Pool 123
- LCP Parameter: Length of Security Pool 125
- LDEUQP Parameter: Length of Unique Descriptor Pool 125
- LDTP Parameter: Length of Distributed Transaction Processing Area (Work Part 4) 126
- LFIOP Parameter: Asynchronous Buffer Pool Control 127
- LFP Parameter: Length of Internal Format Buffer Pool 128
- LI Parameter: Length of Table of ISNs (TBI) 129
- LNKGNAME Parameter: Link Globals Table Name 130
- LOCAL Parameter: Local Adabas Review Hub 130
- LOGGING / LOGxxxx Parameters: Command Logging Control 131
- LOGSIZE Parameter: Maximum Command Log Size 132
- LOGWARN Parameter: CLOG and PLOG Alert Message Interval 133
- LP Parameter: Length of Data Protection Area (Work Part 1) 134
- LQ Parameter: Length of Sequential Command Table 135
- LRDP Parameter: Length of the Redo Pool 135
- LRPL Parameter: Adabas or Event Replicator Replication Pool Size 137
- LS Parameter: Length of Internal Sort Area 138
- LSF Parameter: Logical Save Facility 139
- LTZ Parameter: Time Zone Pool Size 139
- LU Parameter: Length of Intermediate User Buffer Area 140
- LWKP2 Parameter: Length of ISN List Processing Area (Work Part 2) 141
- LWP Parameter: Length of Adabas Work Pool 142
- MLWTO Parameter: Multi-Line Console Message Control 143
- MODE Parameter: Mode of Operation 143
- MSGBUF Parameter: Size of Message Buffer 144
- MSGCONSL/MSGDRUCK/MSGPRINT Parameter: Case of Messages Sent to the Console/DDDRUCK /DDPRINT 145
- MXCANCEL Parameter: Timeout Threshold for Cancelled Peer Nucleus 146
- MXCANCELWARN Parameter: Timeout Threshold for Cancelled Peer Nucleus Warning 147
- MXMSG Parameter: Timeout Threshold for Internucleus Command Processing 148
- MXMSGWARN Parameter: Timeout Threshold for Internucleus Command Processing Warning 149
- MXSTATUS Parameter: Member-Level XCF Status Monitoring Heartbeat Interval 150
- MXTNA Parameter: Maximum Non-activity Time 151
- MXTSX Parameter: Maximum Sx Command Execution Time 152
- MXTT Parameter: Maximum Transaction Time 153
- MXWTOR Parameter: Self-Termination Operator Query Interval 154

- NAB Parameter: Number of Attached Buffers 155
- NALOG Parameter: Number of Audit Log Data Sets 156
- NC Parameter: Number of Command Queue Elements 157
- NCLOG Parameter: Number of Command Log Data Sets 158
- NH Parameter: Number of Hold Queue Elements 159
- NISNHQ Parameter: Number of ISNs in Hold Queue per User 160
- NONDES Parameter: Nondescriptor Searches 161
- NPLOG Parameter: Number of Protection Log Data Sets 161
- NPLOGBUFFERS Parameter: PLOG Protection I/O Buffer Control 162
- NQCID Parameter: Number of Active Command IDs per User 163
- NSISN Parameter: Number of ISNs per ISN Table Element 164
- NT Parameter: Number of Threads 165
- NU Parameter: Number of User Queue Elements 166
- NUCID Parameter: Adabas Cluster Nucleus ID 167
- NWORK1BUFFERS Parameter: Work Part 1 Protection I/O Buffer Control 168
- OPENRQ Parameter: Open Command Required 168
- PGFIX Parameter: EXCPVR Page Fixing Control 169
- PLOGDEV/PLOGSIZE Parameter: Multiple Protection Log Device Type / Size 169
- PLOGRQ Parameter: Protection Log Required 170
- PREFETCH Parameter: Activate Prefetch Feature 171
- PREFICMD/PREFIFIL Parameter: Include Commands/Files in Prefetch/Multifetch Processing 172
- PREFNREC Parameter: Multifetch Record Count 173
- PREFSBL Parameter: Prefetch Buffer Length per Call 174
- PREFSTDD Parameter: Multifetch Statistics DD Name 175
- PREFTBL Parameter: Total Size of Prefetch Buffer 178
- PREFXCMD (PREFXFIL) Parameters: Exclude Commands (Files) from Prefetch/Multifetch Processing 179
- PROGRAM Parameter: Program to Run 180
- QBLKSIZE Parameter: Block Size for Sequential Data Sets 182
- READONLY Parameter: Read-Only Session Control 183
- REFSTPRT Parameter: Print Statistics before Refresh Statistics 184
- REPLICATION Parameter: Replication Control 184
- REVBUFMB Parameter: Buffer Pool Size in Megabytes 185
- REVFILTER Parameter: Review Record Filtering Control 186
- REVIEW Parameter: Adabas Review Control 186
- REVLOGBMAX Parameter: Logged Buffer Size Limit for Review 188
- REVLOGMAX Parameter: Total Logged Buffer Size Limit for a Review Command 188
- REVLOGNR Parameter: Maximum Number of Records per Buffer 189
- REVSTHMV Parameter: Send to HUB Mode Version 190
- REVTIMER Parameter: Send Interval in Seconds 190
- RIAFTERUPDATE Parameter: Controlling RI Command Behavior 191
- RPLCONNECTCOUNT Parameter: Replication Connection Attempt Count 192
- RPLCONNECTINTERVAL Parameter: Replication Connection Attempt Interval 192
- RPLPARMS Parameter: Replication Parameter Location 192
- RPLSORT Parameter: Event Replicator for Adabas Transaction Data Sorting Control 193
- RPWARNINCREMENT Parameter: Replication Pool Warning Message Interval 194

- RPWARNINTERVAL Parameter: Replication Pool Usage Interval 195
- RPWARNMESSAGELIMIT Parameter: Replication Pool Usage Warning Message Count 195
- RPWARNPERCENT Parameter: Replication Pool Usage Threshold 196
- RVCLIENT Parameter: Adabas Review Client Reporting Activation 196
- SECUID Parameter: Security System User ID Requirement Level 197
- SMF Parameter: Enable Adabas SMF Recording 198
- SMF89 Parameter: Enable SMF Type 89 Records 198
- SMFDETAIL Parameter: Selecting Detail SMF Record Sections 199
- SMFINTERVAL Parameter: Enabling Adabas SMF Interval Records 201
- SMFRECNO Parameter: Selecting User-Defined Adabas SMF Record Number 202
- SMFSUBSYS Parameter: Specifying SMF Subsystem Name 202
- SMGT Parameter: Error Handling and Message Buffering Facility 203
- SORTCACHE Parameter: Sort Area Controller Caching 203
- SPT Parameter: Activate Adabas Triggers and Stored Procedures 204
- SRLOG Parameter: Controlling Spanned Record Logging 205
- SVC Parameter: SVC Number 205
- TARGETID Parameter: Entire Net-Work Target ID 206
- TCPIP Parameter: TCP/IP Access Control 206
- TCPURL Parameter: TCP/IP Universal Resource Locator 207
- TEMPCACHE Parameter: Temp Area Controller Caching 208
- TFLUSH Parameter: Synchronous Buffer Flush Duration 209
- TLSCMD Parameter: Time Limit for S1, S2, and S4 Complex Searches 209
- TMDRQ Parameter: Size of ATM Request Queue 210
- TMETDATA Parameter: ET Data Control 211
- TMGTT Parameter: Global Transaction Time Limit 212
- TMLOG Parameter: ATM Manager Logging Control 213
- TMMSGSEV Parameter: Severity Threshold for Warning Messages 214
- TMRESTART Parameter: ATM Restart Control 214
- TMSYNCMGR Parameter: Support for External Transaction Coordinators 215
- TMTCIDPREF Parameter: Dynamic Client ID Prefix 215
- TNAA Parameter: Nonactivity Time Limit for Access-Only Users 216
- TNAE Parameter: Nonactivity Time Limit for ET Logic Users 217
- TNAX Parameter: Nonactivity Time Limit for Exclusive Control Users 217
- TT Parameter: Transaction Time Limit 218
- UEXn Parameter: User Exit 219
- UEXSMF Parameter: Specifying SMF User Exit Module Name 220
- UPDATECONTROL Parameter: Control Scheduling of Update Commands During Buffer Flush 220
- UTIONLY Parameter: Utilities-Only Session Control 221
- V64BIT Parameter: 64-Bit Virtual Storage Option 222
- VISTA Parameter: Adabas Vista Control 223
- WORKCACHE Parameter: Work Area Controller Caching 223
- ZIIP Parameter: Activate Usage of Adabas for zIIP 224

This chapter describes all of the ADARUN parameters available for Adabas.

Parameter	Use to	Values	Default
ALOGDEV	Specify the device type used for multiple audit log data sets. **Note:** This parameter must be specified with the ALOGSIZE parameter.	Valid device types are listed in *Adabas Device Types and Block Sizes.*	value of the DEVICE parameter
ALOGSIZE	Specify the number of blocks available for each audit log in multiple audit log data sets. **Note:** This parameter must be specified with the ALOGDEV parameter.	16 - 16777215	---
AOslog	Log to DDPRINT the commands issued by ADADBS OPERCOM or equivalent AOS functions that modify the active nucleus.	YES | NO	NO
ARExclude	Exclude file(s) from autorestart.	1 - 5000	---
ARMname	Specify the automatic restart management (ARM) program name.	*arm-pgm-name*	---
ARNworkbuffers	Specify the number of WORK I/O buffers allocated during autorestart processing after a failure.	0 - 2147483647	varies based on ADARUN V64BIT parameter setting.
ASSOCache	Activate or deactivate controller caching for the Associator component.	YES | NO	YES
ASSOSpacewarn	Identify up to three pairs of threshold percentages and percentage change increments, expressed as percentages of the total ASSO portion of the database. When ASSO space use reaches a specified threshold percentage, a space usage message is issued. In addition, when the ASSO space use changes by the increment specified for that threshold, additional space usage messages are issued.	0-100,1-99	0,5
ASYtvs	Flush buffers asynchronously based on volume serial number.	YES | NO	YES
AUDITING	Enable auditing of Adabas nuclei.	YES | NO	NO
AUDITLOG	Specify whether auditing data of utility and AOS functions should be activated. **Note:** The AUDITLOG parameter is independent and unrelated to the AUDITING ADARUN parameter.	CONSOLE | NO	NO
AUTOCQEnv	Identify the type of user affected by the AUTOCQTIME parameter setting. Only commands from users in the specified environment will be	ALL | CICS	ALL

Parameter	Use to	Values	Default
	returned from the command queue if they meet the AUTOCQTIME criteria.		
AUTOCQTime	Specify the timeout period, in seconds, after which an eligible unprocessed command in the command queue during online recovery will be returned to the user with response code 22 (ADARSP022), subcode 55.	0 (zero) | 1 - 86400	0
AUTOINCASSOSize	Specifies the size, in cylinders or blocks, by which the highest-numbered (last-defined) Associator (ASSO) data set will be increased by an automatic database increase process.	0 - 2147483647 (cylinders) 0B - 2147483647B (blocks)	0
AUTOINCASSOTHreshold	Specifies the Associator space threshold at which Adabas automatically initiates a database increase process. This threshold is expressed as a percentage of used Associator space to total Associator space available in the database.	0, 50 - 100	0
AUTOINCASSOTOtal	Specifies the maximum total size, in cylinders or blocks, of the entire Associator space of the database that is not to be exceeded by automatic database increase processes.	0 - 2147483647 (cylinders) 0B - 2147483647B (blocks)	0
AUTOINCDATASize	Specifies the size, in cylinders or blocks, by which the highest-numbered (last-defined) Data Storage (DATA) data set will be increased by an automatic database increase process.	0 - 2147483647 (cylinders) 0B - 2147483647B (blocks)	0
AUTOINCDATATHreshold	Specifies the Data Storage space threshold at which Adabas automatically initiates a database increase process. This threshold is expressed as a percentage of used Data Storage space to total Data Storage space available in the database.	0, 50 - 100	0
AUTOINCDATATOtal	Specifies the maximum total size, in cylinders or blocks, of the entire Data Storage space of the database that is not to be exceeded by automatic database increase processes.	0 - 2147483647 (cylinders) 0B- 2147483647B (blocks)	0
CACHe	Load ADACSH (Adabas Caching Facility control) during Adabas session initialization.	YES | NO	NO
CACTivate	Control RABN activation. (Adabas Caching Facility)	YES | NO	NO
CASSODsp	Cache an Associator RABN or RABN range in the data space cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---

Parameter	Use to	Values	Default
CASSOExt	Cache an Associator RABN or RABN range in the extended memory cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
CASSOG64	Cache an Associator RABN or RABN range backed by 2G large pages in the virtual 64-bit storage cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters.	*rabn* | *rabn1* - *rabnx*	---
CASSOHsp	Cache an Associator RABN or RABN range in the hiperspace cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
CASSOL64	Cache an Associator RABN or RABN range backed by 1M large pages in the virtual 64-bit storage cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters.	*rabn* | *rabn1* - *rabnx*	---
CASSOP64	Cache an Associator RABN or RABN range backed by pageable 1M large pages in the virtual 64-bit storage cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters.	*rabn* | *rabn1* - *rabnx*	---
CASSOV64	Cache an Associator RABN or RABN range in the virtual 64-bit storage cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
CASSOMaxs	Specify the unit size in which areas are allocated for Associator cache space. (Adabas Caching Facility)	80K - 2047M (extended memory, data spaces, or hiperspaces) 1M - your installation limit (virtual 64 memory)	1M (extended memory, data spaces, or hiperspaces) 1M (virtual 64 memory)
CBufno	Specify the number of read-ahead buffers for concurrent I/O processing. (Adabas Caching Facility)	0 - the NT parameter value	0
CCtimeout	Specify the cache space area inactivity time limit (in seconds) before it is released back to the system. If demand caching is in effect, use this parameter to specify the inactivity time limit (in seconds) of a RABN range, file or file range, before it is disabled. (Adabas Caching Facility)	60 - 2147483647	7200 seconds (2 hours)
CDATADsp	Cache a Data Storage RABN or RABN range in the data space cache. This parameter is mutually	*rabn* | *rabn1* - *rabnx*	---

Parameter	Use to	Values	Default
	exclusive with the `CFILE` and `CXFILE` parameters. (Adabas Caching Facility)		
`CDATAExt`	Cache a Data Storage RABN or RABN range in the extended memory cache. This parameter is mutually exclusive with the `CFILE` and `CXFILE` parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
`CDATAG64`	Cache a Data Storage RABN or RABN range backed by 2G large pages in the virtual 64-bit storage cache. This parameter is mutually exclusive with the `CFILE` and `CXFILE` parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
`CDATAHsp`	Cache a Data Storage RABN or RABN range in the hiperspace cache. This parameter is mutually exclusive with the `CFILE` and `CXFILE` parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
`CDATAL64`	Cache a Data Storage RABN or RABN range backed by 1M large pages in the virtual 64-bit storage cache. This parameter is mutually exclusive with the `CFILE` and `CXFILE` parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
`CDATAP64`	Cache a Data Storage RABN or RABN range backed by pageable 1M large pages in the virtual 64-bit storage cache. This parameter is mutually exclusive with the `CFILE` and `CXFILE` parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
`CDATAV64`	Cache a Data Storage RABN or RABN range in the virtual 64-bit storage cache. This parameter is mutually exclusive with the `CFILE` and `CXFILE` parameters. (Adabas Caching Facility)	*rabn* | *rabn1* - *rabnx*	---
`CDATAMaxs`	Specify the unit size in which areas are allocated for Data Storage cache space. (Adabas Caching Facility)	80K - 2047M (extended memory, data spaces, or hiperspaces) 1M - your installation limit (virtual 64 memory)	1M (extended m data spaces, or hiperspaces) 1M (virtual 64 memory)

Parameter	Use to	Values	Default
`CDEmand`	Set the lowest Adabas buffer efficiency level accepted before caching is activated. A value of "0" indicates that no demand-level caching occurs. (Adabas Caching Facility)	0 - 2147483647	0
`CDIspstat`	Control whether RABN range statistics display on both the operator console and DDPRINT, or only on DDPRINT. (Adabas Caching Facility)	YES (display on both) | NO (display only on DDPRINT)	YES
`CDX`*`nn`*	Specify the user routine of collation descriptor user exits. The *`nn`* in the parameter name can range from 01 through 08, so up to eight collation descriptor exits may be specified (in any order).	user routine name (up to 8 characters)	---
`CExclude`	Identify command types that should be excluded from read-ahead caching. More than one command type may be specified by separating values with forward slashes (/). (Adabas Caching Facility)	PHYS | LOGI | HIST | FIND	—
`CFile`	Select a file or range of files to be cached. This parameter is also used to indicate the class of caching service, the caching scope, and the cache storage type that should be implemented for the file or file range. Specify this parameter using the following syntax: `CFILE=({`*`fnr`* | *`fnr1`*`-`*`fnrx`*`}[,`*`class`*`][,`*`scope`*`][,`*`cachetype`*`])` where: ■ *`fnr`*, *`fnr1`*, *`fnrx`*: valid file numbers. When a range is specified, the lower file number must be specified first (*`fnr1`*). ■ *`class`*: The percentage of cache space that can be used, which effectively assigns a priority to the files, with lower priority RABNs being purged first. A value of 1 indicates the highest priority, 5 indicates the lowest priority. ■ *`scope`*: Identifies the type of caching used: Associator (A), Data Storage (D), or both (B). ■ *`cachetype`*: Identifies the cache storage type used: data space (D), extended (E), 2G large pages in virtual 64-bit storage (G), hiperspace (H), 1M fixed large pages in virtual 64-bit storage (L), 1M	*`fnr`*, *`fnr1`*, *`fnrx`*: 0 - n *`class`*: 1 | 2 | 3 | 4 | 5 *`scope`*: A | D | B *`cachetype`*: D | E | G | H | L | P | V	*`fnr`*, *`fnr1`*, *`fnrx`*: --- *`class`*: 3 *`scope`*: B *`cachetype`*: D

Parameter	Use to	Values	Default
	pageable large pages in virtual 64-bit storage (P), or virtual 64-bit storage (V). This parameter is mutually exclusive with the `CASSO`*xxx*, `CDATA`*xxx* and `CSTORAGE` parameters. (Adabas Caching Facility)		
`CLOGBmax`	Specify the maximum size (bytes) of a logged buffer.	368 - 2147483647	4096
`CLOGDev`	Specify the device type used for multiple command log data sets. This parameter must be specified with the `CLOGSIZE` parameter.	Valid device types are listed in *Adabas Device Types and Block Sizes*.	value of the `DEV` parameter
`CLOGLayout`	Specify the format of the Adabas command log.	8	8
`CLOGMAx`	Specify the maximum size (bytes) of all of the logged buffers allowed for an Adabas command.	368 - 2147483647	16384
`CLOGMRg`	Indicate whether the Adabas cluster should run with an automatic CLOG merge. (Adabas Cluster Services, Adabas Parallel Services)	YES | NO	NO
`CLOGSize`	Specify the number of blocks available for each command log in multiple command log data sets. This parameter must be specified with the `CLOGDEV` parameter.	16 - 16777215	---
`CLUCACHEExtra`	Specify the additional vector size, in blocks, required to allow Adabas Cluster Services and Adabas Parallel Services to track the blocks held by Adabas Caching Facility in the different cluster nuclei. (Adabas Cluster Services, Adabas Parallel Services)	0 - 2147483647	1000
`CLUCACHEName`	Identifies the name of the Adabas cluster cache structure/area in which the current nucleus participates. (Adabas Cluster Services)	*`cluster-name`*	---
`CLUCACHESize`	Specify the amount of storage to allocate for the global cache area that services the Adabas Parallel Services cluster in which the current nucleus participates. (Adabas Parallel Services)	128K - varies (depending on the cache type specified by the CLUCACHETYPE parameter)	---
`CLUCACHEType`	Identify the virtual storage type for the global cache area: shared dataspace (DSP), virtual 64-bit storage (V64), virtual 64-bit cache backed by page-fixed two-gigabyte (2G) large pages (G64), virtual 64-bit cache backed by page-fixed one-megabyte (1M)	DSP | G64 | L64 | P64 | V64	DSP

Parameter	Use to	Values	Default
	large pages (L64), or virtual 64-bit cache backed by pageable one-megabyte (1M) large pages (P64). (Adabas Parallel Services)		
`CLUCACHEUnchanged`	Indicate whether unchanged blocks are written to global cache during Adabas Parallel Services processing. (Adabas Parallel Services)	YES | NO	NO
`CLUGroupname`	Specify the XCF messaging group name in Adabas Cluster Services or the name of the messaging group in Adabas Parallel Services. (Adabas Cluster Services, Adabas Parallel Services)	*`groupname`*	none in Adabas Cluster Services; the router name in Adabas Parallel Services
`CLULOCKName`	Specify the lock structure name defined in the CFRM policy or the global lock area in which the current nucleus participates. (Adabas Cluster Services)	*`lockstructurename`*	---
`CLULOCKSize`	Specify the amount of storage allocated for the global lock area. Specify values for this parameter using the following syntax: *`nnn`*`[K | M | G]` where *`nnn`*: the number of bytes (no unit specification), kilobyte (K), megabytes (M), or gigabytes (G). (Adabas Parallel Services)	131072 - 2147483147 (bytes) 128K - 2097152K 128K - 2048M 128K - 2G	---
`CLUPublprot`	Specify whether or not cluster nuclei make their protection data available within the cluster before they make the related updates available to the other nuclei. (Adabas Cluster Services, Adabas Parallel Services)	YES | NO	NO
`CLUSter`	Indicate whether the Adabas nucleus will participate in a cluster of nuclei working on the same database, and, if so, the type of cluster in which it will participate. Valid values are "NO" (nucleus not participating in a cluster), "LOCAL" (nucleus participating in a cluster on the same operating system image), and "SYSPLEX" (nucleus participating in a cluster on different systems in a parallel sysplex). (Adabas Cluster Services, Adabas Parallel Services)	NO | LOCAL | SYSPLEX	NO

Parameter	Use to	Values	Default
`CLUWork1cache`	Specify the maximum number of WORK blocks that the cluster nucleus should keep in the global cache at a time.	0 - 32767	0
`CMAXcsps`	Specify the number of storage areas that can be allocated for ADACSH cache space in data spaces, hiperspaces, or virtual 64-bit storage. This parameter has no affect on a cache space in extended memory, where only one Associator and one Data Storage area are allowed. (Adabas Caching Facility)	1 - 16	8
`CRetry`	Specify the number of seconds between Adabas Caching Facility attempts to acquire an area of cache space if the previous GETMAIN or space allocation failed. (Adabas Caching Facility)	60 - 2147483647	900 (15 minutes)
`CStorage`	Identify the type of cache space to which all RABNs for the database are cached (using system defaults). This parameter overrides the use of the CASSO*xxx* and CDATA*xxx* ADARUN parameters. Valid values are "DATASPACE" (dataspace), "EXTENDED" (extended memory), "HIPERSPACE" (hiperspace), "G64" (2G large pages in virtual 64-bit storage), "L64" (fixed 1M large pages in virtual 64-bit storage), "P64" (pageable 1M large pages in virtual 64-bit storage), or "VIRTUAL64" (virtual 64-bit storage). This parameter is mutually exclusive with the `CFILE` and `CXFILE` parameters. (Adabas Caching Facility)	DATASPACE | EXTENDED | G64 | HIPERSPACE | L64 | P64 | VIRTUAL64	---
`CT`	Specify the maximum time limit (in seconds) for interregion communication of results from Adabas to the user.	1 - 2147483647	60
`CWORKStorage`	Identify the type of cache space used for Adabas Caching Facility work areas CWORK2FAC and CWORK3FAC. Valid values are "DATASPACE" (dataspace), "EXTENDED" (extended memory), "G64" (2G large pages in virtual 64-bit storage), "HIPERSPACE" (hiperspace), "L64" (fixed 1M large pages in virtual 64-bit storage), "P64" (pageable 1M large pages in virtual 64-bit storage), or "VIRTUAL64" (virtual 64-bit storage). (Adabas Caching Facility)	DATASPACE | EXTENDED | G64 | HIPERSPACE | L64 | P64 | VIRTUAL64	---

Parameter	Use to	Values	Default
CWORK2fac	Specify the percentage of Work part 2 to be cached. If "0" is specified, Work part 2 is not cached. (Adabas Caching Facility)	0 - 100	0
CWORK3fac	Specify the percentage of Work part 3 to be cached. If "0" is specified, Work part 3 is not cached. (Adabas Caching Facility)	0 - 100	0
CXfile	Select a file or range of files to be excluded from caching. Specify this parameter using the following syntax: CXFILE=({*fnr* | *fnr1*-*fnrx*}[,,*scope*]) where: ■ *fnr*, *fnr1*, *fnrx*: valid file numbers to be excluded. When a range is specified, the lower file number must be specified first (*fnr1*). ■ *scope*: Identifies the scope of exclusion: Associator RABNs only excluded (A), Data Storage RABNs only excluded (D), or both types of RABNs excluded (B). If a scope is specified, you must specify two commas before it. This parameter is mutually exclusive with the CASSO*xxx*, CDATA*xxx*, and CSTORAGE ADARUN parameters. (Adabas Caching Facility)	*fnr*, *fnr1*, *fnrx*: 0 - n *scope*: A | D | B	*fnr*, *fnr1*, *fnrx*: --- *scope*: B
DATACache	Control whether caching control is enabled (YES) or disabled (NO) for the Data Storage data set.	YES | NO	YES
DATASpacewarn	Identify up to three pairs of threshold percentages and percentage change increments, expressed as percentages of the total Data Storage portion of the database. When Data Storage space use reaches a specified threshold percentage, a space usage message is issued. In addition, when the Data Storage space use changes by the increment specified for that threshold, additional space usage messages are issued.	0-100,1-99	0,5
DBid	Specify the physical database ID of the database. **Note:** If you specify the DBID parameter, you must also specify the ADARUN SVC parameter.	1 - 65535	1
DEVIce	Specify the device type on which the first block of the Associator is stored. Valid device types for each	*devtype*	3390

Parameter	Use to	Values	Default
	platform are listed in *Adabas Device Types and Blocks Sizes*.		
`DIrratio`	Together with the `ELEMENTRATIO` ADARUN parameter, define the ratio of directory entries to data elements in Adabas cluster environments. (Adabas Cluster Services, Adabas Parallel Services)	1 - 32767	1
`DSF`	Enable Adabas Delta Save support. (Adabas Delta Save)	YES | NO	NO
`DSFEx1`	Specify the user routine to be given control by Adabas Delta Save. (Adabas Delta Save)	*`exitname`*	---
`DTp`	Indicate whether to enable distributed transaction processing, and, if so, whether the nucleus should be a resource or transaction manager. Valid values are NO (distributed transaction processing is not enabled), RM (the nucleus participates in distributed transaction processing as a resource manager), and TM (the nucleus participates in distributed transaction processing as a transaction manager). (Adabas Transaction Manager)	NO | RM | TM	NO
`DUALALD`	Specify the device type to be used for dual audit logging.	*`devtype`*	value of the DEV parameter
`DUALALS`	Specify the number of blocks available for each dual audit log.	16 - 16777215	---
`DUALCLD`	Specify the device type to be used for dual command logging. Valid device types for each platform are listed in *Adabas Device Types and Blocks Sizes*.	*`devtype`*	value of the `DEV` parameter
`DUALCLS`	Specify the number of blocks available for each dual command log.	16 - 16777215	---
`DUALPLD`	Specify the device type to be used for dual protection logging. Valid device types for each platform are listed in *Adabas Device Types and Blocks Sizes*.	*`devtype`*	value of the `DEV` parameter
`DUALPLS`	Specify the number of blocks available for each dual protection log.	16 - 16777215	---
`ELementratio`	Together with `DIRRATIO` ADARUN parameter, define the ratio of directory entries to data elements in Adabas cluster environments. Adabas Cluster Services, Adabas Parallel Services	1 - 32767	1

Parameter	Use to	Values	Default
`ENCRYPTION`	Indicate whether or not to activate Adabas Encryption	YES | NO	NO
`EXcpvr`	Indicate whether EXCP or EXCPVR should be used when running APF-authorized.	YES | NO	YES
`FAstpath`	Enable Adabas Fastpath support. (Adabas Fastpath)	YES | NO	NO
`FMxio`	Set a limit on the number of I/O operations that can be started in parallel by LFIOP flush processing. The full meaning of this parameter varies, depending on the setting of the `ASYTVS` ADARUN parameter. If ASYTVS=YES, you can use FMXIO to set the limit on the number of I/O ■ If ASYTVS=YES, you can use FMXIO to set a limit on the number of I/O operations to be started in parallel on each volume. ■ If ASYTVS=NO, you can use FMXIO to set a limit on the number of I/O operations to be started in parallel overall.	1 - 16 (ASYTVS=YES) 1 - 100 (ASYTVS=NO)	1 (ASYTVS=YES) 60 (ASYTVS=NO)
`FOrce`	Indicate whether the nucleus or Adabas Review hub can overwrite an existing ID table entry.	YES | NO	NO
`HEX`*`nn`*	Specify the hyperdescriptor exit(s) and associated user routine to be used by the nucleus. Specify this parameter using the following syntax: `HEX`*`nn`*`=`*`user-routine-name`* where: ■ *`nn`*: the number of the hyperdescriptor exit ■ *`user-routine-name`*: the name (up to eight characters) of the user routine that gets control for the associated hyperdescriptor exit.	■ *`nn`*: 01 - 31 ■ *`user-routine-name`*: text (up to eight characters)	■ *`nn`*: --- ■ *`user-routine-name`*: ---
`IGNDIb`	Indicate whether an active nucleus entry in the data integrity block (DIB) should be ignored and deleted.	YES | NO	NO
`IGNDTp`	Indicate whether the two-phase commit area (Work part 4) should be ignored.	YES | NO	NO
`INDEXCrosscheck`	Indicate whether index cross-checking is turned on or off. Index cross-checking is a process Adabas uses to ensure that a data storage record it is reading actually matches the descriptor value in the index by which the record was found.	YES | NO	YES

Parameter	Use to	Values	Default
INDEXUpdate	Indicate whether to use the original or advanced index update processing rules.	ORIGINAL | ADVANCED	ORIGINAL
INFobuffersize	Specify the size of the information buffer pool, in bytes. The information buffer pool is also known as the Adabas event log.	0 - 2147483647	0 (buffer will no allocated)
INTAuto	Specify the time interval (in seconds) between autorestart progress messages, which will be printed in addition to the standard messages at the end of each autorestart phase, if the phase takes longer than the interval specified. The default (0 seconds) indicates that autorestart progress messages should only be printed at the completion of each autorestart phase.	0 - 2147483647	0
INTNas	Specify the time interval (in units of 1.048576 seconds) between SYNS 60 checkpoints.	1 - 2147483647	3600 (approxima one hour)
LAP	Specify the size of the Auditing Pool.	minimum 20000	100000 (if AUDITING=YES) AUDITING=NO)
LARgepage	Indicate whether the Adabas nucleus should use page-fixed one-megabyte (1M) large pages above the two-gigabyte (2G) bar (L64), page-fixed 2G large pages above the 2G bar (G64), or pageable one-megabyte (1M) large pages (P64). A value of "YES" is equivalent to specifying "L64". A value of "NO" indicates that virtual storage above the 2G bar will be backed by four-kilobyte (4K) pages. Values of "G64", "L64", and "YES" can only be specified if the ADARUN V64BIT parameter is also set to "YES". In addition, the operating system must support large pages of the required type.	G64 | L64 | P64 | NO | YES	NO
LBP	Specify the maximum number of bytes to be used for the Adabas buffer pool during a session.	80000 - 18446744073709551615 (although this is really limited by the available amount of virtual and real memory)	350000
LCP	Specify the maximum number of bytes to be used for the Adabas security pool during a session.	2000 - 2147483647	10000
LDEuqp	Specify the number of bytes to be used for the unique (UQ) descriptor pool during a session.	5000 - 2147483647	5000

Parameter	Use to	Values	Default
LDTp	Define the length of the distributed transaction processing area (Work part 4) or the index of Work part 4: If you have Adabas Transaction Manager installed, this parameter defines the size (in blocks) of the Work part 4 index which is used to administer the data on Work part 4 (DDWORKR4). (Adabas Transaction Manager)	8 - one eighth of the WORK4 size	8
LFIop	Enable asynchronous buffer flush operations and define the length of the related buffer flush pool. A value of "0" disables asynchronous buffer flush operations; only synchronous flushing occurs. A value of "1" enables asynchronous buffer flush operations, and defines the buffer pool size as 25% of the buffer pool (LBP parameter) size. Any other value (with a minimum of 80000 allowed) enables asynchronous buffer flush operations and defines the size of the pool directly (in bytes).	0 | 1 | 80000 - 18446744073709551615 (although this is really limited by the available amount of virtual and real memory)	0
LFP	Specify the size (in bytes) of the Adabas internal format buffer pool.	6000 - 2147483647	12000
LI	Specify the size (in bytes) allocated for the table of ISNs (TBI) used to store ISN lists (overflow ISNs or saved ISN lists).	2000 - 2147483647	10000
LNKGNAME	The name of the link globals table to be employed by an Adabas 8 batch/TSO link routine. **Note:** If you specify the LNKGNAME parameter, you must also specify the ADARUN DBID and SVC parameters.	valid module name	LNKGBLS
LOCal	Indicate whether an Adabas nucleus or Adabas Review hub is isolated and available for local use only. The isolated nucleus or hub will be unknown to the network.	YES | NO	YES
LOGAbdx	Indicate whether Adabas buffer descriptions (ABDs) should be logged during this session. For the dependencies of the various logging parameters, please review the detailed description.	YES | NO	NO
LOGCB	Indicate whether Adabas extended control blocks (ACBX) should be logged during this Adabas session. For the dependencies of the various logging parameters, please review the detailed description.	YES | NO	NO
LOGCLex	Indicate whether the Adabas command log extension (CLEX) should be logged. For the	YES | NO	NO

Parameter	Use to	Values	Default
	dependencies of the various logging parameters, please review the detailed description.		
LOGFb	Indicate whether Adabas format buffers (FB) should be logged.	YES | NO	NO
LOGGing	Indicate whether command logging should be used for the Adabas session. If this is set to "NO", all other LOG*xxxx* parameters are ignored.	YES | NO	NO
LOGIB	Indicate whether Adabas ISN buffers (IB) should be logged.	YES | NO	NO
LOGIO	Indicate whether Adabas I/O activity should be logged.	YES | NO	NO
LOGMb	Indicate whether Adabas multifetch buffers (MB) should be logged. For the dependencies of the various logging parameters, please review the detailed description.	YES | NO	NO
LOGRb	Indicate whether Adabas record buffers (RB) should be logged.	YES | NO	NO
LOGSB	Indicate whether Adabas search buffers (SB) should be logged.	YES | NO	NO
LOGSIze	Specify the maximum block size for the sequential command log.	100 - 32760	value of the QBL ADARUN parar
LOGUX	Indicate whether Adabas user exit B data should be logged. For the dependencies of the various logging parameters, please review the detailed description.	YES | NO	NO
LOGVB	Indicate whether Adabas value buffers (VB) should be logged.	YES | NO	NO
LOGVOlio	Indicate whether the extended I/O list should be written to the command log for CLOGLAYOUT=8.	YES | NO	NO
LOGWarn	Specify the frequency, in seconds, at which the PLOG and CLOG status is checked and resulting alert messages are produced. This parameter is optional, but a valid user exit 2 or user exit 12 must also be specified when this parameter is set to any value other than zero (0). A value of zero (0), the default, indicates that no CLOG and PLOG status checks should occur and that no alert messages should be produced. Any other valid LOGWARN value represents the interval (in seconds) at which the PLOG and CLOG status is checked and resulting alert messages are produced.	0 - 2147483647	0

Parameter	Use to	Values	Default
LP	Specify the number of blocks to allocate to the data protection area (Work part 1). The highest value you can set for this parameter is limited by the size of the Work data set and the sizes of Work parts 2, 3, and 4.	200 - *	1000
LQ	Specify the size (in bytes) of the table of sequential commands, which contains entries required during the processing of Adabas read sequential (L2/L5, L3/L6, and L9) commands.	2000 - 2147483647	10000
LRDp	Specify the size (in bytes) of the redo pool, which allows for deferred publishing. If the value of this parameter is set to 0, updated database blocks are always written to global cache at the time the update is made; no deferred publishing occurs. (Adabas Cluster Services, Adabas Parallel Services)	0 | 80000 - 2137483647	For Adabas Parallel Services 8.2 or later, the default is 0 (zero). For Adabas Cluster Services 8.2 or later, the default is either the value of the ADARUN LFIOP parameter or 10 megabytes, whichever is smaller. For earlier releases of Adabas Parallel Services and Adabas Cluster Services, the default is the value of the ADARUN LFIOP parameter.
LRPl	Specify the size of the Adabas or Event Replicator replication pools, in bytes. This parameter should be specified for only: ■ an Adabas nucleus that also has REPLICATION=YES set. ■ an Event Replicator database The values for the Adabas nucleus and the Event Replicator database may be different.	20000 - *	100000 (if REPLICATION=YES) 0 (if REPLICATION=NO)
LS	Specify the maximum number of bytes for internal sort processing. **Note:** To ensure that a single search command does not use too much of the work pool space, the maximum LS value must be less than or equal to the LWP (work pool length) divided by two minus the minimum LS value (19968) (LS = LWP/2 - 19968).	19968 - *	49920

Parameter	Use to	Values	Default
LSF	Indicate whether Logical Save Facility should be enabled (YES) or disabled (NO) for the nucleus.	YES | NO	NO
LTZ	Specify the size of the time zone pool used by a user session, in bytes.	0, 12288 - 2147483647	32768
LU	Specify the length (in bytes) of the intermediate user buffer area. On z/OS systems, LU cannot exceed a value greater than that produced by this calculation: (*NABvalue* x 4096)	a positive integer, minimum and maximum governed by the operating system.	65535
LWKP2	Specify the number of Work blocks to use for intermediate ISN lists Work part 2).	0 - 16777215	0
LWP	Specify the size (in bytes) of the Adabas work pool. This value must be equal to or greater than twice the sum of the specified internal sort area length (LS parameter) plus the minimum LS value (19968). In other words, the LWP value must be greater than or equal to a value calculated as follows: (*LS-value* + 19968)x 2 The minimum value you can specify is 80000 or a value not less than the value of the NT parameter multiplied by 25000.	80000-2147483647	150000
MLwto	Specify whether multi-line messages that are normally written to the operator console as a series of single-line write-to-operator (WTO) messages should instead appear as multi-line WTO messages. (Event Replicator for Adabas only)	YES | NO	NO
MODe	Identify the Adabas operating mode. Valid operating modes include running in single-user mode (SINGLE) or running in multiuser mode (MULTI).	SINGLE | MULTI	MULTI
MSGBuf	Specify the size (in kilobytes) of the message buffer area, which is used to buffer messages for use for performance and tuning and problem analysis.	0 - 128	0
MSGConsl	Specify the case of Adabas messages sent to the console.	Upper | Mixed	Mixed
MSGDruck	Specify the case of Adabas messages sent to the DDDRUCK data set.	Upper | Mixed	Mixed
MSGPrint	Specify the case of Adabas messages sent to the DDPRINT data set.	Upper | Mixed	Mixed

Parameter	Use to	Values	Default
MXCANCEL	Specify the timeout threshold, in seconds, in which processing of an intracluster cancellation request must occur. Adabas Cluster Services, Adabas Parallel Services	15 - 2147483647	The value of the MXMSG parameter.
MXCANCELWarn	Specify the timeout threshold, in seconds, for an intracluster cancellation request to occur before issuing an early warning about the failure of the cancellation request. The value of this parameter must be less than the value of the MXCANCEL parameter. A value of "0" indicates that no warnings should be issued. Adabas Cluster Services, Adabas Parallel Services	0 | 4 - (*MXCANCEL_value* - 1)	The value of the MXCANCEL parameter divided by 4.
MXMSG	Specify the timeout threshold, in seconds, in which the processing of an intracluster command must occur. Adabas Cluster Services, Adabas Parallel Services	15 - 32767	300
MXMSGWarn	Specify the timeout threshold, in seconds, for a response to an intracluster command to occur before issuing an early warning about the failure of the request. The value of this parameter must be less than the value of the MXMSG parameter. A value of "0" indicates that no warnings should be issued. Adabas Cluster Services, Adabas Parallel Services	0 | 4 - (*MXMSG_value* - 1)	The value of the MXMSG parameter divided by 4.
MXStatus	Specify the interval, in seconds, at which the member-level XCF status monitoring heartbeat should occur. A value of "0" indicates that no member-level status monitoring should occur. Adabas Cluster Services	0 | 15 - 21474836	15
MXTNa	Set the maximum non-activity time, in seconds, that can be specified in the OP command for an individual user.	1 - 65535	3600
MXTSx	Set the maximum execution time (in seconds) that can be specified for the ADARUN TLSCMD parameter and for any override execution time setting that might be specified in the Adabas control block of an OP command.	1 - 65535	3600

Parameter	Use to	Values	Default
MXTT	Set the maximum transaction time limit, in seconds, that can be specified in the OP command for an individual user.	1 - 65535	3600
MXWtor	Specify the timeout interval, in seconds, in which an operator response to a self-termination operator query must occur before issuing a warning. A value of "0" indicates that no warning message should be issued. Adabas Cluster Services	0 | 15 - 64800	0
NAb	Specify the number of attached buffers to be used during the session. The maximum value you can specify for this parameter varies, depending on the amount of available virtual storage.	1 -varies	16
NALOG	Specify the maximum number of audit log data sets to be opened and logged in the PPT for the session.	0 | 2 - 8	0
NC	Specify the maximum number of command queue elements (CQEs) for the session. The number of CQEs established for the session determines the maximum number of Adabas commands that may be queued or be in process at any one time during the session.	20 - 32767	200
NCLog	Specify the maximum number of command log data sets to be opened and logged in the PPT for the session. A value of "0" disables this parameter.	0 | 2 - 8	0
NH	Specify the maximum number of hold queue elements for the session.	20 - 16777215	500
NIsnhq	Specify the maximum number of records that can be placed in hold status (hold queues) at the same time by a single user. The maximum value that can be specified for this parameter is the value set for the NH ADARUN parameter (which has a maximum value of 16777215). The default is the larger of 20 or the value of the NH parameter divided by the value of the NU parameter with the result multiplied by two ((NH/NU) * 2).	1 - 16777215 (NH parameter maximum)	20 | (NH/NU) * (whichever is la
NOndes	Indicate whether to allow (YES) searches using nondescriptors in search buffers.	YES | NO	YES
NPLOG	Specify the maximum number of protection log data sets to be opened and logged in the PPT. A value of "0" disables this parameter.	0 | 2 - 8	0
NPLOGBuffers	Specify the number of PLOG protection I/O buffers allocated in the Adabas nucleus session. PLOG protection I/O buffers are only allocated when	1 - 2,147,483,687	1

Parameter	Use to	Values	Default
	ADARUN parameter LFIOP is set to a nonzero value.		
NQcid	Specify the maximum number of active command IDs per user.	1 - 65535	20
NSisn	Specify the maximum number of ISNs per ISN table (TBI) element. The maximum value that can be specified varies based on the device type of the Work data set or, in the case where user-defined Work block sizes are used, the maximum value is based on a formula. For more information, refer to the more detailed NSISN documentation.	7 - *	51
NT	Specify the number of threads to be used during the Adabas session.	4 - 250	5
NU	Specify the maximum number of user queue elements for the Adabas session.	20 - 16777215	200
NUCid	Specify the ID of an Adabas nucleus in an Adabas cluster. Adabas Cluster Services, Adabas Parallel Services	0 - 65000	0
NWork1buffers	Specify the number of Work part 1 protection I/O buffers allocated in the Adabas nucleus session. Work part 1 protection I/O buffers are only allocated when ADARUN parameter LFIOP is set to a nonzero value.	1 - 2,147,483,687	1
OPenrq	Indicate whether an open command (OP) must be issued (is required) as the first command of a user session.	YES | NO	YES
PGfix	Indicate whether pages containing I/O control blocks are released after I/O processing is completed or after the job has ended when using EXCPVR. When this parameter is set to *YES*, pages containing the I/O control blocks are fixed for the duration of the job. When set to "NO", pages are fixed only for the duration of I/O processing.	YES | NO	NO
PLOGDev	Specify the device type used for multiple protection log data sets. This parameter must be specified with the PLOGSIZE parameter.	Valid device types are listed in *Adabas Device Types and Block Sizes*	value of the DEVICE parameter
PLOGRq	Indicate whether a protection log is required for the Adabas session. For details on the meanings of the different values for this parameter, refer to the detailed PLOGRQ documentation.	YES | NO | SEL | FORCE	YES
PLOGSIze	Specify the number of blocks available for each protection log in multiple protection log data sets.	16 - 16777215	---

Parameter	Use to	Values	Default
	This parameter must be specified with the PLOGDEV parameter.		
PREFEtch	Indicate whether the Adabas command-level prefetch or multifetch features should be automatically enabled. YES enables the multifetch (M) option; OLD enables the prefetch (P) option; NO disables the prefetch feature.	YES | OLD | NO	NO
PREFICmd	Identify the commands that should be included in prefetch or multifetch processing. To specify multiple values for this parameter, list the values separated by forward slashes (/). For example: PREFICMD=L3/L9. The maximum number of entries is 50. The same number of list items must be specified for this parameter as are specified for the PREFIFIL parameter. If you want several commands to be prefetched or multifetched for a single file, repeat the same file number in the PREFIFIL parameter. The values for this parameter are used in conjunction with the value of the PREFIFIL parameter to customize prefetch and multifetch processing during the session. You cannot specify this parameter in the same session as the PREFXCMD parameter; if you do, the values of the PREFXCMD parameter are used and the values for this parameter are ignored.	L1 - L9	---
PREFIFil	Identify the files that should be included in prefetch or multifetch processing. To specify multiple values for this parameter, list the values separated by commas. The maximum number of entries is 50. The same number of list items must be specified for this parameter as are specified for the PREFICMD parameter. If you want several commands to be prefetched or multifetched for a single file, repeat the same file number in the PREFIFIL parameter. The values for this parameter are used in conjunction with the value of the PREFICMD parameter to customize prefetch and multifetch processing during the session. You cannot specify this parameter in the same session as the PREFXFIL parameter; if you do, the values of the PREFXFIL parameter are used and the values for this parameter are ignored.	1 - 5000	---

Parameter	Use to	Values	Default
PREFNrec	Specify the number records that can be prefetched or multifetched. The maximum number of records that can be multifetched depends on the maximum that can be held by the record and ISN buffers.	0 - *	0
PREFSBl	Specify the size, in bytes, of the internal buffer used by multifetch/prefetch processes, which determines how much record data is prefetched.	1024 - *<PREFTBL value>*	3000
PREFSTdd	Specify the job statement label (DD name) in the nucleus startup job that defines the location to which multifetch statistics should be written.	*ddname*	---
PREFTbl	Specify the size, in bytes, of the total prefetch buffer area.	6144 - 2147483647	30000
PREFXCmd	Identify the commands that should be *excluded* in prefetch or multifetch processing. To specify multiple values for this parameter, list the values separated by forward slashes (/). For example: PREFXCMD=L3/L9. The maximum number of entries is 50. The same number of list items must be specified for this parameter as are specified for the PREFXFIL parameter. If you want several commands to be prefetched or multifetched for a single file, repeat the same file number in the PREFXFIL parameter. The values for this parameter are used in conjunction with the value of the PREFXFIL parameter to customize prefetch and multifetch processing during the session. You cannot specify this parameter in the same session as the PREFICMD parameter; if you do, the values of this parameter are used and the values for the PREFICMD parameter are ignored.	L1 - L9	---
PREFXFil	Identify the files that should be excluded in prefetch or multifetch processing. To specify multiple values for this parameter, list the values separated by commas. The maximum number of entries is 50. The same number of list items must be specified for this parameter as are specified for the PREFXCMD parameter. If you want several commands to be prefetched or multifetched for a single file, repeat the same file number in the PREFXFIL parameter. The values for this parameter are used in conjunction with the value of the PREFICMD parameter to customize prefetch and multifetch processing during the session. You cannot specify	1 - 5000	---

Parameter	Use to	Values	Default
	this parameter in the same session as the `PREFXFIL` parameter; if you do, the values of the `PREFXFIL` parameter are used and the values for this parameter are ignored.		
`PROGram`	Identify the program to be run.	ADACOM | ADANUC | ADAREV | NETWRK | RENTUSER | USER | *`utility-name`*	USER
`QBlksize`	Specify the block size, in bytes, for sequential data sets used by ADAIOR, the Adabas I/O component. A value of "0" does not indicate that 0-byte block sizes should be used, but that the block size should be determined by the media type. For more information, read the detailed QBLKSIZE documentation.	0 - 262144	0
`READonly`	Indicate whether the Adabas session is a read-only session or it should permit database updates.	YES | NO	NO
`REFstprt`	Indicate whether statistics should be printed after they are refreshed after an ADADBS REFRESHSTATS utility function run or after interval statistics have been refreshed using Adabas Online System (option **R** in **Session Monitoring**).	YES | NO	YES
`REPlication`	Indicate whether replication should be activated for the Adabas nucleus. (Event Replicator for Adabas)	YES | NO	NO
`REVBUFMB`	Specify the size of the Review Buffer Pool used as temporary storage when sending the CLOG records via Network (ADALNK) to the Review Hub. (Adabas Review)	1 - 1024	10
`REVFilter`	Indicate whether Adabas Review record filtering should be allowed during the session. (Adabas Review)	YES | NO	YES
`REVIew`	Indicate whether Adabas Review should be run in local mode, hub mode (specifying the hub ID), or not at all. (Adabas Review)	NO | LOCAl | *`hubid`*	NO
`REVLOGBmax`	Specify the maximum allowable number of bytes of a logged buffer for Adabas Review. (Adabas Review)	0 - 30000	5120

Parameter	Use to	Values	Default
REVLOGMax	Specify the maximum size of all of the logged buffers allowed for an Adabas Review command. (Adabas Review)	2000 - 32764 (32K - 4)	16384
REVLOGNR	Hub mode: Specify whether the CLOG records are sent as buffered records or as single records via Network (ADALNK) to the Review Hub. Local mode: Specify the CLOG record after which the REVIEW subtask is posted for processing.	0 - 64	Hub mode:0 Local mode: 16
REVSTHMV	Specify whether the ADARMT/ADARST interface or the ADARMX/ADARSX interface is used to send data via Network (ADALNK) to the Review Hub.	V52 V53	V52
REVTIMER	Hub mode: Specify the largest delay for sending buffered CLOG records via Network (ADALNK) in case of low database activity. Local mode: Send the CLOG records to the Review hub after no later than *N* seconds.	1 - 60	1
RIafterupdate	Indicate whether Adabas should suppress the response code 113 (ADARSP113) subcode 5 and response code 2 (ADARSP002) subcode 5 that are returned when an application issues an RI command for a record that has been updated in the current transaction. The RI command returns response code 0 instead, but does not release any updated records from hold.	YES | NO	NO
RPLCONNECTCount	Specify the number of connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails.	0 - 2147483647	0
RPLCONNECTInterval	Specify the interval (in seconds) between connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails.	0 - 2147483647	0
RPLParms	Identify the location where your replication definitions (initialization parameters) should be read from. Replication definitions can be read from the Replicator system file (FILE), from DDKARTE (PARMS), both the Replicator system file and from DDKARTE (where the DDKARTE specifications	BOTH | FILE | NONE | PARMS	If the Replicator system file resides on the Event Replicator database, definitions are read first from the Replicator system file

Parameter	Use to	Values	Default
	override any duplicates with the Replicator system file), or not at all (NONE). (Event Replicator for Adabas)		and then from DDKARTE. If th Replicator syste does not reside Event Replicator database, the definitions are r from DDKARTE
`RPLSort`	Indicate whether Event Replicator for Adabas transaction data sorting should occur. (Event Replicator for Adabas)	YES | NO	YES
`RPWARNINCrement`	Specify the interval (in percentage of `LRPL` usage) at which warning messages should be sent indicating that the replication pool usage has exceeded the threshold set by the `RPWARNPERCENT` parameter. (Event Replicator for Adabas)	1 - 99	10
`RPWARNINTerval`	Specify the interval, in seconds, during which replication pool usage warning messages are suppressed on the console. (Event Replicator for Adabas)	1 - 2147483647	60
`RPWARNMessagelimit`	Specify the number of replication pool usage warning messages that can be issued to the console before message suppression begins. (Event Replicator for Adabas)	1 - 2147483647	5
`RPWARNPercent`	Specify the threshold for replication pool usage (expressed as a percentage of LRPL) at which replication pool usage warning messages should be sent. A value of *0* for this parameter indicates that no warning messages are issued. (Event Replicator for Adabas) .	0 - 99	0
`RVclient`	Specify whether Adabas Review client reporting should be activated in batch environments. **Note:** This ADARUN parameter is only valid when ADARUN PROGRAM=USER. (Adabas Review)	ACTIVE | INACTIVE	INACTIVE
`SEcuid`	Specify the requirement level of security system user IDs for a database. Using it you can indicate how Adabas handles calls from users without a	ALLOW | REQUIRE | WARN	ALLOW

Parameter	Use to	Values	Default
	security system user ID or with a security system user ID that changed during the Adabas session.		
SMF	Specify whether or not Adabas SMF recording should be enabled.	YES | NO	NO
SMF89	Specify whether or not the Adabas nucleus should register with z/OS for type 89 SMF records.	YES | NO	NO
SMFDETAIL	Specify the type of detail sections in SMF Interval and Termination records (subtypes 2 and 3) that should be included in Adabas SMF records. All specifications (regardless of how many) for this parameter should be enclosed in parentheses. When NONE or ALL are specified for this parameter, they should be specified alone; no other values can be combined with NONE or ALL. Valid detail section names are ALL, CMD, FILE, IODD, NONE, PARM, SESS, STG, THRD, ZIIP and USER. List of section names must be separated by commas. In cluster environments (Adabas Cluster Services and Adabas Parallel Services), additional detail section names can be specified: CSHB, CSHF, CSHG, CSHP, LOCK, MSGB, MSGC, and MSGH.	(NONE | ALL | *section-name-list*)	(NONE)
SMFINTERVAL	Specify whether SMF interval records should be generated and the interval at which they should be generated.	NONE | GLOBAL | SUBSYS | *minutes*	NONE
SMFRECNO	Specify the Adabas SMF record number used for user-defined SMF records.	128 - 255	255
SMFSUBSYS	Specify the name of the IBM or user-defined SMF subsystem from PARMLIB member SMFPRM*xx*.	CURRENT | *subsystem-name*	CURRENT
SMGt	Indicate whether the error handling and message buffering facility should be enabled for the session.	YES | NO	NO
SOrtcache	Indicate whether to enable (YES) or disable (NO) controller caching for the Adabas Sort data set.	YES | NO	YES
SPt	Indicate whether use of Adabas triggers and stored procedures should be enabled for the session.	YES | NO	NO
SRlog	Indicate how spanned records should be logged to the protection logs.	ALL | UPD | PART	UPD
SVc	Specify the Adabas SVC number or Adabas Review hub SVC number to be used for the session.	200 - 255	249
TARgetid	Specify the unique Entire Net-Work target ID for this node.	1 - 65535	1

Parameter	Use to	Values	Default
TCPIp	Indicate whether a direct TCP/IP link to the Adabas nucleus should be activated for this session.	YES | NO	NO
TCPUrl	Identify the universal resource locator (URL) for the direct TCP/IP link to the Adabas nucleus (when TCPIP is set to "YES").	*url*	---
TEmpcache	Indicate whether to enable (YES) or disable (NO) controller caching for the Adabas Temp data set.	YES | NO	YES
TFlush	Specify the time, in seconds, to allow for a synchronous buffer flush.	1 - 30	1
TLscmd	Specify the maximum time, in seconds, to be used to process a single Adabas S1, S2, or S4 command with complex search criteria.	1 - the setting of the ADARUN MXTSX parameter	300
TMDrq	Set the number of entries allowed in the Adabas Transaction Manager's internal request queue. (Adabas Transaction Manager)	10 - 32767	10
TMEtdata	Identify the database or databases that will store Adabas Transaction Manager's ET data. (Adabas Transaction Manager)	ATM | TARGETS	TARGETS
TMGtt	Set the time limit, in seconds, during which a global transaction can be open without being prepared. (Adabas Transaction Manager)	1 - 16777215	720
TMLog	Identify the logging option for the current execution of the Adabas Transaction Manager. (Adabas Transaction Manager)	YES | NO | NEVER	NO
TMMsgsev	Set the severity threshold for the suppression of Adabas Transaction Manager warning messages. If this parameter is set to "0", no messages are suppressed. (Adabas Transaction Manager)	0 | 4 | 8	0
TMRestart	Identify how restart processing of problematic transactions should be handled. (Adabas Transaction Manager)	NORMAL | FORCE | FORCEALL	NORMAL
TMSyncmgr	Indicate whether the Adabas Transaction Manager is to interact with an external transaction coordinator. (Adabas Transaction Manager)	NONE | RRMS	NONE

Parameter	Use to	Values	Default
TMTcidpref	Define the first one or two characters (the prefix) of dynamically allocated client IDs. Up to two alphanumeric characters can be specified. (Adabas Transaction Manager)	*prefix*	TM
TNAA	Specify the maximum elapsed time, in seconds, that an access-only user may be considered active without issuing an Adabas command.	1 - 2147483647	900
TNAE	Specify the maximum elapsed time, in seconds, that an ET logic user may be considered active without issuing an Adabas command.	1 - 2147483647	900
TNAX	Specify the maximum elapsed time, in seconds, that an exclusive control user may be considered active without issuing an Adabas command.	1 - 2147483647	900
TT	Specify the maximum elapsed time, in seconds, permitted for a logical transaction issued by an ET logic user.	1 - 2147483647	900
UEX*nn*	Activate a user exit and specify the name of the user routine for the exit. The value of *nn* in the parameter name must be an integer in the range from 1 through 12, inclusive that identifies the number of the user exit. The value assigned the parameter must be a one to eight-character name of the corresponding user routine that gets control. **Note:** User exit 2 (UEX2) and user exit 12 (UEX12) are mutually exclusive.	*user-routine-name*	---
UEXSMF	Specify the name of the SMF user exit module.	*module-name*	---
UPdatecontrol	Specify whether or not to shortly delay the scheduling of update commands at the end of buffer flushes.	DELAY | NODELAY	DELAY
UTIonly	Indicate whether the session should be restricted to Adabas utilities only.	YES | NO	NO
V64bit	Indicate whether the Adabas nucleus should use virtual storage above the 2 gigabyte bar. In order for the nucleus to use 64-bit virtual storage, the operating system must also support 64-bit virtual storage.	YES | NO	NO
VIsta	Indicate whether support for Adabas Vista should be enabled. (Adabas Vista)	YES | NO	NO
WOrkcache	Indicate whether to enable (YES) or disable (NO) controller caching for the Adabas Work data set.	YES | NO	YES

Parameter	Use to	Values	Default
ZIIP	Indicate whether or not to activate Adabas for zIIP.	YES | NO	NO

ALOGDEV Parameter: Multiple Audit Logs Device Type

Parameter	Specify . . .	Possible Values	Default
ALOGDEV	The device type to be used for multiple audit logs.	See text	value of DEVICE parameter

The ALOGDEV parameter specifies the type of external storage device used for multiple audit logs.

Note: The ALOGDEV and ALOGSIZE parameters must be used together. If you specify one parameter, you must also specify the other parameter.

You must also specify the NALOG parameter to use multiple audit logs. If you specify a user exit, it must be UEX12.

Refer to the *Adabas for Mainframes* documentation > *Reference* > *DBA Reference Summary* > section *Adabas Device Types and Blocks Sizes* for information about the device types that Adabas supports.

All audit logging data sets must be formatted using the ADAFRM utility before they are used.

The maximum size of the data set depends on the limits of the operating system and on the requirements of the user.

Example

The following example runs the Adabas Audit Server with audit logging. It uses six audit log data sets and a device type 3390 for audit logging. The size of each audit log data set is 1500 blocks based on an allocation of 10 cylinders. Each cylinder of a 3390 device has 15 tracks, and each track can contain 10 blocks of audit log data. Thus, 10 cylinders = 10 x 15 x 10 = 1500 blocks.

```
ADARUN PROG=ADANUC,LOGGING=YES,NALOG=6,ALOGDEV=3390,ALOGSIZE=1500
```

ALOGSIZE Parameter: Multiple Audit Log Size

Parameter	Specify . . .	Minimum	Maximum	Default
ALOGSIZE	The number of blocks available for each audit log.	16	16777215	none

The ALOGSIZE parameters specifies the number of blocks available for each audit log data set.

Note: The ALOGDEV and ALOGSIZE parameters must be used together. If you specify one parameter, you must also specify the other parameter.

You must also specify the NALOG parameter to use multiple audit logs. If you specify a user exit, it must be UEX12.

Refer to the *Adabas for Mainframes* documentation > *Reference* chapter > *DBA Reference Summary* > section *Adabas Device Types and Blocks Sizes* for information about the device types that Adabas supports.

All audit logging data sets must be formatted using the ADAFRM utility before they are used.

The maximum size of the data set depends on the limits of the operating system and on the requirements of the user.

Example

The following example runs the Adabas Audit Server with audit logging. It uses six audit log data sets and a device type 3390 for audit logging. The size of each audit log data set is 1500 blocks based on an allocation of 10 cylinders. Each cylinder of a 3390 device has 15 tracks, and each track can contain 10 blocks of audit log data. Thus, 10 cylinders = 10 x 15 x 10 = 1500 blocks.

```
ADARUN PROG=ADANUC,LOGGING=YES,NALOG=6,ALOGDEV=3390,ALOGSIZE=1500
```

AOSLOG Parameter: Log Active Nucleus Update Commands

Parameter	Specify . . .	Possible Values	Default
AOSLOG	whether to log update commands from ADADBS OPERCOM or AOS that modify the active nucleus	YES | NO	NO

Value	Meaning
YES	Log to DDPRINT Adabas calls that modify the active nucleus (not read/display calls) issued by ADADBS OPERCOM or the equivalent Adabas Online System (AOS) functions.
NO	The default setting. Commands are not logged.

Example

In the following example, Adabas calls that modify the nucleus are logged.

```
ADARUN PROG=ADANUC,AOSLOG=YES
```

AREXCLUDE Parameter: Exclude Files from Autorestart Processing

Parameter	Specify . . .	Possible Values	Default
AREXCLUDE	the file(s) to exclude from autorestart processing.	1-5000	none

The AREXCLUDE parameter is intended for use only when autorestart processing fails (abends) due to response codes on one or more Adabas files. In this case, excluding the problem file(s) from autorestart might allow for successful autorestart processing on the other files.

Excluded files are marked as "utility use only". we recommend that you restore excluded files from the latest save tapes and regenerate them.

Multiple file numbers can be specified on the same parameter if separated by commas.

Example

The following example excludes files 199 and 231 from autorestart processing.

```
ADARUN PROG=ADANUC,AREXCLUDE=199,231
```

ARMNAME Parameter: Activate Automatic Restart Management (ARM) Program

Parameter	Specify . . .	Possible Values	Default
ARMNAME	and activate a particular automatic restart management (ARM) program.	*arm-pgm-name*	none

Automatic restart management (ARM) is used to automatically restart a nucleus when it abends. It is suppressed when the abend is intentional; for example, when it results from a parameter error.

ARM can be used in both cluster and noncluster nucleus environments.

ARM programs ('elements') are defined by the system programmer in the ARM 'policy'. They specify when, where, and how often a restart is to be attempted.

The ARMNAME parameter has no effect if an ARM policy has not been defined.

The name can be 1-16 characters long. In a cluster environment, the name must be unique across the cluster. Uppercase alphabetic characters, the numbers 0-9, and the special characters $, @, and #, and the underscore (_) may be used. The first character may not be numeric. Names that start with A through I and with SYS are reserved for use by IBM.

Example

The following example activates the automatic restart management program RESTART002.

```
ADARUN PROG=ADANUC,ARMNAME=RESTART002
```

ARNWORKBUFFERS Parameter: Autorestart WORK I/O Buffer Controls

Parameter	Specify . . .	Possible Values	Default
ARNWORKBUFFERS	the number of WORK I/O buffers allocated during autorestart processing after a failure.	0 - 2147483647	varies based on ADARUN V64BIT parameter setting.

When multiple WORK I/O buffers are allocated and used during autorestart processing, multiple WORK blocks may be read in each I/O. The use of multiple WORK buffers during autorestart processing may reduce the elapsed time needed to complete autorestart processing.

The buffers are allocated prior to the start of the backward repair phase of autorestart processing and deallocated at the end of autorestart processing.

If ARNWORKBUFFERS is set to zero or one, one WORK block will be read in each WORK read I/O during autorestart processing. The highest number of WORK I/O buffers allocated will be the number of WORK buffers in two cylinders.

In a noncluster nucleus, one set of buffers is allocated. In a cluster nucleus, one set of buffers is allocated for each cluster nucleus active at the time of the preceding failure. For example, if five cluster nuclei are active at the time all nuclei fail, five sets of WORK buffers are allocated.

When the ADARUN parameter V64BIT is set to YES, the autorestart WORK I/O buffers are allocated in virtual storage above the two gigabyte bar. When V64BIT=NO, the autorestart WORK I/O buffers are allocated in virtual storage above the 16 megabyte line and below the two gigabyte bar.

The default value for ARNWORKBUFFERS depends on the setting of the ADARUN V64BIT parameter.

- If V64BIT=NO, the default value for ARNWORKBUFFERS is one track worth of buffers. For example, on a 3390 device, nine WORK blocks fit in one track; in this example, the default value is "9".
- If V64BIT=YES, the default value for ARNWORKBUFFERS is two cylinders worth of buffers. For example, on a 3390 device, nine WORK blocks fit in one track and there are 15 tracks per cylinder. In this example, the default value is "270" (in other words, `9*15*2`).

ASSOCACHE Parameter: Associator Controller Caching

Parameter	Specify . . .	Possible Values	Default
ASSOCACHE	whether to enable (YES) or disable (NO) controller caching for the Associator (ASSO) data set.	YES | NO	YES

Value	Meaning
YES	The default setting. ASSO blocks are preread and maintained as long as possible in the caching facilities of the disk controller.
NO	ASSO blocks are not preread.

Notes:

1. If ASSO blocks are highly scattered or the update frequency is low, controller caching can cause longer I/O response times.
2. ADARUN caching control parameters also exist for Data Storage (DATACACHE), the work area (WORKCACHE), the sort area (SORTCACHE), and the temp area (TEMPCACHE).

Example

The following example runs the nucleus without controller caching for the ASSO data set.

```
ADARUN PROG=ADANUC,ASSOCACHE=NO
```

ASSOSPACEWARN Parameter: Associator Space Notifications

Parameter	Specify . . .	Possible Values	Default
ASSOSPACEWARN	Up to three pairs of threshold percentages and percentage change increments, expressed as percentages of the total Associator (ASSO) portion of the database. When ASSO space use reaches a specified threshold percentage, a space usage message is issued. In addition, when the ASSO space use changes by the increment specified for that threshold, additional space usage messages are issued. The full syntax of the ASSOSPACEWARN parameter is given below.	0-100,1-99	0,5

Note: For the purposes of database space notifications, the space used in the Adabas database comprises all database blocks that have been assigned either to some common purpose (for example, DSST, PPT, DLOG area) or to a file in the database. Only space listed in the free space table (FST) is not calculated as used. In particular, free space allocated for some file but not yet used is calculated as used.

The syntax of the ASSOSPACEWARN ADARUN parameter is:

ASSOSPACEWARN=*threshold1***[,***inc1***[,***threshold2***[,***inc2***[,***threshold3***[,***inc3***]]]]]**

The following specifications can be made in the ASSOSPACEWARN parameter:

Parameter	Description
threshold#	A threshold, expressed as a percentage of the total Associator portion of the database. When ASSO space use exceeds or falls below any of the thresholds set, a space usage message is issued. The default is zero (0). If zero is specified for the first threshold setting, no space messages are issued. When more than one threshold is specified, the second threshold specification must be greater than the first. Likewise, the third threshold specification (if made) must be greater than the second.
inc#	An increment percentage, expressed as a percentage of the total ASSO portion of the database. When ASSO space use changes (increases or decreases) from the associated threshold by the amount given in this increment percentage, another space usage message is issued. The default is five percent (5).

Up to three threshold-increment pairs can be specified.

Example

In the following ADARUN parameter example, a space usage message is printed to the console when the percentage of ASSO space used is greater than or equal to 50%, 60%, 70%, 80%, 90% and 100% of the total ASSO space available. When ASSO space use increases from under 50% to 55%, a space usage message is printed specifying the used percentage of 55%. If the ASSO space use then decreases to less than 50% (for example, to a value of 38%), a space usage message is printed indicating that the ASSO space usage is now 38%.

```
ADARUN PROG=ADANUC,ASSOSPACEWARN=50,10
```

ASYTVS Parameter: Asynchronous Buffer Flush

Parameter	Specify . . .	Possible Values	Default
ASYTVS	whether to flush buffers asynchronously based on volume serial number.	YES | NO	YES

Specific Product Recommendations

Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

Example

The following example disables the asynchronous buffer flush based on volume.

```
ADARUN PROG=ADANUC,ASYTVS=NO
```

AUDITING Parameter: Enable Nucleus Auditing

Parameter	Specify . . .	Possible Values	Default
AUDITING	Enables or disables Adabas nucleus auditing.	YES | NO	NO

The AUDITING parameter enables auditing on Adabas nuclei.

Note: For further information, refer to the documentation of *Adabas Auditing* > *Installation* chapter > *Activating Auditing for Application Files*.

Example

```
ADARUN PROG=ADANUC,AUDITING=YES
```

AUDITLOG Parameter: Audit of AOS and Utility Functions

Parameter	Specify . . .	Possible Values	Default
AUDITLOG	Whether auditing data of utility and AOS functions should be activated.	CONSOLE | NO	NO

This ADARUN parameter activates the writing of auditing data for utility and AOS functions. Specifying the option CONSOLE results in AAUD01 messages logged on the console for utility and AOS functions. The AAUD01 messages consist of the function and the user that initiated the call. AUDITLOG can be switched off by specifying NO or having no setting at all.

If a utility or AOS request affects multiple functions, the AAUD01 message mentions only the first requested function that is processed. For example, if your request is "reorder file", which is going to "reorder ASSO" and "reorder DATA" for that file, AAUD01 is going to indicate only the "re-ordered ASSO" function. Look for additional console or output messages indicating what was performed by the action.

Example

In the following example, AUDITLOG is activated and Audit data (AAUD01 messages) pertaining to the utility and AOS functions will be written to the console.

```
ADARUN AUDITLOG=CONSOLE
```

AUTOCQENV Parameter: Environment Affected by Command Return from the Command Queue

Parameter	Specify . . .	Possible Values	Default
AUTOCQENV	the type of user affected by the AUTOCQTIME parameter setting. Only commands from users in the specified environment will be returned from the command queue if they meet the AUTOCQTIME criteria.	ALL | CICS	ALL

The AUTOCQENV parameter and its supplement, AUTOCQTIME, have an effect only in cluster databases (i.e., Adabas Cluster Services or Adabas Parallel Services). After the failure of a nucleus in the cluster, while online recovery is in progress, surviving Adabas cluster nuclei stop selecting user commands from the command queue. During this time, new commands issued by users accumulate in the command queue and wait to be processed. The AUTOCQENV and AUTOCQTIME parameters can be used to prevent overly long delays for many commands.

When set to ALL, unprocessed commands from all users in the command queue are reviewed during online recovery to determine if they meet the criteria specified by the AUTOCQTIME parameter. When set to CICS, only unprocessed commands from CICS sessions are reviewed.

When a command in the command queue is returned to the user, no related information is written to the CLOG data set.

The setting of this parameter is local and fixed. It applies only to the Adabas nucleus for which it is set and it retains its value for the entire nucleus session.

Example

The following example will review unprocessed commands from CICS sessions. If the reviewed commands meet the criteria specified by the AUTOCQTIME parameter, the commands are returned to the user with response code 22 (ADARSP022), subcode 55.

```
ADARUN PROG=ADANUC,AUTOCQENV=CICS
```

AUTOCQTIME Parameter: Wait Time Threshold for Commands in Command Queue during Online Recovery

Parameter	Specify . . .	Possible Values	Default
AUTOCQTIME	the timeout period, in seconds, after which an eligible unprocessed command in the command queue during online recovery will be returned to the user with response code 22 (ADARSP022), subcode 55.	0 (zero) | 1 - 86400	0

The AUTOCQTIME parameter and its supplement, AUTOCQENV, have an effect only in cluster databases (i.e., Adabas Cluster Services or Adabas Parallel Services). After the failure of a nucleus in the cluster, while online recovery is in progress, surviving Adabas cluster nuclei stop selecting user commands from the command queue. During this time, new commands issued by users accumulate in the command queue and wait to be processed. The AUTOCQTIME and AUTOCQENV parameters can be used to prevent overly long delays for many commands.

When this parameter is set to 0 (zero), command time spans are not reviewed and commands remain in the command queue until they are processed.

When this parameter is set to a positive value, Adabas will, during online recovery, regularly review the command queue for unprocessed commands that have been waiting for the specified number of seconds or longer and meet the AUTOCQENV parameter criteria. Eligible unprocessed commands will be automatically returned with response code 22 (ADARSP022), subcode 55

When a command in the command queue is returned to the user, no related information is written to the CLOG data set.

The setting of this parameter is local and fixed. It applies only to the Adabas nucleus for which it is set and it retains its value for the entire nucleus session.

Example

The following example will return unprocessed commands in the command queue to the issuing user with response code 22, subcode 55 after the commands have been in the command queue for at least 60 seconds during an online recovery process following the failure of a cluster nucleus.

```
ADARUN PROG=ADANUC,AUTOCQTIME=60
```

AUTOINCASSOSIZE Parameter: Associator Automatic Database Increase Size

Parameter	Specify . . .	Possible Values	Default
AUTOINCASSOSIZE	the size, in cylinders or blocks, by which the highest-numbered (last-defined) Associator (ASSO) data set will be increased by an automatic database increase process.	0 - 2147483647 (cylinders) 0B - 2147483647B (blocks)	0

This parameter only has an effect if the AUTOINCASSOTHRESHOLD has a non-zero value. A numeric value without the suffix B indicates the increment is in cylinders. A numeric value followed by the letter B indicates the value specifies a number of blocks, which is then rounded up to the next highest amount that fills the final track or other platform-dependent minimum space allocation unit.

If an automatic database increase process is started to increase the ASSO component of the database, the last ASSO container data set will be increased by the number of cylinders or blocks specified by AUTOINCASSOSIZE. (The number of cylinders or blocks added may be less, however, if the limit set by the AUTOINCASSOTOTAL parameter is reached.)

If there is insufficient space at the end of the dataset, and the AUTOINCASSOSIZE value is greater than the secondary space allocation for the dataset, a new disk extent equivalent to the AUTOINCASSOSIZE value will be allocated.

If there is insufficient space at the end of the dataset, and the AUTOINCASSOSIZE value is less than or equal to the secondary space allocation for the dataset, a new disk extent equivalent to the secondary space value will be allocated.

If the last ASSO container data set cannot be increased by the specified size, but only by a lesser amount, the automatic increase process will increase the data set by the amount of space that it can get. Note that one data set can span at most 59 volumes with up to 16 extents on each volume. When these limits are reached, the database should be increased manually by adding a new container data set with the ADADBS ONLADD utility function.

When 4-byte RABNs are used, valid values for this parameter range from from zero (0) to 2147483647 blocks or the equivalent in cylinders. When 3-byte RABNs are used, the limits range from zero (0) to 16777215 blocks or the equivalent in cylinders. A value of "0" or "0B" means that no automatic increase will be performed for ASSO space. The default value for this parameter is zero, which functionally disables auto-increase for ASSO space.

If both AUTOINCASSOSIZE and the secondary space amount are set to a small value, the automatic database increase process may be initiated multiple times and cause the container allocation to reach platform-dependent limits on extents and volumes. Once that occurs, there can be no more auto-increases until the container allocations are consolidated, which requires stopping all nuclei using the container. Alternatively, a manual increase can be used to add another container without stopping the Adabas nucleus.

This parameter can be changed at runtime via operator command or corresponding AOS and ADADBS functions. In a cluster, this is a global parameter; its setting is the same for all nuclei in the cluster.

Example

In the following example, the highest-numbered ASSO data set (the last) will be increased in size by 500000 blocks when the AUTOINCASSOTHRESHOLD is reached.

```
ADARUN PROG=ADANUC,AUTOINCASSOSIZE=500000B ↩
```

AUTOINCASSOTHRESHOLD Parameter: Associator Space Threshold

Parameter	Specify . . .	Possible Values	Default
AUTOINCASSOTHRESHOLD	the Associator space threshold at which Adabas automatically initiates a database increase process. This threshold is expressed as a percentage of used Associator space to total Associator space available in the database. ASSO space is considered used if it is not listed as free in the Free Space Table (FST).	0, 50 - 100	0

If the used Associator space matches or exceeds the percentage defined by the specified AUTOINCASSOTHRESHOLD setting, Adabas automatically initiates a database increase process for ASSO space.

Note that the database increase process may take some time, the length of which is related to the size by which the database is increased (the value of the AUTOINCASSOSIZE setting). The AUTOINCASSOTHRESHOLD setting should be configured so that the increase process is given sufficient time to complete before the database fills up, assuming a reasonable space consumption rate.

Once a database increase process has been initiated, another automatic increase for ASSO space may be started only after the first process has finished and the ASSO space usage again (or still) matches or exceeds the specified threshold.

Valid values for this parameter are either zero (0) or integers ranging from 50 through 100. A value of zero means that no automatic increase will be performed for ASSO space. A value of 100 means that an automatic increase process will be initiated only when the ASSO space grows full (i.e., no free space is left at all). The default value for this parameter is zero, which functionally disables auto-increase processes for ASSO space.

This parameter can be changed at runtime via operator command or corresponding AOS and ADADBS functions. In a cluster, this is a global parameter; its setting is the same for all nuclei in the cluster.

Example

In the following example, a database increase process for ASSO space will be initiated when the used ASSO space matches or exceeds 70% of the total ASSO space defined.

```
ADARUN PROG=ADANUC,AUTOINCASSOTHRESHOLD=70
```

AUTOINCASSOTOTAL Parameter: Associator Space Maximum Size

Parameter	Specify . . .	Possible Values	Default
AUTOINCASSOTOTAL	the maximum total size, in cylinders or blocks, of the entire Associator space of the database that is not to be exceeded by automatic database increase processes.	0 - 2147483647 (cylinders) 0B - 2147483647B (blocks)	0

This parameter specifies the total maximum size, in cylinders or blocks, of the entire Associator space of the database. A numeric value followed by the letter B indicates the value specifies a number of blocks, which is then rounded up to the next highest amount that fills the final track or other platform-dependent minimum space allocation unit. A numeric value without a suffix B indicates the increment is in cylinders.

An automatic database increase process for ASSO space will not exceed the size specified by the AUTOINCASSOTOTAL parameter. This parameter can be used to limit the total amount of disk space used for the database, in order to protect the disk space remaining available for other databases and other users of the system. Note that this parameter limits only automatic increases by Adabas; it does not limit increasing the database manually using the ADADBS ADD, INCREASE, ONLADD or ONLINCREASE utility functions.

AUTOINCASSOTOTAL specifies the number of cylinders or blocks beyond which automatic database increase processes will not extend the ASSO space of the database. If AUTOINCASSO-

TOTAL is less than the current ASSO size plus AUTOINCASSOSIZE, an automatic increase will increase the ASSO space only by the difference between AUTOINCASSOTOTAL and the current size – that is, by less than the increase requested by the AUTOINCASSOSIZE parameter. If AUTOINCASSOTOTAL is not greater than the current ASSO size, no further automatic increases will be performed for ASSO space.

When 4-byte RABNs are used, valid values for this parameter range from from zero (0) to 2147483647 blocks or the equivalent in cylinders. When 3-byte RABNs are used, the limits range from zero (0) to 16777215 blocks or the equivalent in cylinders. A value of "0" or "0B" means that no automatic increase will be performed for ASSO space. The default value for this parameter is zero, which functionally disables auto-increase for ASSO space.

The total size of ASSO space may be constrained to less than the AUTOINCASSOTOTAL value for either of the following reasons.

- There may not be enough storage resources to extend to the specified limit.
- When using 4-byte RABNs, a component can have at most 2147483647 blocks, and for 3-byte RABNs the limit is 16777215 blocks.

This parameter can be changed at runtime via operator command or corresponding AOS and ADADBS functions. In a cluster, this is a global parameter; its setting is the same for all nuclei in the cluster.

Example

In the following example, the total ASSO space for a database is set to 16777215 blocks.

```
ADARUN PROG=ADANUC,AUTOINCASSOTOTAL=16777215B
```

AUTOINCDATASIZE Parameter: Data Storage Automatic Database Increase Size

Parameter	Specify . . .	Possible Values	Default
AUTOINCDATASIZE	the size, in cylinders or blocks, by which the highest-numbered (last-defined) Data Storage (DATA) data set will be increased by an automatic database increase process.	0 - 2147483647 (cylinders) 0B - 2147483647B (blocks)	0

This parameter only has an effect if the AUTOINCDATATHRESHOLD has a non-zero value. A numeric value without the suffix B indicates the increment is in cylinders. A numeric value followed by the letter B indicates the value specifies a number of blocks, which is then rounded up to the next highest amount that fills the final track or other platform-dependent minimum space allocation unit.

If an automatic database increase process is started to increase the DATA component of the database, the last DATA container data set will be increased by the number of cylinders or blocks specified by AUTOINCDATASIZE. (The number of cylinders or blocks added may be less, however, if the limit set by the AUTOINCDATATOTAL parameter is reached.)

If there is insufficient space at the end of the dataset, and the AUTOINCDATASIZE value is greater than the secondary space allocation for the dataset, a new disk extent equivalent to the AUTOINCDATASIZE value will be allocated.

If there is insufficient space at the end of the dataset, and the AUTOINCDATASIZE value is less than or equal to the secondary space allocation for the dataset, a new disk extent equivalent to the secondary space value will be allocated.

If the last DATA container data set cannot be increased by the specified size, but only by a lesser amount, the automatic increase process will increase the data set by the amount of space that it can get. Note that one data set can span at most 59 volumes with up to 16 extents on each volume. When these limits are reached, the database should be increased manually by adding a new container data set with the ADADBS ONLADD utility function.

Even though the increase is for the DATA component, a response 77 is still possible for ASSO if the increase requires an ASSO extent to be taken and no ASSO space is available. Therefore, it is suggested to make sure ASSO storage is available as well.

When 4-byte RABNs are used, valid values for this parameter range from from zero (0) to 2147483647 blocks or the equivalent in cylinders. When 3-byte RABNs are used, the limits range from zero (0) to 16777215 blocks or the equivalent in cylinders. A value of "0" or "0B" means that no automatic increase will be performed for DATA space. The default value for this parameter is zero, which functionally disables auto-increase for DATA space.

If both AUTOINCDATASIZE and the secondary space amount are set to a small value, the automatic database increase process may be initiated multiple times and cause the container allocation to reach platform-dependent limits on extents and volumes. Once that occurs, there can be no more auto-increases until the container allocations are consolidated, which requires stopping all nuclei using the container. Alternatively, a manual increase can be used to add another container without stopping the Adabas nucleus.

This parameter can be changed at runtime via operator command or corresponding AOS and ADADBS functions. In a cluster, this is a global parameter; its setting is the same for all nuclei in the cluster.

Example

In the following example, the highest-numbered Data Storage data set (the last) will be increased in size by 5000 cylinders when the AUTOINCDATATHRESHOLD is reached.

```
ADARUN PROG=ADANUC,AUTOINCDATASIZE=5000
```

AUTOINCDATATHRESHOLD Parameter: Data Storage Space Threshold

Parameter	Specify . . .	Possible Values	Default
AUTOINCDATATHRESHOLD	the Data Storage space threshold at which Adabas automatically initiates a database increase process. This threshold is expressed as a percentage of used Data Storage space to total Data Storage space available in the database. DATA space is considered used if it is not listed as free in the Free Space Table (FST).	0, 50 - 100	0

If the used Data Storage space matches or exceeds the percentage defined by the specified AUTOINCDATATHRESHOLD setting, Adabas automatically initiates a database increase process for DATA space.

Note that the database increase process may take some time, the length of which is related to the size by which the database is increased (the value of the AUTOINCDATASIZE setting). The AUTOINCDATATHRESHOLD setting should be configured so that the increase process is given sufficient time to complete before the database fills up, assuming a reasonable space consumption rate.

Once a database increase process has been initiated, another automatic increase for DATA space may be started only after the first process has finished and the DATA space usage again (or still) matches or exceeds the specified threshold.

Valid values for this parameter are either zero (0) or integers ranging from 50 through 100. A value of zero means that no automatic increase will be performed for DATA space. A value of 100 means that an automatic increase process will be initiated only when the DATA space grows full (i.e., no free space is left at all). The default value for this parameter is zero, which functionally disables auto-increase processes for DATA space.

This parameter can be changed at runtime via operator command or corresponding AOS and ADADBS functions. In a cluster, this is a global parameter; its setting is the same for all nuclei in the cluster.

Example

In the following example, a database increase process for DATA space will be initiated when the used DATA space matches or exceeds 75% of the total DATA space defined.

```
ADARUN PROG=ADANUC,AUTOINCDATATHRESHOLD=75
```

AUTOINCDATATOTAL Parameter: Data Storage Space Maximum Size

Parameter	Specify . . .	Possible Values	Default
AUTOINCDATATOTAL	the maximum total size, in cylinders or blocks, of the entire Data Storage space of the database that is not to be exceeded by automatic database increase processes.	0 - 2147483647 (cylinders) 0B - 2147483647B (blocks)	0

This parameter specifies the total maximum size, in cylinders or blocks, of the entire Data Storage space of the database. A numeric value followed by the letter B indicates the value specifies a number of blocks, which is then rounded up to the next highest amount that fills the final track or other platform-dependent minimum space allocation unit. A numeric value without a suffix B indicates the increment is in cylinders.

An automatic database increase process for DATA space will not exceed the size specified by the AUTOINCDATATOTAL parameter. This parameter can be used to limit the total amount of disk space used for the database, in order to protect the disk space remaining available for other databases and other users of the system. Note that this parameter limits only automatic increases by Adabas; it does not limit increasing the database manually using the ADADBS ADD, INCREASE, ONLADD or ONLINCREASE utility functions.

AUTOINCDATATOTAL specifies the number of cylinders or blocks beyond which automatic database increase processes will not extend the DATA space of the database. If AUTOINCDATATOTAL is less than the current DATA size plus AUTOINCDATASIZE, an automatic increase will increase the DATA space only by the difference between AUTOINCDATATOTAL and the current size – that is, by less than the increase requested by the AUTOINCDATASIZE parameter. If AUTOINCDATATOTAL is not greater than the current DATA size, no further automatic increases will be performed for DATA space.

When 4-byte RABNs are used, valid values for this parameter range from from zero (0) to 2147483647 blocks or the equivalent in cylinders. When 3-byte RABNs are used, the limits range from zero (0) to 16777215 blocks or the equivalent in cylinders. A value of "0" or "0B" means that no automatic increase will be performed for DATA space. The default value for this parameter is zero, which functionally disables auto-increase for DATA space.

The total size of DATA space may be constrained to less than the AUTOINCDATATOTAL value for either of the following reasons.

- There may not be enough storage resources to extend to the specified limit.
- When using 4-byte RABNs, a component can have at most 2147483647 blocks, and for 3-byte RABNs the limit is 16777215 blocks.

This parameter can be changed at runtime via operator command or corresponding AOS and ADADBS functions. In a cluster, this is a global parameter; its setting is the same for all nuclei in the cluster.

Example

In the following example, the total DATA space for a database is set to 100000 cylinders.

```
ADARUN PROG=ADANUC,AUTOINCDATATOTAL=100000
```

CACHE Parameter: Adabas Caching Facility Control

`CACHE` (or `CACH`) specifies whether or not ADACSH is to be loaded during Adabas session initialization.

CACHE = { YES | NO }

Value	Description
YES	ADACSH is loaded and available for operation.
NO	ADACSH is not loaded. Caching is not available until the Adabas session is terminated and subsequently restarted with `CACHE=YES`.

Example

```
ADARUN CACHE=YES
```

ADACSH is loaded during the Adabas session initialization. ADACSH operator commands are accepted and caching is in effect.

CACTIVATE Parameter: Adabas Caching Facility RABN Activation

`CACTIVATE` (or `CACT`) specifies whether or not *all* RABN ranges are to be activated at session open.

CACTIVATE = { YES | NO }

Value	Description
YES	All RABN ranges are activated at session open. This provides for immediate use of the cache without consideration of the `CDEMAND` threshold level.
NO	None of the RABN ranges is activated unless: ■ the `CDEMAND` threshold level is reached; or ■ the `CENABLE` operator command is accepted by ADACSH.

Example

```
ADARUN CACTIVATE=YES
```

All RABN ranges are made available for immediate use; that is, as soon as the Adabas nucleus needs to move a block from the Adabas buffer pool into a RABN range, the RABN range is allocated.

CASSODSP / CASSOEXT / CASSOHSP / CASSOG64 / CASSOL64 / CASSOP64 /CASSOV64 Parameters: Adabas Caching Facility Associator RABN Cache Space Control

{ CASSODSP | CASSOEXT | CASSOHSP | CASSOG64 | CASSOL64 |CASSOP64 | CASSOV64 }

Note: `CASSO`*`xxx`* parameters are mutually exclusive with the `CFILE` and `CXFILE` parameters.

The `CASSO`*`xxx`* parameters identify one or more Associator RABNs to be cached where *`xxx`* indicates the type of cache space.

- `CASSODSP` (or `CASSOD`) caches the Associator RABNs in data space.
- `CASSOEXT` (or `CASSOE`) caches the Associator RABNs in extended memory.
- `CASSOHSP` (or `CASSOH`) caches the Associator RABNs in hiperspace.
- `CASSOG64` (or `CASSOG`) caches the Associator RABNs in virtual 64-bit storage backed by 2G large pages.

  Note: If "G64" is specified, we recommend you specify a cache space size (`CASSOMAXS`) that is a multiple of two gigabytes. The specified size will be rounded up to the nearest multiple of two gigabytes and any difference between the specified size and the allocated size will remain unused. In this case, specifying a multiple of two gigabytes avoids leaving a potentially large amount of storage unused. If "G64" is specified and 2G large pages are not

available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

- `CASSOL64` (or `CASSOL`) caches the Associator RABNs in virtual 64-bit storage backed by fixed 1M large pages.
- `CASSOP64` (or `CASSOP`) caches the Associator RABNs in virtual 64-bit storage backed by pageable 1M large pages.
- `CASSOV64` (or `CASSOV`) caches the Associator RABNs in virtual 64-bit storage backed by pageable 4K pages.

{ CASSODSP | CASSOEXT | CASSOHSP | CASSOG64 | CASSOL64 | CASSOP64 | CASSOV64 } = { rabn | rabn - rabn } , ...

Value	Description
rabn	The Associator RABN to be cached.
rabn - rabn	The range of Associator RABNs to be cached. The range is specified as two numbers separated by a hyphen with no intervening spaces or special characters. More than one RABN range (separated by a comma) may be specified on the same line. There are no continuations. If more than one line is required, additional `CASSO`*xxx* parameters must be specified.

If `CASSO`*xxx* is specified, *only* the specified RABNs are cached. Unspecified RABNs are *not* cached and are subject to normal Adabas I/O operations.

Associator RABNs 1 through 30 are not cached; they are ignored if specified by the user.

Notes:

1. If an invalid RABN range is specified in the `CASSOxxx` parameter, subsequent RABN ranges for the `CASSO`*xxx* parameter are ignored. The ignored parameters can be added after initialization by using an operator command.
2. The `CSTORAGE` parameter overrides any `CASSO`*xxx* parameter and ADACSH will cache all Associator RABNs as defined for the database.

RABNs for the ranges specified are cached in the cache space indicated by the parameter suffix. RABN ranges defined with this parameter must therefore compete for the space available in a given cache space. When the maximum is allocated, RABNs are purged from the cache space based on a first-in-first-out (FIFO) mechanism and are reused to accommodate the latest RABNs being purged from the Adabas buffer pool.

Example

```
 ADARUN CASSODSP=1330-2349,126-187,124-125,205-216
ADARUN CASSOHSP=218-252,203-204,260-275,277-335
```

ADACSH caches the specified RABN ranges. No other Associator RABNs are cached.

The first four RABN ranges defined are cached in the Associator cache space allocated in data space (suffix DSP). The second four RABN ranges defined are cached in the Associator cache space allocated in hiperspace (suffix HSP).

CASSOMAXS Parameter: Associator Cache Space Unit Size

The `CASSOMAXS` (or `CASSOM`) parameter specifies the unit size in which areas are allocated for Associator cache space.

CASSOMAXS = *n*

Value	Description
n	The unit size in which areas are allocated for Associator cache space. Size may be specified in bytes, in kilobytes (K), in megabytes (M) or in gigabytes (G). For extended memory, data spaces, and hiperspaces: ■ the minimum unit size is 81920 bytes (or 80K); ■ the maximum unit size is 2147475456 bytes (or 2097144K; or 2047M); and ■ the default unit size is 1048576 bytes (or 1024K; or 1M). ■ In these cases, `CASSOMAXS` is always rounded up to the nearest 4-kilobyte boundary. For virtual 64 memory and virtual 64 backed by large pages, virtual storage above the 2G bar is allocated in terms of memory objects, each of which comprises one or more virtual segments or pages of one megabyte each. In essence, `CASSOMAXS` determines the number of 1-megabyte segments to be allocated for the memory object: ■ the minimum unit size and the default unit size is 1M (1048576 bytes or 1024K). ■ the maximum unit size depends on the limit set for your installation. ■ For memory objects, `CASSOMAXS` is rounded up to the nearest megabyte boundary. For cache spaces for which storage is backed by 2G large pages ("G64"), `CASSOMAXS` is rounded up to the nearest multiple of two gigabytes. **Note:** If "G64" is specified for a cache space, we recommend that you specify a `CASSOMAXS` value that is a multiple of two gigabytes. The specified size will be rounded up to the nearest multiple of two gigabytes and any difference between the specified size and the allocated size will remain unused. In this case, specifying a multiple of two gigabytes avoids leaving a potentially large amount of storage unused. If "G64" is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

The total space available is equal to:

```
Maximum-ASSO-cache-space-size = CASSOMAXS x CMAXCSPS
```

Storage areas for a cache space are allocated when required. The Adabas nucleus informs ADACSH when a RABN is about to be purged from the buffer pool. If ADACSH determines that this RABN is being cached, it attempts to allocate space in the appropriate cache space.

When no space is available in the currently allocated areas for a cache space, or no areas are allocated, a new area of storage is allocated with the size specified by the `CASSOMAXS` parameter. For cache storage in

- extended memory, additional storage is requested in this area;
- data space, an additional data space is allocated;
- hiperspace, an additional hiperspace is built; and
- virtual 64 memory backed by large pages, an additional memory object is allocated; and
- virtual 64 memory, an additional memory object is allocated.

This occurs until the maximum number of areas as determined by the `CMAXCSPS` parameter is reached. At this point, space is made for the latest RABNs to be purged from the Adabas buffer pool based on the aging algorithms.

The `CASSOMAXS` limit used for the allocation of new areas in a cache space is based on the last specification for this value. If no operator or Adabas Online System command is issued, this is the value specified in the startup parameters. If this value is specified by an operator command or through Adabas Online System, it takes effect the next time an additional area is allocated in an ASSO cache space.

Example

```
ADARUN CASSOMAXS=15000000
```

Cache space areas for ASSO RABNs are allocated in units of 15003648 bytes (or 14652K; or approximately 14M).

CBUFNO Parameter: Number of Read-Ahead Buffers for Concurrent I/O Processing

`CBUFNO` specifies the number of read-ahead buffers available for concurrent I/O processing.

`CBUFNO` is one of two parameters (`CBUFNO` and `CEXCLUDE`) used for read-ahead caching. Read-ahead caching performs a read-ahead of consecutive Associator and/or Data Storage blocks when processing serial Adabas commands (`Read Logical`, `Read Physical`, `Histogram`), and when searching using nondescriptors.

The read-ahead is performed by issuing a single EXCP to read all the consecutive Associator or Data Storage blocks that reside on a single track of the disk device. The blocks are kept in cache and are immediately available when the nucleus requests the next block in a sequence.

This feature may enhance performance by reducing the number of physical read I/Os. For example, for a 3390 Associator, a reduction factor of 17:1 may be achieved.

CBUFNO = *n*

Value	Description
n	The number of read-ahead buffers available for concurrent I/O processing. The recommended setting is a value that accommodates the expected number of concurrent `L2`, `L3`, and `L9` commands. The minimum number of read-ahead buffers is 0 (the default). If `CBUFNO=0` is specified, *no* read-ahead caching is performed. The maximum number is the same as the number of threads specified for the nucleus parameter `NT`, because there cannot be more than `NT` number of commands active at any time.

During normal cache processing, if a read I/O request qualifies for read-ahead processing but no read-ahead buffer is available, the read I/O is satisfied with a single block EXCP. In other words, no optimization is done to issue an EXCP to read the entire track.

Note: Read-end-of-track processing is not available in Cluster and SMP environments.

Example

```
ADARUN CBUFNO=5,CEXCLUDE=FIND/HIST
```

Five buffers are available for read-ahead caching. No read-ahead processing is performed for `S1` (nondescriptor search) and `L9` (histogram) commands.

CCTIMEOUT Parameter: Cache Space Area or RABN Inactivity Time

The `CCTIMEOUT` (or `CC`) parameter specifies the amount of time (in seconds) a cache space area can remain inactive before it is released back to the system.

If demand caching (`CDEMAND`) is in effect, `CCTIMEOUT` specifies the number of seconds a RABN range, file, or file range can remain inactive before it is disabled.

CCTIMEOUT = *n*

Value	Description
n	The amount of time (in seconds) a cache space area can remain inactive before it is released back to the system. If demand caching (`CDEMAND`) is in effect, `CCTIMEOUT` specifies the number of seconds a RABN range, file, or file range can remain inactive before it is disabled. The minimum number of seconds is 60; the maximum is 2147483647. The default number of seconds is 7200 (2 hours).

If a cache space area has not been accessed for the specified period of time, ADACSH releases the storage associated with the area back to the operating system. A message is displayed on the operator console indicating that the cache space area was timed out. The cache space area can be reallocated immediately by the caching system when required.

If a RABN range, file, or file range is not referenced for the specified period of time, and demand caching is in effect, the RABN range, file, or file range is disabled. A message is displayed on the operator console informing the user that the RABN range, file, or file range was timed out. No further caching occurs for that RABN range, file, or file range until the Adabas buffer efficiency goes below the level specified by `CDEMAND`. Once this occurs, the RABN range, file, or file range is enabled again and begins caching its associated RABNs.

Activity is checked at an interval of one half of the `CCTIMEOUT` value. For example, if `CCTIMEOUT=3600` is specified, the allocated cache space areas, RABN ranges, files, or file ranges are checked every 1800 seconds (30 minutes).

A change to `CCTIMEOUT` using an operator command causes the check to be performed immediately, and subsequently at intervals of one half of the new `CCTIMEOUT` value.

Example

```
ADARUN CCTIMEOUT=3600
```

A cache space area is released (or a RABN range/file/file range is disabled if demand caching is in effect) if there is no activity associated with it for a continuous period of 1 hour (3600 seconds).

CDATADSP / CDATAEXT / CDATAHSP / CDATAG64 / CDATAL64 / CDATAP64 / CDATAV64 Parameter: Adabas Caching Facility Data Storage RABN Cache Space Control

{ CDATADSP | CDATAEXT | CDATAHSP | CDATAG64 | CDATAL64 |CDATAP64 | CDATAV64 }

Note: `CDATA`*`xxx`* parameters are mutually exclusive with the `CFILE` and `CXFILE` parameters.

The `CDATA`*`xxx`* parameters identify one or more Data Storage RABNs to be cached where *`xxx`* indicates the type of cache space.

- `CDATADSP` (or `CDATAD`) caches the Data Storage RABNs in data space.
- `CDATAEXT` (or `CDATAE`) caches the Data Storage RABNs in extended memory.
- `CDATAHSP` (or `CDATAH`) caches the Data Storage RABNs in hiperspace.
- `CDATAG64` (or `CDATAG`) caches the Data Storage RABNs in virtual 64-bit storage backed by 2G large pages.

  Note: If "G64" is specified, we recommend you specify a cache space size (`CDATAMAXS`) that is a multiple of two gigabytes. The specified size will be rounded up to the nearest multiple of two gigabytes and any difference between the specified size and the allocated size will remain unused. In this case, specifying a multiple of two gigabytes avoids leaving a potentially large amount of storage unused. If "G64" is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

- `CDATAL64` (or `CDATAL`) caches the Data Storage RABNs in virtual 64-bit storage backed by fixed 1M large pages.
- `CDATAP64` (or `CDATAP`) caches the Data Storage RABNs in virtual 64-bit storage backed by pageable 1M large pages.
- `CDATAV64` (or `CDATAV`) caches the Data Storage RABNs in virtual 64-bit storage backed by pageable 4K pages.

{ CDATADSP | CDATAEXT | CDATAHSP | CDATAG64 | CDATAL64 | CDATAP64 | CDATAV64 } = { rabn | rabn - rabn } , ...

Value	Description
rabn	The Data Storage RABN to be cached.
rabn - rabn	The range of Data Storage RABNs to be cached. The range is specified as two numbers separated by a hyphen with no intervening spaces or special characters. More than one RABN range (separated by a comma) may be specified on the same line. There are no continuations. If more than one line is required, additional `CDATA`*xxx* parameters must be specified.

If `CDATA`*xxx* is specified, *only* the specified RABNs are cached. Unspecified RABNs are *not* cached and are subject to normal Adabas I/O operations.

Notes:

1. If an invalid RABN range is specified in the `CDATA`*xxx* parameters, subsequent RABN ranges for the `CDATA`*xxx* parameters will be ignored. The ignored parameters can be added after initialization by using an operator command.
2. The `CSTORAGE` parameter overrides any `CDATA`*xxx* parameter and ADACSH will cache all Data Storage RABNs as defined for the database.

RABNs for the ranges specified are cached in the cache space indicated by the parameter suffix. RABN ranges defined with this parameter must therefore compete for the space available in a given cache space. When the maximum is allocated, RABNs are purged from the cache space based on a first-in-first-out (FIFO) mechanism and are reused to accommodate the latest RABNs being purged from the Adabas buffer pool.

Example

```
 ADARUN CDATADSP=1-4900,6523-7853
ADARUN CDATAHSP=5032-6204,8123-10382
```

ADACSH caches the specified RABN ranges. No other DATA RABNs are cached.

The first two RABN ranges defined are cached in the Data Storage cache space allocated in data space (suffix DSP). The second two RABN ranges defined are cached in the Data Storage cache space allocated in hiperspace (suffix HSP).

CDATAMAXS Parameter: Data Storage Cache Space Unit Size

The `CDATAMAXS` (or `CDATAM`) parameter specifies the unit size in which areas are allocated for Data Storage cache space.

CDATAMAXS = *n*

Value	Description
n	The unit size in which areas are allocated for Data Storage cache space. Size may be specified in bytes, in kilobytes (K), in megabytes (M) or in gigabytes (G). For extended memory, data spaces, and hiperspaces: ■ the minimum unit size is 81920 bytes (or 80K); ■ the maximum unit size is 2147475456 bytes (or 2097144K; or 2047M); and ■ the default unit size is 1048576 bytes (or 1024K; or 1M). ■ In these cases, `CDATAMAXS` is always rounded up to the nearest 4-kilobyte boundary. For virtual 64 memory and virtual 64 backed by large pages, virtual storage above the 2G bar is allocated in terms of memory objects, each of which comprises one or more virtual segments or pages of one megabyte each. In essence, `CDATAMAXS` determines the number of 1-megabyte segments to be allocated for the memory object: ■ the minimum unit size and the default unit size is 1M (1048576 bytes or 1024K). ■ the maximum unit size depends on the limit set for your installation. ■ For memory objects, `CDATAMAXS` is rounded up to the nearest megabyte boundary. For cache spaces for which storage is backed by 2G large pages ("G64"), `CDATAMAXS` is rounded up to the nearest multiple of two gigabytes. **Note:** If "G64" is specified for a cache space, we recommend that you specify a `CDATAMAXS` value that is a multiple of two gigabytes. The specified size will be rounded up to the nearest multiple of two gigabytes and any difference between the specified size and the allocated size will remain unused. In this case, specifying a multiple of two gigabytes avoids leaving a potentially large amount of storage unused. If "G64" is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

The total space available is equal to:

```
Maximum-DATA-cache-space-size = CDATAMAXS x CMAXCSPS
```

Storage areas for a cache space are allocated when required. The Adabas nucleus informs ADACSH when a RABN is about to be purged from its buffer pool. If ADACSH determines that this RABN is being cached, it attempts to allocate space in the appropriate cache space.

When no space is available in the currently allocated areas for a cache space, or no areas are allocated, a new area of storage is allocated with the size specified by the `CASSOMAXS` parameter. For cache storage in

- extended memory, additional storage is requested in this area;
- data space, an additional data space is allocated;
- hiperspace, an additional hiperspace is built; and
- virtual 64 memory backed by large pages, an additional memory object is allocated; and
- virtual 64 memory, an additional memory object is allocated.

This occurs until the maximum number of areas as determined by the `CMAXCSPS` parameter is reached. At this point, space is made for the latest RABNs to be purged from the Adabas buffer pool based on the aging algorithms.

The `CDATAMAXS` limit used for the allocation of new areas in a cache space is based on the last specification for this value. If no operator or Adabas Online System command is issued, this is the value specified in the startup parameters. If this value is specified by an operator command or through Adabas Online System, it takes effect the next time an additional area is allocated in a Data Storage cache space.

Example

```
ADARUN CDATAMAXS=15000000
```

Cache space areas for Data Storage RABNs are allocated in units of 15003648 bytes (or 14652K; or approximately 14M).

CDEMAND Parameter: Lowest Adabas Buffer Efficiency for Caching

The `CDEMAND` (or `CDE`) parameter specifies the lowest Adabas buffer efficiency level accepted before caching is activated.

Calculated by the nucleus, Adabas buffer efficiency is the number of read requests divided by the actual number of reads to storage. If the Adabas buffer efficiency level drops below the specified `CDEMAND` level, ADACSH activates all disabled RABN ranges. In this way, RABN ranges are dynamically activated as demand on the Adabas buffer pool increases.

The `CDEMAND` parameter should be used in conjunction with the `CCTIMEOUT` parameter to establish resource usage boundaries that permit the timely acquisition and release of system resources. In

other words, CDEMAND may be used to activate RABN ranges when the demand load on Adabas is high, and CCTIMEOUT may be used to deactivate RABN ranges when Adabas is idle.

Value	Description
0	If CDEMAND=0 (the default), *no* demand-level caching occurs.
n	Demand-level caching is to be activated whenever the Adabas buffer efficiency drops below the value specified. An integer value between 1 and 2147483647 is permitted.

If CDEMAND is not zero, the Adabas buffer efficiency is first checked approximately 30 minutes after the Adabas nucleus is initialized. Thereafter, the Adabas buffer efficiency is checked at intervals of approximately 15 minutes.

CDEMAND does not affect the activation of caching for Work parts 2 and 3 because these areas do not influence the Adabas buffer efficiency. Only Associator and Data Storage blocks reside in the Adabas buffer pool and, consequently, only the cache spaces for the Associator and Data Storage serve to increase the Adabas buffer efficiency level.

Example

```
ADARUN CDEMAND=7
```

ADACSH activates the RABN ranges if the Adabas buffer efficiency level drops below 7.

CDISPSTAT Parameter: Adabas Caching Facility RABN Range Statistics Display Control

The CDISPSTAT (or CDI) parameter specifies whether the RABN range statistics are to be displayed on both the operator console and DDPRINT, or only on DDPRINT.

The individual RABN range statistics are displayed whenever a RABN range is released due to a parallel utility operation or an inactivity timeout, or when a RABN range is dynamically added or deleted.

CDISPSTAT = { YES | NO }

Value	Description
YES	RABN range statistics are to be displayed on both the operator console and DDPRINT.
NO	RABN range statistics are to be displayed only on DDPRINT.

Example

```
ADARUN CDISPSTAT=NO
```

When the status of a RABN range is changed due to a parallel utility operation, a timeout, or an operator command to add or delete, the individual RABN range statistics are to be displayed on DDPRINT only.

CDXnn Parameter: Collation Descriptor User Exit

Parameter	Specify . . .	Possible Values	Default
CDX*nn*	the collation descriptor exit and the user routine to be used with it.	see text	none

Used in conjunction with the universal encoding subsystem (UES), the format of the collation descriptor user exit parameter is

```
CDXnn=exit-name
```

where

nn	is the number of the collation descriptor exit, a two-digit decimal integer in the range 01-08 inclusive.
exit-name	is the name of the user routine that gets control at the collation descriptor exit; the name can be up to 8 characters long.

Only one program may be specified for each collation descriptor exit. Up to 8 collation descriptor exits may be specified (in any order). See the *Adabas User, Hyperdescriptor, Collation Descriptor, and SMF Exits Manual* for more information.

Example

The following example runs the Adabas nucleus using collation descriptor exits 1 and 2. It also gives control to the user routine UCOLDE1 at collation descriptor exit 1 and gives control to the user routine UCOLDE2 at collation descriptor exit 2.

```
ADARUN PROG=ADANUC,CDX01=UCOLDE1,CDX02=UCOLDE2
```

CEXCLUDE Parameter: Adabas Caching Facility Excluded Command Types

CEXCLUDE specifies the command types to be excluded from read-ahead caching.

CEXCLUDE is one of two parameters (CBUFNO and CEXCLUDE) used for read-ahead caching. Read-ahead caching performs a read-ahead of consecutive Associator and/or Data Storage blocks when processing serial Adabas commands (Read Logical, Read Physical, Histogram), and when searching using nondescriptors.

The read-ahead is performed by issuing a single EXCP to read all the consecutive Associator or Data Storage blocks that reside on a single track of the disk device. The blocks are kept in cache and are immediately available when the nucleus requests the next block in a sequence.

This feature may enhance performance by reducing the number of physical read I/Os. For example, for a 3390 Associator, a reduction factor of 17:1 may be achieved.

CEXCLUDE = { PHYS | LOGI | HIST | FIND } / ...

Value	Description
PHYS	Exclude L2 and L5 (physical sequence) commands.
LOGI	Exclude L3 and L6 (logical sequence) commands.
HIST	Exclude L9 (histogram) command.
FIND	Exclude S1 (nondescriptor search) command.

The exclusion list must be specified with each four-byte name separated by a slash (/). No intervening spaces or special characters, such as commas, are permitted.

The exclusion is made globally; for example, if CEXCLUDE=PHYS is specified, all L2 and L5 commands are excluded from read-ahead processing, regardless of the file to which the command was directed.

The exclusion list may be changed during the nucleus session using the CEXCLUDE and CINCLUDE operator commands.

Example

```
ADARUN CBUFNO=5,CEXCLUDE=FIND/HIST
```

Five buffers are available for read-ahead caching. No read-ahead processing is performed for S1 (nondescriptor search) and L9 (histogram) commands.

CFILE Parameter: Adabas Caching Facility File Range

Note: `CFILE` is mutually exclusive with the `CASSO`*xxx*, `CDATA`*xxx*, and `CSTORAGE` parameters. If any of these parameters is used with `CFILE`, they will be ignored.

The `CFILE` parameter defines a file or file range to be cached. It is also used to provide the settings used to control how caching is to be implemented for the file or file range.

CFILE = ({ *fnr* | *fnr* - *fnr* } , [*cls*] , [*scope*] , [*cache*])

Value	Description
fnr	The file number of the file to be cached. The minimum value is 0; the maximum is the actual number of files in the database. File number 0 may be specified to cache RABNs that are not associated with any given file. Currently, this is only the DSST on the Associator data set.
fnr - *fnr*	Range of files to be cached specified by two file numbers (the lower number first) separated by a hyphen with no intervening spaces or special characters.
cls	Class of service. The percentage of available cache space that can be used to cache RABNs associated with the specified file(s). This term effectively assigns a priority to the file(s). RABNs belonging to the file(s) with the lowest priority are purged first. Valid values are: ■ 1: 100% (highest priority) ■ 2: 75% ■ 3: 50%; the default ■ 4: 25% ■ 5: 10% (lowest priority)
scope	Caching scope. Valid values are: ■ ASSO (or A) : Caching is to be used only for Associator for the file ■ DATA (or D) : Caching is to be used only for Data Storage for the file ■ BOTH (or B) : Caching is to be used for both Associator and Data Storage for the file The default is BOTH.
cache	Specifies the type of cache storage in which RABNs for the file will be cached. Valid values are: ■ DATASPACE (or D) ■ EXTENDED (or E) ■ G64 (or G)

Value	Description
	■ HIPERSPACE (or H) ■ L64 (or L) ■ P64 (or P) ■ VIRTUAL64 (or V) The default is DATASPACE. **Note:** 1. Specify "VIRTUAL64" to request that cache storage be virtual storage above the 2G bar; specify "L64" to request that cache storage be virtual storage above the 2G bar backed by page-fixed 1M large pages; specify "P64" to request that cache storage be virtual storage above the 2G bar backed by pageable 1M large pages; specify "G64" to request that cache storage be virtual storage above the 2G bar backed by page-fixed 2G large pages. 2. If "G64" is specified, we recommend you specify a cache space size (CASSOMAXS/CDATAMAXS) that is a multiple of two gigabytes. The specified size will be rounded up to the nearest multiple of two gigabytes and any difference between the specified size and the allocated size will remain unused. In this case, specifying a multiple of two gigabytes avoids leaving a potentially large amount of storage unused. If "G64" is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

Within the parentheses, the value terms are positional. This means that you can omit a term if you want to accept the default value for it, but you must enter a comma to identify the position if you are specifying a subsequent term within the parentheses.

Multiple CFILE parameters may be specified.

CFILE and CXFILE parameters together are processed in order from left to right (or top to bottom if different ADARUN statements are used). Parameters or statements that specify file ranges are processed before those that specify specific files.

Examples

```
ADARUN CACHE=YES,CFILE=(7,2)
```

File-level caching is active for file number 7. The file is permitted to use up to 75% of the maximum total available cache space (class of service 2). Both Associator and Data Storage RABNS are cached (the default) and the type of cache space used is data space (the default).

```
ADARUN CACHE=YES,CFILE=(3-6,,DATA,HIPERSPACE)
```

File-level caching is active for file numbers 3 through 6. The files are permitted to use up to 50% of the maximum total available cache space (class of service 3, the default). Only Data Storage RABNS are cached, and the type of cache space used is hiperspace.

```
ADARUN CACHE=YES,CFILE=(14-17,,,HIPERSPACE)
```

File-level caching is active for file numbers 14 through 17. The files are permitted to use up to 50% of the maximum total available cache space (class of service 3, the default). Both Associator and Data Storage RABNs are cached (the default), and the type of cache space used is hiperspace.

CLOGBMAX Parameter: Logged Buffer Size Limit

Parameter	Specify . . .	Possible Values	Default
CLOGBMAX	the maximum allowable number of bytes of a logged buffer.	368 - 2147483647	4096

If a buffer is longer than this value, the logged buffer is truncated from the point at which its size exceeds the setting of the CLOGBMAX parameter. The CLOGBMAX setting affects the **ADARUN LOGGING** parameter specifications for CLOGLAYOUT=8. The minimum value (368) is the size of the length of the CLOGLAYOUT=8 basic record plus the length of the extended Adabas control block (ACBX).

Example

The following example runs the Adabas nucleus using a logged buffer size limit of 512. Individual logged buffers for an Adabas command cannot exceed 512 bytes.

```
ADARUN PROG=ADANUC,CLOGB=512
```

CLOGDEV/CLOGSIZE Parameter: Multiple Command Log Device Type / Log Size

Parameter	Specify . . .	Possible Values	Default
CLOGDEV	the device type to be used for multiple command logs.	see text	value of DEVICE parameter
CLOGSIZE	the number of blocks available for each command log.	Min 16, Max 16777215	none

The CLOGDEV and CLOGSIZE parameters specify the type of external storage device used for multiple command logs and the number of blocks available for each command log data set. These parameters must be used *together*; if you specify one parameter, you must also specify the other parameter.

You must also specify the NCLOG parameter to use multiple command logs. If you specify a user exit, it must be UEX12.

The section *Adabas Device Types and Blocks Sizes* in the Adabas Installation documentation specifies the device types that Adabas supports.

All command logging data sets must be formatted using the ADAFRM utility before they are used.

The maximum size of the data set depends on the limits of the operating system and on the requirements of the user.

Example

The following example runs the Adabas nucleus with command logging. It uses six command log data sets and a model 3390 device for command logging. The size of each command log data set is 1500 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3390 device has 15 tracks, and each track can contain 10 blocks of command log data. Thus, 10 cylinders = 10 x 15 x 10 = 1500 blocks).

```
ADARUN
   PROG=ADANUC,LOGGING=YES,NCLOG=6,CLOGDEV=3390,CLOGSIZE=1500
```

CLOGLAYOUT Parameter: Command Logging Format

Parameter	Specify . . .	Possible Values	Default
CLOGLAYOUT	the command log format.	8	8

This parameter specifies the format of the Adabas command log. User programs designed for Adabas 8 or later should specify CLOGLAYOUT=8.

Use the command log format from Adabas 8. Both ACB and ACBX control blocks are converted to a common format. This format includes the corresponding Adabas buffer descriptions (ABDs) as well. Each segmented buffer (format, record, or multifetch) is written separately and uniquely identified.

The format of CLOGLAYOUT=8 records is described in the LORECX DSECT distributed with Adabas 8.

Note: Timestamps in an Adabas 8 command log created using CLOGLAYOUT=8 are stored in machine time (GMT). The LORECX record layout that describes the CLOGLAYOUT=8 command log includes a differential time field that stores the difference between machine time and local time at the time the CLOG record is written. This field allows you to calculate the local time of a command log record.

Example

The following example uses the Adabas 8 format of the command log.

```
ADARUN PROG=ADANUC,CLOGLAYOUT=8
```

CLOGMAX Parameter: Total Logged Buffer Size Limit for a Command

Parameter	Specify . . .	Possible Values	Default
CLOGMAX	the maximum size of all of the logged buffers allowed for an Adabas command.	368 - 2147483647	16384

When the sum of sizes of the logged buffers for an Adabas command reaches the value of the CLOGMAX parameter, the buffer exceeding the limit is truncated and all following buffers are omitted. The size of CLOGMAX must at least be as large as the ADARUN CLOGBMAX parameter setting in order to accommodate the largest buffer of the Adabas command. The minimum value (368) is the size of the length of the CLOGLAYOUT=8 basic record plus the length of the extended Adabas control block (ACBX).

The CLOGMAX setting affects the **ADARUN LOGGING** parameter specifications for the ADARUN CLOGLAYOUT=8 parameter setting.

Example

The following example runs the Adabas nucleus using a command logged buffer size limit of 32K. The sum of all logged buffers for an Adabas command to this nucleus cannot exceed 32K.

```
ADARUN PROG=ADANUC,CLOGMAX=32768
```

CLOGMRG Parameter: Cluster Command Log Merge

Parameter	Specify . . .	Possible Values	Default
CLOGMRG	whether the Adabas cluster is to run with automatic CLOG merge.	YES | NO	NO

An automatic CLOG merge process provides a single cluster-wide command log containing, in chronological order, all Adabas commands executed by any of the cluster nuclei in the time period covered by the log.

CLOGMRG specifies whether the Adabas cluster is to run with the automatic CLOG merge:

Value	Action
YES	enables an automated process that copies dual or multiple command logs to sequential data sets and merges those data sets across a cluster into a single cluster-wide command log. This process automatically merges command logs when an ADARES CLCOPY is submitted from user exit 2/12 and executed. The cluster nuclei and the ADARES utility perform coordinated CLOG switches and CLCOPY functions with merging of CLOG records, much the same way the PLOG merge process works. For the automated CLOG MERGE to synchronize correctly, ADARES must be invoked from the exit.
NO	(the default), each nucleus produces its own CLOG, independent of any other nucleus in the cluster; dual or multiple CLOGs can be manually copied using ADARES CLCOPY to a sequential data set; and the resulting sequential CLOGs can be manually merged across a cluster using the ADARES MERGE CLOG function.

Note: If 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 a CLOGMRG=YES environment.

CLOGMRG is a global (that is, a cluster-wide) parameter, which means that the setting of this parameter in the first cluster nucleus to become active is propagated to all nuclei that subsequently become active.

Parameter Dependencies

It is possible to specify both LOGGING=NO and CLOGMRG=YES. In a cluster environment, it is also possible that a CLOG will be written to even if LOGGING=NO. As long as CLOGMRG=NO, nothing will be written to the CLOG if LOGGING=NO. However, if LOGGING=NO and CLOG-MRG=YES, control records necessary for ADARES CLCOPY will be written to the CLOG.

CLOGMRG is effective only in an Adabas cluster environment; that is, when CLUSTER is specified with a value other than NO and NUCID is specified with a nonzero value.

Dynamic Modification

The setting of the CLOGMRG parameter can be changed dynamically using the CLOGMRG command from the operator console, the ADADBS OPERCOM CLOGMRG function, or the Modify Parameter function of Adabas Online System.

Because CLOGMRG is a global parameter, the change request to one nucleus is automatically propagated to all nuclei in the cluster.

CLUCACHEEXTRA Parameter: Adjust Shared Cache Vector Size

Parameter	Specify . . .	Possible Values	Default
CLUCACHEEXTRA	the additional vector size, in blocks, required to allow Adabas Cluster Services and Adabas Parallel Services to track the blocks held by Adabas Caching Facility in the different cluster nuclei.	0 - 2147483647	1000

This parameter is local and unchangeable. It has effect only if both a cluster product (Adabas Cluster Services or Adabas Parallel Services) and Adabas Caching Facility are active.

If Adabas Cluster Services or Adabas Parallel Services are running on the same system as the Adabas Caching Facility, use the CLUCACHEEXTRA parameter to adjust the vector size of the shared cache. This ensures that Adabas Cluster Services and Adabas Parallel Services can track all of the blocks held privately by the Adabas Caching Facility.

Set the value to the maximum number of blocks expected to be cached by the Adabas Caching Facility. This number can be derived from the CASSOMAXS, CDATAMAXS, and CMAXCSPS parameter settings of the Adabas Caching Facility and the ASSO and DATA block sizes of the database.

CLUCACHENAME Parameter: Cluster Cache Structure/Area Name

Parameter	Specify . . .	Possible Values	Default
CLUCACHENAME	the name of the cache structure/area of the Adabas cluster in which the current nucleus participates.	text	none

For Adabas Cluster Services, the CLUCACHENAME identifies the particular sysplex cache structure (defined in the CFRM policy) associated with a particular Adabas sysplex nucleus cluster. The same name must be set in each nucleus in the cluster. CLUCACHENAME must be unique for each cluster; that is, each database.

Adabas Parallel Services automatically creates a default name for each global cache area allocated by ADACOM. However, if you wish to control the name of the global cache area, you can set this parameter.

Note: we recommend that you allow the system to build the default name.

For Adabas Parallel Services, the CLUCACHENAME identifies a particular global cache area associated with all cluster nuclei that have the same SVC/DBID set as the current nucleus. The same name must be set in each cluster nucleus that has the same SVC/DBID set. CLUCACHENAME must be unique for each SVC/DBID set.

In all cases, the name can be 1-16 characters long. Valid characters are numeric, uppercase alphabetic, the special characters $, @, and #. The name must begin with an alphabetic character (A-Z). IBM names begin with SYS, UNDESIG, or the characters A through I (these characters are not reserved).

Note: Additionally, the special character underscore (_) is allowed in Adabas Cluster Services; that is, sysplex environments.

Example

In the following example, the sysplex cache structure name is defined as "SAG1SMP#CACHE1".

```
ADARUN PROG=ADANUC, CLUCACHENAME=SAG1SMP#CACHE1
```

CLUCACHESIZE Parameter: Size of the Global Cache Area

Parameter	Specify . . .	Minimum	Maximum	Default
CLUCACHESIZE	the amount of storage to allocate for the global cache area that services the Adabas Parallel Services cluster in which the current nucleus participates.	128K	depends on the type of cache (CLUCACHETYPE parameter)	none

The CLUCACHESIZE parameter specifies the amount of storage to be allocated for the global cache area which all nuclei in the Adabas Parallel Services cluster use to share Associator and Data blocks from the database. It has an effect only if CLUSTER=LOCAL is also specified.

The location and maximum size of the global cache area depends on the CLUCACHETYPE parameter. If CLUCACHETYPE=DSP, the global cache is kept in a dataspace shared between the Adabas Parallel Services nuclei. Otherwise, the cache is kept in shared 64-bit virtual storage.

CLUCACHESIZE must be specified for Adabas Parallel Services nuclei; there is no default.

Each Adabas Parallel Services nucleus has its own buffer pool sized by the ADARUN parameter LBP. Optionally, it may also have one or more local cache spaces administered by the Adabas Caching Facility (refer to the description of the ADARUN parameter CACHE). In addition, Adabas Parallel Services uses a global cache area to ensure that current images of Associator and Data blocks from the database are available to all Adabas Parallel Services cluster nuclei.

The global cache area must be large enough to retain:

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

The ADARUN parameters DIRRATIO **and** ELEMENTRATIO determine the ratio between the number of directory and data elements.

CLUCACHESIZE may be specified in bytes, in kilobytes followed by a "K", in megabytes followed by an "M", or in gigabytes followed by a "G":

- the minimum size is 131072 bytes (or 128K);
- for CLUCACHETYPE=DSP, the maximum size is 2147483147 bytes (or 2097152K; or 2048M; or 2G); and
- the maximum allowed numeral is 2147483147; for CLUCACHETYPE=V64, use the K, M or G suffix (for example, 10G or 10240M or 10485760K for a global cache of 10 gigabytes size).

The value is rounded up to the nearest four-kilobyte boundary. If CLUCACHETYPE=L64, the value is rounded up to the nearest one-megabyte boundary. If CLUCACHETYPE=G64, the value is rounded up to the nearest two-gigabyte boundary.

Note: If "G64" is specified for a cache space, we recommend that you specify a CLUCACHESIZE value that is a multiple of two gigabytes. The specified size will be rounded up to the nearest multiple of two gigabytes and any difference between the specified size and the allocated size will remain unused. In this case, specifying a multiple of two gigabytes avoids leaving a potentially large amount of storage unused. If "G64" is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

Appropriate sizes for the global cache area depend very much on the application load in the cluster. For more details about these calculations, read *Global Cache Area Size* in the *Adabas Parallel Services Operations Guide*. You can use the Adabas Online System cache structure size calculator described in *Estimating Sizes for the Cache Structure in a Cluster Environment* in the *Adabas Parallel Services Operations Guide* to determine appropriate global cache area sizes.

Example

In the following example, the global cache area allocated is 500 M.

```
ADARUN PROG=ADANUC,CLUCACHESIZE=500M
```

CLUCACHETYPE Parameter: Global Cache Area Storage Type Option

Parameter	Specify . . .	Valid Values	Default
CLUCACHETYPE	the virtual storage type for the global cache area	DSP | G64 | L64 | P64 | V64	DSP

The CLUCACHETYPE parameter specifies the type of virtual storage to be used for the global cache of an Adabas Parallel Services cluster. It has an effect only if CLUSTER=LOCAL is also specified. Valid values are "DSP", "V64", "G64", "L64", and "P64".

The size of the global cache is specified in the CLUCACHESIZE **parameter**.

CLUCACHETYPE Value	Description
DSP	This is the default value. It indicates that a shared dataspace will be used for the global cache. In this case, the maximum cache size is two gigabytes (2G).
G64	If "G64" is specified, the global cache space shared by the cluster nuclei will reside in 64-bit virtual storage that is backed by page-fixed 2G large pages (provided the large page pool has been configured to a sufficient size and is available in the system). If the system cannot satisfy the allocation request with storage backed by 2G large pages, storage backed by pageable 4K pages will be allocated instead. **Note:** Allocation requests for storage to be backed by 2G large pages will be rounded up to the nearest multiple of two gigabytes. Any difference between the specified size and the allocated size will remain unused. To avoid leaving a potentially large amount of allocated storage unused when specifying CLUCACHETYPE=G64, we recommend that you supply a multiple of two gigabytes for the `CLUCACHESIZE` **parameter**. If "G64" is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.
L64	If "L64" is specified, the global cache space shared by the cluster nuclei will reside in 64-bit virtual storage that is backed by page-fixed one-megabyte (1M) large pages (provided the large page pool has been configured to a sufficient size and is available in the system). If the system cannot satisfy the allocation request with storage backed by 1M large pages, storage backed by pageable four-kilobyte (4K) pages will be allocated instead.
P64	If "P64" is specified, the global cache space shared by the cluster nuclei will reside in 64-bit virtual storage that is backed by pageable one-megabyte (1M) large pages. If the system cannot satisfy the allocation request with storage backed by pageable 1M large pages, storage backed by pageable four-kilobyte (4K) pages will be allocated instead.
V64	If "V64" is specified, a storage area in 64-bit virtual storage will be used for the global cache. Users should note that the cache can be made much larger than 2G, but is still subject to the MEMLIMIT value in effect for the ADACOM job for the Adabas Parallel Services database.

Notes:

1. If one ADACOM job is used to hold the shared cache and lock spaces for multiple Adabas Parallel Services clusters, L64 and G4 cache spaces of different clusters are slightly less isolated from one another than DSP or V64 cache spaces, creating a marginally higher risk that a cache-related error in one cluster might induce an error in a different cluster. This very small risk can be avoided altogether by using different ADACOM jobs for different Adabas Parallel Services clusters that use L64 or G64 cache spaces, but this is not mandatory.
2. Virtual 64-bit storage backed by page-fixed 1M large pages can only be used on mainframes with IBM large page support enabled. You must also allocate the size of the large page pool (use the LFAREA parameter in the IEASYS*xx* member of SYS1.PARMLIB). The LFAREA parameter allows you to specify the amount of real storage to be used for page-fixed 1M large pages;

this parameter cannot be changed dynamically and, if it is not set, page frames will remain allocated at a size of 4K.

3. Virtual 64-bit storage backed by page-fixed 2G large pages can only be used on mainframes with 2G large page support enabled. You must also allocate the size of the large page pool (use the LFAREA parameter in the IEASYS*xx* member of SYS1.PARMLIB). The LFAREA parameter is extended to allow you to specify the amount of real storage to be used for page-fixed 1M and 2G large pages; this parameter cannot be changed dynamically and, if it is not set, page frames will remain allocated at a size of 4K.

Example

In the following example, the global cache area will be allocated in shared 64-bit virtual storage:

```
ADARUN PROG=ADANUC,CLUCACHETYPE=V64
```

CLUCACHEUNCHANGED Parameter: Global Cache Unchanged Block Option

Parameter	Specify . . .	Possible Values	Default
CLUCACHEUNCHANGED	whether unchanged blocks are written to global cache during Adabas Parallel Services processing.	YES | NO	NO

The CLUCACHEUNCHANGED parameter specifies whether an Adabas Parallel Services nucleus should write unchanged blocks to the global cache. It has an effect only if CLUSTER=LOCAL is also specified. Valid values are "YES" and "NO".

- The default value is "NO", which indicates that the nucleus will write only changed (updated) Associator and Data blocks from the database to the global cache.
- If "YES" is specified, the nucleus will write changed and unchanged Associator and Data blocks to the global cache.

Recommendation

If a lot of virtual storage can be made available for local and global caching by the Adabas Parallel Services cluster (where local caching is controlled by the ADARUN LBP parameter and possibly the Adabas Caching Facility add-on product), assigning the bulk of the available storage to the global cache (ADARUN CLUCACHESIZE parameter), which is shared by all Adabas Parallel Services nuclei for the database, provides for more economic use of the storage than assigning it to the local caches of the individual cluster nuclei (ADARUN LBP parameter and the Adabas Caching Facility ADARUN CASSOMAXS and CDATAMAXS parameters), which are private to each nucleus. However, each cluster nucleus requires sufficient local buffer pool space (LBP parameter) to hold the entire working set of Associator and Data blocks needed for parallel command processing.

In cases where the global cache size (CLUCACHESIZE parameter) is set to a large value, CLUCACHEUNCHANGED should be set to "YES", so that the global cache achieves a better hit rate (efficiency).

Example

In the following example, the CLUCACHEUNCHANGED parameter is set to that both changed and unchanged Associator and Data blocks are written to the global cache.

```
ADARUN PROG=ADANUC, CLUCACHEUNCHANGED=YES
```

CLUGROUPNAME Parameter: Cluster Group Name

Parameter	Specify . . .	Possible Values	Default
CLUGROUPNAME	the name of the XCF messaging group in Adabas Cluster Services.	text	none
	the name of the messaging group in Adabas Parallel Services.	text	the router name

The CLUGROUPNAME parameter need not be specified for Adabas Parallel Services except when Adabas Parallel Services is used to recover an Adabas Cluster Services database after a failure that resulted in a remnant DIB entry. In this case, the CLUGROUPNAME parameter for Adabas Parallel Services must be set to the same value as it was for the failed Adabas Cluster Services nuclei.

For Adabas Cluster Services, the CLUGROUPNAME parameter provides the XCF group name to be used by the cluster nuclei for their communication using XCF.

All nuclei in the same cluster must have the same CLUGROUPNAME. Nuclei belonging to different clusters must have different CLUGROUPNAMEs.

In addition, the CLUGROUPNAME must be different from the XCF group names used by other subsystems that communicate using XCF.

The name can be 1-8 characters long. Valid characters are numeric, uppercase alphabetic, and the special characters $, @, and #. The name must begin with an alphabetic character (A-Z). IBM names begin with SYS, UNDESIG, or the characters A through I (these characters are not reserved).

Example

The following examples specifies the name of the XCF messaging group as "SAG1CLU".

```
ADARUN PROG=ADANUC,CLUGROUPNAME=SAG1CLU
```

CLULOCKNAME Parameter: Cluster Lock Structure/Area Name

Parameter	Specify . . .	Possible Values	Default
CLULOCKNAME	the name of the lock structure defined in the coupling facility's CFRM policy or the global lock area in which the current nucleus participates	text	none

For Adabas Cluster Services, CLULOCKNAM identifies a particular sysplex lock structure (defined in the CFRM policy) associated with a particular Adabas nucleus cluster. The same name must be set in each nucleus in the cluster. CLULOCKNAME must be unique for each cluster; that is, each database.

Adabas Parallel Services automatically creates a default name for each global lock area allocated by ADACOM. However, if you wish to control the name of the global lock area, you can set this parameter.

Note: we recommend that you allow the system to build the default name.

For Adabas Parallel Services, the CLULOCKNAME identifies a particular global lock area associated with all Adabas Parallel Services cluster nuclei that have the same SVC/DBID set as the current nucleus. The same name must be set in each cluster nucleus that has the same SVC/DBID set. CLULOCKNAME must be unique for each SVC/DBID set.

In all cases, the name can be 1-16 characters long. Valid characters are numeric, uppercase alphabetic, the special characters $, @, and #. The name must begin with an alphabetic character (A-Z). IBM names begin with SYS, UNDESIG, or the characters A through I (these characters are not reserved).

Note: Additionally, the special character underscore (_) is allowed in Adabas Cluster Services; that is, sysplex environments.

Example

The following example set the name of the lock structure or global lock area as "SAG1SMP#LOCK1".

```
ADARUN PROG=ADANUC,CLULOCKNAME=SAG1SMP#LOCK1
```

CLULOCKSIZE Parameter: Size of the Global Lock Area

Parameter	Specify . . .	Minimum	Maximum	Default
CLULOCKSIZE	the amount of storage allocated for the global lock area	128K	2G	none

The CLULOCKSIZE parameter specifies the amount of storage to be allocated for the global lock area which all nuclei in the Adabas Parallel Services cluster use to synchronize their processing. It has an effect only if CLUSTER=LOCAL is also specified.

CLULOCKSIZE must be specified for Adabas Parallel Services nuclei; there is no default.

CLULOCKSIZE may be specified in bytes, in kilobytes followed by a "K", in megabytes followed by an "M", or in gigabytes followed by a "G".

- The minimum size is 131072 bytes (or 128K)
- The maximum size is 2147483147 bytes (or 2097152K, 2048M, or 2G).

The value is rounded up to the nearest 4-kilobyte boundary.

Appropriate sizes for the global lock area depend on the expected amount of parallel command processing in the cluster. For more details about these calculations, read *Global Cache Area Size* in the *Adabas Parallel Services Operations Guide*. You can use the Adabas Online System lock structure size calculator described in *Estimating Sizes for the Lock Structure in a Cluster Environment* in the *Adabas Parallel Services Operations Guide* to determine appropriate global lock area sizes.

CLUPUBLPROT Parameter: Publishing of Protection Data

Parameter	Specify . . .	Possible Values	Default
CLUPUBLPROT	whether or not cluster nuclei make their protection data available within the cluster before they make the related updates available to the other nuclei.	YES | NO	NO

The optional CLUPUBLPROT parameter specifies whether or not cluster nuclei make their protection data available within the cluster before they make the related updates available to the other nuclei. Valid values are YES and NO. The default value is NO. CLUPUBLPROT is a global, modifiable parameter - that is, all active nuclei in the cluster run with the same setting and this setting can be changed at runtime. When a starting nucleus joins a cluster, it takes over the current setting of the nuclei that are already active.

If CLUPUBLPROT is specified as YES, each nucleus in the cluster writes updated ASSO and DATA blocks to the global cache only after it has first written the protection data for the latest updates

to these blocks either to its WORK dataset or to the global cache. With this setting, when one nucleus in the cluster performs a buffer flush, the other nuclei need not be able to participate in the execution of the flush - the nuclei are more independent of one another.

If CLUPUBLPROT is not specified or is specified as NO, the nuclei in the cluster do not write protection data to the global cache. They continue to write protection data to their respective WORK datasets when this is necessary to provide for the recovery of the database after any failure of a nucleus. With this setting, when one nucleus in the cluster performs a buffer flush, all other nuclei must also take a few actions required for the execution of the flush (such as writing out their latest protection data to their WORK datasets).

Setting CLUPUBLPROT=YES is performance-sensitive for update-intensive workloads. The related `CLUWORK1CACHE` parameter should be configured suitably for all nuclei in the cluster before CLUPUBLPROT is set to YES before or during a nucleus session.

CLUSTER Parameter: Adabas Nucleus Cluster Session Control

Parameter	Specify . . .	Possible Values	Default
`CLUSTER`	whether the Adabas nucleus will participate in a cluster of nuclei working on the same Adabas database, and, if so, the type of cluster (Adabas Parallel Services or Adabas Cluster Services) in which it will participate.	NO | SYSPLEX | LOCAL	NO

The `CLUSTER` parameter specifies whether the Adabas nucleus will participate in a cluster of nuclei working on the same Adabas database, and if so, which type of cluster:

- If CLUSTER=NO (the default), the nucleus will not participate in an Adabas cluster. If CLUSTER=NO, the `NUCID` parameter must be set (or default) to 0.
- If CLUSTER=LOCAL, the nucleus will participate in a cluster where all nuclei run on the same operating system image. The add-on product Adabas Parallel Services is required to run in this mode.
- If CLUSTER=SYSPLEX, the nucleus will participate in a cluster where the nuclei may run on different systems within a parallel sysplex. The add-on product Adabas Cluster Services is required to run in this mode.

If CLUSTER=LOCAL or SYSPLEX, the `NUCID` parameter must be set to a nonzero value that identifies the individual cluster nucleus. Also, the parameter settings MODE=SINGLE, READONLY=YES, LFIOP=0 and DTP=TM, and the use of sequential protection log data sets (DDname DDSIBA) are not allowed for cluster nuclei.

Example

The following example includes the nucleus in a cluster run with Adabas Cluster Services:

```
ADARUN DBID=53,CLUSTER=SYSPLEX,NUCID=5301
```

CLUWORK1CACHE Parameter: Number of WORK Blocks in Global Cache

Parameter	Specify . . .	Possible Values	Default
CLUWORK1CACHE	the maximum number of WORK blocks that the cluster nucleus should keep in the global cache at a time. If the specified value for CLUWORK1CACHE is larger than NWORK1BUFFERS+1 or LP / 20, the smaller of these two values is taken as the effective CLUWORK1CACHE value. If the effective CLUWORK1CACHE value is equivalent to more than 1% of the total size of the data elements in the global cache, it is reduced to 1% of that size, in order to limit the portion of the cache that may be used for protection data.	0 - 32767	0

The optional CLUWORK1CACHE parameter specifies how a cluster nucleus should make its latest protection data available within the cluster, if the CLUPUBLPROT parameter is set to YES. Valid values are integers in the range 0 through 32,767. The default value is 0. CLUWORK1CACHE is a local, unmodifiable parameter — that is, different nuclei in the same cluster may run with different CLUWORK1CACHE values and the value for a nucleus cannot be changed while that nucleus is running.

If the specified value for CLUWORK1CACHE is larger than NWORK1BUFFERS+1 or LP / 20, the smaller of these two values is taken as the effective CLUWORK1CACHE value. If the effective CLUWORK1CACHE value is equivalent to more than 1% of the total size of the data elements in the global cache, it is reduced to 1% of that size, in order to limit the portion of the cache that may be used for protection data. Any reductions of the CLUWORK1CACHE value to these boundaries are reported in the ADAX5E and ADAX5F messages.

CLUWORK1CACHE specifies the maximum number of WORK blocks that the cluster nucleus should keep in the global cache at a time. It also defines the size of an extension of the area on the WORK dataset where the nucleus stores protection data. Another nucleus in the cluster may copy protection data to this extension area from the global cache if this nucleus is slow or unable to participate in a buffer flush.

Apart from defining the size of the protection data extension area on WORK, the CLU-WORK1CACHE parameter has an effect only when the related CLUPUBLPROT parameter is set to YES.

If CLUWORK1CACHE is specified as or defaults to zero and CLUPUBLPROT=YES, the nucleus always writes protection data directly to its WORK dataset before it writes updated ASSO and

DATA blocks to the global cache, if the protection data relates to updates to those blocks. The nucleus does not write protection data to the global cache.

If CLUWORK1CACHE is specified greater than zero and CLUPUBLPROT=YES, the nucleus writes protection data to the global cache that is not yet present on WORK and relates to updates to be written to the cache. At most the number of WORK blocks specified by CLUWORK1CACHE will be stored in the cache. If further protection data needs to be made available within the cluster before updates can be written to the cache, older protection data is written to WORK first to make space for new WORK blocks in the cache.

When CLUPUBLPROT=YES, the setting of the CLUWORK1CACHE parameter is performance-sensitive for update-intensive workloads. Setting CLUWORK1CACHE=0 may lead to a significant number of additional WORK writes. Setting the parameter to a large value may lead to a significant number of additional cache writes. Any number in-between may lead to some additional WORK writes and some additional cache writes. Trials with different settings may be necessary to find the right balance that is best for the overall performance of the cluster for a given workload.

CMAXCSPS Parameter: Adabas Caching Facility Storage Area Quantities

The `CMAXCSPS` (or `CM`) parameter specifies the number of storage areas that can be allocated for an ADACSH cache space in data spaces, hiperspaces, or memory objects (64-bit virtual storage). It has no effect on a cache space in extended memory, where only one Associator and one Data Storage area are allowed.

In conjunction with the `CASSOMAXS` and `CDATAMAXS` parameters, this parameter is used to control both the total amount of space that can be allocated for a cache space and the granularity with which it is allocated.

Value	Description
n	The number of storage areas that can be allocated for an ADACSH cache space in data spaces, hiperspaces, or memory objects (64-bit virtual storage). It has no effect on a cache space in extended memory, where only one ASSO and one DATA storage area are allowed. The minimum value permitted is 1; the maximum is 16, the default is 8. This means that a potential maximum of 34 storage areas can be allocated for each type of cache space used: 16 for Associator RABNs, 16 for Data Storage RABNs, and 1 each for Work parts 2 and 3. **Note:** Increasing the value of `CMAXCSPS` can increase the overall CPU load used by ADACSH as more storage areas must be searched when allocating storage. This value must be carefully selected in association with the `CASSOMAXS` and `CDATAMAXS` values to obtain the correct mix of granularity and CPU overhead. Experience indicates that the default `CMAXCSPS` setting (8) usually provides an appropriate mix.

Example

```
ADARUN CMAXCSPS=6
```

Up to six Associator cache space areas of size CASSOMAXS and up to six Data Storage cache space areas of size CDATAMAXS can be allocated for a particular type of cache space.

CRETRY Parameter: Adabas Caching Facility Retry Time

The CRETRY (or CR) parameter specifies the amount of time (in seconds) between attempts to acquire an area of cache space if the previous GETMAIN or space allocation failed.

CRETRY = *n*

Value	Description
n	The number of seconds between attempts to acquire an area of cache space if the previous GETMAIN or space allocation failed. The minimum number of seconds is 60; the maximum is 2147483647. The default number of seconds is 900 (15 minutes).

If an area of cache space cannot be allocated, ADACSH waits for the specified number of seconds before attempting to allocate the area of cache space again. During this period of time, caching continues in the other areas of the cache space already allocated, but the maximum allowed cache space is not available. If, after the time period has elapsed, there is a request to cache a RABN and no space is available in the current allocation, ADACSH attempts to allocate the area of cache space again. These attempts continue until the cache space is successfully allocated.

Note: CRETRY does not apply to an area of cache space that was released due to a timeout (see CCTIMEOUT).

Example

```
ADARUN CRETRY=500
```

If an attempt to allocate an additional area of a cache space fails, another attempt to cache a RABN in the cache space in question is made after 500 seconds. If the currently allocated space is still full, the attempt is again made to allocate an additional area of cache space.

CSTORAGE Parameter: Adabas Caching Facility Override Cache Type

The CSTORAGE parameter can be used to indicate that all RABNs for the database are to be cached using the system defaults and the type of cache specified by this parameter.

The CSTORAGE (or CS) parameter overrides any CASSOxxx or CDATA*xxx* parameter that may be specified. If the CSTORAGE parameter is specified, all RABNs for the database are cached using the system defaults and the type of cache space specified by CSTORAGE.

Note: This parameter is mutually exclusive with the CFILE and CXFILE parameters.

```
CSTORAGE={DATASPACE | EXTENDED | HIPERSPACE | G64 | L64 | P64 | VIRTUAL64}
```

Value	Description
DATASPACE	Data space
EXTENDED	Extended memory
HIPERSPACE	Hiperspace
G64	Virtual 64-bit storage backed by 2G large pages **Note:** If "G64" is specified, we recommend you specify a cache space size (CASSOMAXS/CDATAMAXS) that is a multiple of two gigabytes. The specified size will be rounded up to the nearest multiple of two gigabytes and any difference between the specified size and the allocated size will remain unused. In this case, specifying a multiple of two gigabytes avoids leaving a potentially large amount of storage unused. If "G64" is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.
L64	Virtual 64-bit storage backed by fixed 1M large pages
P64	Virtual 64-bit storage backed by pageable 1M large pages
VIRTUAL64	Virtual 64-bit storage

Example

```
ADARUN CASSOHSP=1-1795
ADARUN CDATAHSP=81-135,1140-1298,136-170,181-230
ADARUN CSTORAGE=EXTENDED
```

ADACSH caches all Associator and Data Storage RABNs in extended memory (above the 16 megabyte line). All CASSOHSP and CDATAHSP RABN ranges are ignored.

CT Parameter: Command Timeout Limit

Parameter	Specify . . .	Minimum	Maximum	Default
CT	the maximum time (seconds) for interregion communication of results from Adabas to the user.	1	2147483647	60

For Adabas Review, this is the maximum number of seconds (more precisely, units of 1.048576 seconds) that can elapse from the time an Adabas Review hub command has been completed until the results are returned to the user through the interregion communication (operating-system-dependent).

This parameter is used to prevent a command queue element and attached buffer from being held for a long period of time for a user who has terminated abnormally.

Possible causes of a command timeout are

- user region is swapped out or cannot be dispatched;
- user is canceled;
- user has low priority in high activity system.

If the CT limit is exceeded,

- the command queue element and attached buffer are released;
- a message ADAM93 is printed; and
- if the user has not terminated, response code 254 (ADARSP254) is returned to the user program.

Example

The following example permits about 30 seconds to obtain a result through interregion communication from the Adabas Review hub

```
ADARUN PROG=ADAREV,CT=30
```

CWORKSTORAGE Parameter: Adabas Caching Facility Work Area Cache Type

The CWORKSTORAGE (or CWORKS) parameter specifies the type of cache space to use for the CWORK2FAC and/or CWORK3FAC areas.

```
CWORKSTORAGE={DATASPACE | EXTENDED | HIPERSPACE | G64 | L64 | P64 | VIRTUAL64}
```

Value	Description
DATASPACE	Data space
EXTENDED	Extended memory
HIPERSPACE	Hiperspace
G64	Virtual 64-bit storage backed by 2G large pages **Note:** 'If "G64" is specified, the sizes of the cache spaces for Work parts 2 and 3 will be rounded up to the nearest multiple of two gigabytes and any difference between the specified size and the allocated size will remain unused. To avoid leaving a large amount of cache storage unused, we recommend you use "G64" only if the sizes of the Work part 2 and Work part 3 cache spaces (as per the `CWORK2FAC` and `CWORK3FAC` parameters) will not be much smaller than a multiple of two gigabytes. If “G64” is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.
L64	Virtual 64-bit storage backed by fixed 1M large pages
P64	Virtual 64-bit storage backed by pageable 1M large pages
VIRTUAL64	Virtual 64-bit storage

If `CWORKSTORAGE` is not specified, but the `CWORK2FAC` and/or `CWORK3FAC` parameters are specified, then the default `CWORKSTORAGE` is DATASPACE.

Example

```
ADARUN CWORKSTORAGE=HIPERSPACE
```

For Work parts 2 and/or 3, cache space is allocated in hiperspace.

CWORK2FAC Parameter: Adabas Caching Facility Work Part 2 Percentage

The `CWORK2FAC` (or `CWORK2`) parameter specifies the percentage of Work part 2 to be cached.

Value	Description
n	The percentage of Work part 2 to be cached. The minimum and the default is 0; the maximum is 100. If `CWORK2FAC=0` is specified, Work part 2 is not cached.

The number of Work part 2 RABNs to be cached is computed by multiplying the specified percentage factor by the number of RABNs allocated for Work part 2.

The cache space allocation is based on the number of RABNs and the device block size, rounded up to the nearest 4-kilobyte boundary.

```
cache size = int ( ( ( #RABNs x blksize) + 4095) ÷ 4096) x 4096
```

- where

```
#RABNs = int ( ( ( WORK part 2 size x CWORK2FAC ) +50) ÷ 100)
```

The RABN index table is allocated to obtain space for #RABN blocks.

If the specified `CWORK2FAC` cannot accommodate the working set of Work part 2 RABNs, the RABNs beyond the computed `CWORK2FAC` are not cached. Normal I/O operations are performed for the non-cached Work part 2 RABNs.

For any percentage less than 100, the writes to Work are still done; however, RABNs read from Work will be obtained from the cache if the RABN is available.

Example

```
ADARUN CWORK2FAC=50
```

ADACSH caches 50 percent of Work part 2, beginning from the first block. If Work part 2 contains 1000 blocks, a cache space is allocated for 500 blocks. If the Work device type is a 3390, the cache space size required is

```
cache size = int ( ( ( #RABNs x blksize ) + 4095 ) ÷ 4096 ) x 4096
               = int ( ( ( 500 x 5724 ) + 4095 ) ÷ 4096) x 4096
               = 2,863,104
```

The RABN index table is allocated to obtain space for 500 blocks.

CWORK3FAC Parameter: Adabas Caching Facility Work Part 3 Percentage

The `CWORK3FAC` (or `CWORK3`) parameter specifies the percentage of Work part 3 to be cached.

Value	Description
n	The percentage of Work part 3 to be cached. The minimum and the default is 0; the maximum is 100. If `CWORK3FAC=0` is specified, Work part 3 is not cached.

The number of Work part 3 RABNs to be cached is computed by multiplying the specified percentage factor by the number of RABNs allocated for Work part 3.

The cache space allocation is based on the number of RABNs and the device block size, rounded up to the nearest 4-kilobyte boundary.

```
cache size = int ( ( ( #RABNs x blksize ) + 4095 ) ÷ 4096 ) x 4096
```

- where

```
#RABNs = int ( ( ( WORK part 3 size x CWORK3FAC ) + 50 ) ÷ 100)
```

The RABN index table is allocated to obtain space for all blocks of WORK part 3.

If the cache space, specified by `CWORK3FAC` cannot accommodate the working set of Work part 3 RABNs, an aging process similar to ASSO/DATA RABNs is etablished. In case of a "short on storage" situation, the oldest RABN, i.e. the RABN in cache space which was the longest time not accessed, will be reused.

For any percentage less than 100, the writes to Work are still done; however, RABNs read from Work will be obtained from the cache if the RABN is available.

Example

```
ADARUN CWORK3FAC=50
```

ADACSH caches 50 percent of Work part 3. If Work part 3 contains 1000 blocks, a cache space is allocated for 500 blocks. If the Work device type is a 3390, the cache space size required is

```
cache size = int ( ( ( #RABNs x blksize ) + 4095 ) ÷ 4096 ) x 4096
               = int ( ( ( 500  5724 ) + 4095 ) ÷ 4096 ) x 4096
               = 2,863,104
```

The RABN index table is allocated to obtain space for 1000 blocks.

CXFILE Parameter: Adabas Caching Facility Excluded Files

The `CXFILE` parameter is used to exclude files from caching.

Note: `CXFILE` is mutually exclusive with `CASSO`*xxx*, `CDATA`*xxx*, and `CSTORAGE` parameters. If any of these parameters is used with `CXFILE`, they will be ignored.

CXFILE = ({*fnr* | *fnr* - *fnr* } [,, *scope*])

Value	Description
fnr	The file to be excluded specified by file number. The minimum value is 0; the maximum is the actual number of files in the database. File number 0 may be specified to exclude RABNs that are not associated with any given file. Currently, this is only the DSST on the Associator data set.
fnr - fnr	A range of files to be excluded specified by two file numbers (the lower number first) separated by a hyphen with no intervening spaces or special characters.
scope	Specifies the scope of exclusion: ■ ASSO (or A): Only Associator RABNs are to be excluded ■ DATA (or D): Only Data Storage RABNs are to be excluded ■ BOTH (or B): Both Associator and Data Storage RABNs are to be excluded If the scope term is specified, it must be preceded by two commas to maintain its position within the parentheses. See the `CFILE` **parameter** for more information about the positional terms used within the parentheses. The *cls* and *cache* terms are not used for the `CXFILE` parameter but the position of the *scope* term is nevertheless maintained.

Multiple `CXFILE` parameters may be specified.

`CFILE` and `CXFILE` parameters together are processed in order from left to right (or top to bottom if different ADARUN statements are used).

Examples

```
ADARUN CACHE=YES,CFILE=(13-19),CXFILE=(17)
```

Files 13 through 19 are cached using the standard defaults for class of service (3), scope (BOTH), and cache (DATASPACE) with the exception of file 17, which is not cached.

```
ADARUN CACHE=YES,CFILE=(13-19),CXFILE=(17),CXFILE=(16,,DATA)
```

Files 13 through 19 are cached using the standard defaults; however, file 17 is not cached and only the associated Associator RABNs are cached for file 16 since the Data Storage RABNs are excluded.

DATACACHE Parameter: Data Storage Controller Caching

Parameter	Specify . . .	Possible Values	Default
DATACACHE	whether to enable (YES) or disable (NO) controller caching for Data Storage (DATA) data set.	YES | NO	YES

Value	Meaning
YES	The default setting. Data Storage blocks are preread and maintained as long as possible in the caching facilities of the disk controller.
NO	Data Storage blocks are not preread.

Notes:

1. If Data Storage blocks are highly scattered or the update frequency is low, controller caching can cause longer I/O response times.
2. ADARUN caching control parameters also exist for the Associator (ASSOCACHE), the work area (WORKCACHE), the sort area (SORTCACHE), and the temp area (TEMPCACHE).

Example

The following example runs the nucleus without controller caching for the Data Storage data set.

```
ADARUN PROG=ADANUC,DATACACHE=NO
```

DATASPACEWARN Parameter: Data Storage Space Notifications

Parameter	Specify . . .	Possible Values	Default
DATASPACEWARN	Up to three pairs of threshold percentages and percentage change increments, expressed as percentages of the total Data Storage (DATA) portion of the database. When Data Storage space use reaches a specified threshold percentage, a space usage message is issued. In addition, when the Data Storage space use changes by the increment specified for that threshold, additional space usage messages are issued. The full syntax of the DATASPACEWARN parameter is given below.	0-100,1-99	0,5

Note: For the purposes of database space notifications, the space used in the Adabas database comprises all database blocks that have been assigned either to some common purpose (for example, DSST, PPT, DLOG area) or to a file in the database. Only space listed in the free space table (FST) is not calculated as used. In particular, free space allocated for some file but not yet used is calculated as used.

The syntax of the DATASPACEWARN ADARUN parameter is:

DATASPACEWARN=*threshold1***[,***inc1***[,***threshold2***[,***inc2***[,***threshold3***[,***inc3***]]]]]**

The following specifications can be made in the DATASPACEWARN parameter:

Parameter	Description
threshold#	A threshold, expressed as a percentage of the total Data Storage (DATA) portion of the database. When DATA space use exceeds or falls below any of the thresholds set, a space usage message is issued. The default is zero (0). If zero is specified for the first threshold setting, no space messages are issued. When more than one threshold is specified, the second threshold specification must be greater than the first. Likewise, the third threshold specification (if made) must be greater than the second.
inc#	An increment percentage, expressed as a percentage of the total DATA portion of the database. When DATA space use changes (increases or decreases) from the associated threshold by the amount given in this increment percentage, another space usage message is issued. The default is five percent (5).

Up to three threshold-increment pairs can be specified.

Example

In the following example, the ADARUN parameter setting for DATASPACEWARN specifies that a space usage message is printed when the percentage of Data Storage space use is greater than or equal to 50%, 60%, 70%, 80%, 85%,90%, 95%, 96%, 97%, 98%, 99%, and 100% of the total DATA space available. So when DATA space use exceeds 50%, a space usage message is printed. Thereafter, a space usage message is printed for every 10% increase until the DATA space use reaches 80%. Once DATA space use reaches 80%, a space usage message is issued every time DATA space use increases 5% until DATA space use reaches 95%. Once DATA space use reaches 95%, a space usage message is printed every time DATA space use increases 1%.

Likewise, similar space usage messages are produced as DATA space use decreases from the percentages (50%, 80%, and 95%) listed and at the increments listed (10%, 5%, and 1%) in the DATASPACEWARN setting. In this case, space usage messages would be issued when DATA space use drops below 100%, 99%, 98%, 97%, 96%, 95%, 90%, 85%, 80%, 70%, 60%, and 50% of the total DATA space available.

```
ADARUN PROG=ADANUC,DATASPACEWARN=50,10,80,5,95,1
```

DBID Parameter: Database Identification

Parameter	Specify . . .	Minimum	Maximum	Default
DBID	the physical ID of the database.	1	65535	1

Note: If you specify the DBID parameter, you must also specify the ADARUN SVC parameter.

Example

The following example runs the ADAINV utility against physical database 2 using SVC 241.

```
ADARUN PROG=ADAINV,DBID=2,SVC=241
```

DEVICE Parameter: Device Type

Parameter	Specify . . .	Possible Values	Default
DEVICE	the device type on which the first block of the Associator is stored.	see text	3390

The device type of the external storage device on which the first block of the Associator is stored.

This device type is also used as default for most device type parameters.

The Adabas Installation documentation specifies the device types supported by Adabas.

Example

The following example runs the ADALOD utility and stores the first block of the Associator on a 3390 device.

```
ADARUN PROG=ADALOD,DEVICE=3390
```

DIRRATIO / ELEMENTRATIO Parameters: Cluster Cache Directory-to-Data Elements Ratio

The DIRRATIO and ELEMENTRATIO parameters together define the ratio of directory elements to data elements in the Adabas Cluster Services coupling facility cache structure or the Adabas Parallel Services global cache area.

This ratio influences the allocation of the cache structure/area when the first cluster nucleus starts. It has no influence on the cache structure/area when a subsequent nucleus starts.

Parameter	Specify . . .	Minimum	Maximum	Default
DIRRATIO	the directory portion of the directory-to-data element ratio	1	32767	4
ELEMENTRATIO	the element portion of the directory-to-data element ratio	1	32767	1

The DIRRATIO / ELEMENTRATIO ratio is optionally used in Adabas cluster environments to tune the cache structure/area for better performance.

Directory elements are used to keep track of the cluster members that have individual blocks in their local caches (buffer pool and Adabas Caching Facility caches), so that the blocks can be invalidated should other members modify them.

Data elements are used to hold the contents of Associator and Data blocks from the database.

If the number of directory elements is insufficient, existing directory elements are reused and the blocks associated with those directory elements are invalidated and can then no longer be reused. When blocks become invalid even though they have not been modified, they must be reread and registered again the next time they are referenced and validated.

It is generally better to reassign storage for data elements to keep more Associator and Data blocks in the cache structure/area than to define too many directory elements in the cache. More data elements than necessary can be used to keep additional blocks to improve the cache efficiency.

The number of directory elements need not be greater than the sum of the sizes of all buffer pools plus the amount of space represented by data elements (number of data elements times data element size), divided by the smallest Associator block size.

You can use the Adabas Online System cache structure size calculator described in *Estimating Sizes for the Cache Structure in a Cluster Environment* in the *Adabas Parallel Services Operations Guide* or the *Adabas Cluster Services Operations Guide* to determine appropriate global cache area sizes and DIRRATIO/ELEMENTRATIO parameter values.

When connecting to the cache structure/area during startup, the ADAX57 message reports the number of directory and data elements allocated and the size of a data element.

DSF Parameter: Adabas Delta Save Facility Control

Parameter	Specify . . .	Possible Values	Default
DSF	whether to enable support for the (optional) Delta Save Facility.	YES | NO	NO

The DSF parameter controls the Adabas Delta Save Facility for an Adabas nucleus or utility. If Adabas Delta Save is not installed on your system, specifying this parameter may cause an error.

Value	Meaning
YES	Loads certain Delta Save Facility program modules and runs the nucleus or utility (as specified in the ADARUN PROG parameter) in DSF mode. Delta Save functions are enabled: the DLOG area can be installed or removed, Delta Save operations can be performed, full and delta save tape can be merged. DSF=YES is required for Delta Save functions of the nucleus and the ADASAV, ADARES, and ADAULD utilities; it is optional for other functions and utilities. It is required for ADASAV SAVE database (if a DLOG area exists), SAVE DELTA, MERGE, and RESTORE DELTA; for ADARES PLCOPY (only if online saves are performed), for COPY (only for rebuilding the DSIM data set); and for ADAULD UNLOAD SAVETAPE, if the inputs are full and delta save tapes produced by DSF.
NO	(default) Runs the nucleus or utility in non-DSF mode. If DSF=NO is specified or the parameter is omitted, Delta Save functions cannot be performed. Specifying DSF=NO after specifying DSF=YES switches the nucleus from DSF to non-DSF mode. If a DLOG area is defined when the nucleus is started and DSF=NO is specified, the nucleus removes the DLOG area. If a DLOG area is defined when the nucleus is started, the DSF parameter may not be omitted. You can manually remove the DSF logging area from the nucleus using the Remove DSF command through Adabas Online System.

Example

The following example runs the Adabas nucleus in DSF mode; performing Adabas Delta Save functions is possible for this nucleus.

```
ADARUN PROG=ADANUC,DSF=YES
```

DSFEX1 Parameter: Delta Save Facility Exit

Parameter	Specify . . .	Possible Values	Default
DSFEX1	the user routine to be given control by the Delta Save Facility.	see text	none

Used in conjunction with the Adabas Delta Save Facility, the format of the user exit parameter is

```
DSFEX1=exit-name
```

where

exit-name is the name of the user routine that gets control at the DSF user exit; the name can be up to 8 characters long.

Only one program may be specified for the DSF user exit. Only one DSF user exit may be specified. See the Adabas Delta Save Facility documentation for more information.

Example

This example indicates that the Adabas nucleus should use the Delta Save Facility with the user exit routine DSAVEPGM.

```
ADARUN PROG=ADANUC,DSF=YES,DSFEX1=DSAVEPGM
```

DTP Parameter: Distributed Transaction Processing Control

Parameter	Specify . . .	Possible Values	Default
DTP	whether to enable distributed transaction processing and, if so, whether the nucleus is to be a resource or transaction manager.	NO | RM | TM	NO

The DTP parameter controls the distributed transaction processing status of the Adabas nucleus. If Adabas Transaction Manager is not installed on a separate nucleus in your system, specifying this parameter may cause an error.

Value	Meaning
NO	(default) The nucleus does not support distributed transaction processing.
RM	The nucleus is to participate in distributed transaction processing as a resource manager (RM) with other resource managers (databases) coordinated by a transaction manager. This setting is only possible when - MODE=MULTI - READONLY=NO If you have Adabas Transaction Manager installed and if DTP=RM, the LDTP parameter specifies the size (in blocks) of the WORK4 index which is used to administer the data on DDWORKR4.
TM	The nucleus supports distributed transaction processing as a transaction manager (TM). Only one such nucleus is allowed per operating system instance. This setting is only possible when - MODE=MULTI - the nucleus is not part of a cluster (Parallel Services or Cluster Services) - LOCAL=NO - READONLY=NO

Example

The following example indicates that the nucleus will participate as a resource manager in a distributed transaction processing system coordinated by a transaction manager.

```
ADARUN PROG=ADANUC,DTP=RM
```

DUALALD Parameter: Dual Audit Log Device Type

Parameter	Specify . . .	Possible Values	Default
DUALALD	The device type to be used for dual audit logs.	See text	value of DEVICE parameter

The DUALALD parameter specifies the type of external storage device used for dual audit logging.

DUALALD and DUALALS parameters must be used together; if you specify one parameter, you must also specify the other parameter. If you specify a user exit, it must be UEX2.

Refer to the *Adabas for Mainframes* documentation > *Reference* chapter > *DBA Reference Summary* > section *Adabas Device Types and Blocks Sizes* for information about the device types that Adabas supports.

Both dual audit logging data sets must be formatted using the ADAFRM utility before they are used.

The maximum size of the data set depends on the limits of the operating system and on the requirements of the user.

Example

The following example runs the Adabas Audit Server with audit logging and uses a device type 3390 for dual audit logging. The size of each audit log data set is 1500 blocks based on an allocation of 10 cylinders. Each cylinder of a 3390 device has 15 tracks, and each track can contain 10 blocks of audit log data. Thus, 10 cylinders = 10 x 15 x 10 = 1500 blocks.

```
ADARUN PROG=ADANUC,LOGGING=YES,DUALALD=3390,DUALALS=1500
```

DUALALS Parameter: Dual Audit Log Size

Parameter	Specify . . .	Minimum	Maximum	Default
DUALALS	The number of blocks available for each dual audit log.	16	16777215	none

The DUALALS parameter specifies the number of blocks available for each dual audit log data set.

DUALALD and DUALALS parameters must be used together; if you specify one parameter, you must also specify the other parameter. If you specify a user exit, it must be UEX2.

Refer to the *Adabas for Mainframes* documentation > *Reference* chapter > *DBA Reference Summary* > section *Adabas Device Types and Blocks Sizes* for information about the device types that Adabas supports.

Both dual audit logging data sets must be formatted using the ADAFRM utility before they are used.

The maximum size of the data set depends on the limits of the operating system and on the requirements of the user.

Example

The following example runs the Adabas Audit Server with audit logging and uses a device type 3390 for dual audit logging. The size of each audit log data set is 1500 blocks based on an allocation of 10 cylinders. Each cylinder of a 3390 device has 15 tracks, and each track can contain 10 blocks of audit log data. Thus, 10 cylinders = 10 x 15 x 10 = 1500 blocks.

```
ADARUN PROG=ADANUC,LOGGING=YES,DUALALD=3390,DUALALS=1500
```

DUALCLD/DUALCLS Parameter: Dual Command Log Device Type / Log Size

Parameter	Specify . . .	Possible Values	Default
DUALCLD	the device type to be used for dual command logging.	see text	value of DEVICE parameter
DUALCLS	the number of blocks available for each dual command log.	Min 16, Max 16777215	none

The DUALCLD and DUALCLS parameters specify the type of external storage device used for dual command logging and the number of blocks available for each dual command log data set. These parameters must be used *together*; if you specify one parameter, you must also specify the other parameter. If you specify a user exit, it must be UEX2.

The Adabas Installation documentation specifies the device types that Adabas supports.

Both dual command logging data sets must be formatted using the ADAFRM utility before they are used.

The maximum size of the data set depends on the limits of the operating system and on the requirements of the user.

Example

The following example runs the Adabas nucleus with command logging and uses a model 3390 device for dual command logging. The size of each command log data set is 1500 blocks, based

on an allocation of 10 cylinders. (Each cylinder of a 3390 device has 15 tracks, and each track can contain 10 blocks of command log data. Thus, 10 cylinders = 10 x 15 x 10 = 1500 blocks).

```
ADARUN PROG=ADANUC,LOGGING=YES,DUALCLD=3390,DUALCLS=1500
```

DUALPLD/DUALPLS Parameter: Dual Protection Log Device Type / Log Size

Parameter	Specify . . .	Possible Values	Default
DUALPLD	the device type to be used for dual protection logging.	see text	value of DEVICE parameter
DUALPLS	the number of blocks available for each dual protection log.	Min 16, Max 16777215	none

The DUALPLD and DUALPLS parameters specify the type of external storage device used for dual data protection logging and the number of blocks available for each log data set. These parameters must be used *together*; if you specify one parameter, you must also specify the other parameter. If you specify a user exit, it must be UEX2.

When these parameters are specified, dual protection logging is activated for the session. See ***Adabas Restart and Recovery*** for information about dual protection logging.

The Adabas Installation documentation specifies the device types that Adabas supports.

Both dual protection log data sets must be formatted using the ADAFRM utility before they are used. The appropriate JCL for the data sets is required.

Example

The following example runs the Adabas nucleus with dual data protection logging and uses a model 3390 device; the size of each dual log data set is 1350 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3390 device has 15 tracks, and each track contains 9 blocks of protection log data. Thus, 10 cylinders = 10 * 15 * 9 = 1350 blocks).

```
ADARUN PROG=ADANUC,DUALPLD=3390,DUALPLS=1350
```

ENCRYPTION Parameter: Activate Adabas Encryption

Parameter	Specify . . .	Possible Values	Default
ENCRYPTION	Whether or not to activate Adabas Encryption.	YES | NO	NO

The ENCRYPTION parameter controls whether Adabas supports encrypted database container datasets.

If ENCRYPTION=YES is specified, any container dataset of the database (ASSO, DATA, WORK, CLOG, PLOG, RLOG, DSIM, SORT, or TEMP) may be encrypted.

If ENCRYPTION=NO is specified (the default), Adabas does not support encrypted database container datasets. In this case, if a database container dataset is encrypted, Adabas terminates with an error when it attempts to open the dataset.

Note: Adabas Encryption requires its own license module (named AEZLIC) or file (DD-name DDLAEZ), in addition to the standard Adabas license. If the AEZ license is missing or invalid, the Adabas nucleus will refuse to start.

Setting ENCRYPTION=YES is required if one or more of the database container datasets accessed by the Adabas nucleus or utility are encrypted. (YES may also be specified if none of the container datasets is encrypted.)

In a cluster, ENCRYPTION is a local, fixed parameter. Still, if ASSO, DATA, or any of the xLOG datasets are encrypted, ENCRYPTION must be set to YES for all nuclei in the cluster.

EXCPVR Parameter: Control EXCP or EXCPVR Use

Parameter	Specify . . .	Possible Values	Default
EXCPVR	whether EXCP or EXCPVR should be used when running APF-authorized.	YES | NO	YES

The EXCPVR parameter controls whether EXCP or EXCPVR should be used when from an APF-authorized load library. If, EXCPVR=YES is specified, EXCPVR is used; if EXCPVR=NO is specified, EXCP is used.

Note: If EXCPVR=YES is specified when running from a non-APF-authorized load library, this parameter is ignored.

Example

The following example runs Adabas APF-authorized using EXCP.

```
ADARUN PROG=ADANUC,EXCPVR=NO
```

FASTPATH Parameter: Adabas Fastpath Control

Parameter	Specify . . .	Possible Values	Default
FASTPATH	whether to enable Adabas Fastpath support.	YES | NO	NO

The FASTPATH parameter controls the optional Adabas Fastpath product. If Adabas Fastpath is installed on your system, FASTPATH=YES enables the full functionality of the product.

If Adabas Fastpath is not installed on your system, FASTPATH=YES invokes the Adabas Fastpath command analysis sampler AFPLOOK that is provided with Adabas.

The demo version of Adabas Online System that is supplied with Adabas provides access to the Adabas Fastpath online system SYSAFP so that the AFPLOOK sample can be viewed online exactly as it would appear if Adabas Fastpath were installed. For complete information about AFPLOOK, read *AFPLOOK* in the *Adabas DBA Tasks Manual*.

Example

The following example runs Adabas Fastpath with Adabas.

```
ADARUN PROG=ADANUC,FASTPATH=YES
```

FMXIO Parameter: Parallel LFIOP I/O Operations

Parameter	Specify . . .	ASYTVS=	Minimum	Maximum	Default
FMXIO	the maximum number of parallel I/O operations by LFIOP flush processing.	YES	1	16	1
		NO	1	100	60

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.

Parameter Dependencies

The precise meaning of the FMXIO parameter depends on the setting of the **ASYTVS parameter**):

- When ASYTVS=YES (perform buffer flushes 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 the specified number exceeds that, it is automatically reduced to 16.

- When ASYTVS=NO (perform buffer flushes in ascending RABN sequence, without regard to the distribution of the blocks over volumes), FMXIO specifies the number of I/Os to be started in parallel overall. The minimum number is 1; the maximum value is 100; and the default is 60.

Dynamic Modification

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

Specific Product Recommendations

Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

Example

The following example specifies that a maximum of 30 I/O operations for asynchronous buffer flushes can be started in parallel:

```
ADARUN PROG=ADANUC,ASYTVS=NO,FMXIO=30
```

FORCE Parameter: Allow Nucleus Database ID or Review Hub Table Entry Overwrite

Parameter	Specify . . .	Possible Values	Default
FORCE	whether the nucleus or Adabas Review hub can overwrite an existing ID table entry.	YES | NO	NO

If running Adabas Review, this indicates whether the Adabas Review hub can overwrite an existing ID table entry. When a Review hub starts up, ADARUN scans the ID table to ensure that no entry exists for the Review hub. You can use the FORCE parameter to indicate whether the Review hub can overwrite an existing ID table entry.

The ID table entry is derived from the database ID and the job name. For Adabas Review, the ID table entry is derived from the Review hub ID (REVIEW=). The ID table entry is deleted when the nucleus terminates normally.

The FORCE parameter allows the nucleus or Adabas Review hub to overwrite the existing ID table entry and access the database.

Caution: Do not use the FORCE parameter unless absolutely necessary, or the integrity of the database could be lost. Ensure that no nucleus or Review hub is active for the ID table entry being overwritten.

Value	Meaning
YES	The nucleus or Adabas Review hub that is starting can overwrite an existing ID table entry. FORCE=YES is required when restarting a session that terminated abnormally with an ADAM98 message. In this case, the ID table still contains an active entry for the nucleus or Review hub. Overwriting the existing entry by specifying FORCE=YES prevents further communication to the overwritten nucleus or hub and causes loss of cross-memory environment resources, which cannot be restored until the next IPL.
NO	(default) If the ID table contains an entry for the nucleus or Adabas Review hub that is starting, the nucleus is denied access to the database or the Review hub is not permitted to start.

Note: In an Adabas Cluster Services or Adabas Parallel Services environment, the FORCE parameter applies to the NUCID, rather than the database ID, because a cluster nucleus builds an ID table entry for the NUCID.

A data integrity block (DIB) entry will only be removed once the ID Table initialization has been successful. Therefore, you must set IGNDIB and FORCE both to "YES" if either of the following occur:

- You receive a PARM ERROR 26 after parameter settings IGNDIB=NO and FORCE=YES were applied;
- You receive a PARM ERROR 23 after parameter settings IGNDIB=YES and FORCE=NO were applied.

Examples

The following example specifies that if the ID table contains an active entry for DBID 7, overwrite the entry.

```
ADARUN PROG=ADANUC,FORCE=YES,DBID=7
```

The following example specifies that if the ID table contains an active entry for the Adabas Review hub, overwrite the entry.

```
ADARUN PROG=ADAREV,FORCE=YES,REVIEW=202
```

HEXnn Parameter: Hyperdescriptor User Exit

Parameter	Specify . . .	Possible Values	Default
HEX*nn*	the hyperdescriptor exit and the user routine to be used with it.	see text	none

Used in conjunction with the hyperdescriptor facility, the format of the parameter is

```
HEXnn=exit-name
```

where

nn	is the number of the hyperdescriptor exit, a two-digit decimal integer in the range 01-31 inclusive.
exit-name	is the name of the user routine that gets control at the hyperdescriptor exit; the name can be up to 8 characters long.

Only one program may be specified for each hyperdescriptor exit. Up to 31 hyperdescriptor exits may be specified (in any order). See the *Adabas User, Hyperdescriptor, Collation Descriptor, and SMF Exits Manual* for more information.

Example

The following example runs the Adabas nucleus using hyperdescriptor exits 1 and 2. It also gives control to the user routine UHYPER1 at hyperdescriptor exit 1 and gives control to the user routine UHYPER2 at hyperdescriptor exit 2.

```
ADARUN PROG=ADANUC,HEX01=UHYPER1,HEX02=UHYPER2
```

IGNDIB Parameter: Ignore DIB Entry for Adabas Nucleus

Parameter	Specify . . .	Possible Values	Default
IGNDIB	whether to ignore and delete an active nucleus entry in the data integrity block.	YES | NO	NO

The DIB is the data *integrity* block in the Associator. The DIB contains an entry that indicates whether an Adabas session is active for the database. This feature prevents an update nucleus session from being started while another one is running on the same database; it is especially important when several CPUs share the same database DASD devices. The DIB entry is automatically released when the Adabas session is terminated normally.

This parameter indicates whether this update nucleus session should ignore and delete the entry in the data integrity block (DIB) in the Associator.

Value	Meaning
YES	Adabas ignores and deletes the existing DIB entry and initiates a new session.
NO	The default setting. If the DIB entry indicates that an Adabas session is already active for the database, Adabas rejects an attempt to start an update session against the database. This is true even when the session that the DIB indicates is active has terminated abnormally.

Note: we recommend that you use IGNDIB=YES only once and immediately remove it from the JCL. If no DIB entry exists when IGNDIB=YES is specified, the nucleus terminates abnormally.

A data integrity block (DIB) entry will only be removed once the ID Table initialization has been successful. Therefore, you must set IGNDIB and FORCE both to "YES" if either of the following occur:

- You receive a PARM ERROR 26 after parameter settings IGNDIB=NO and FORCE=YES were applied;
- You receive a PARM ERROR 23 after parameter settings IGNDIB=YES and FORCE=NO were applied.

Example

The following example runs an Adabas update session and ignores and deletes the DIB entry for an active Adabas update nucleus.

```
ADARUN PROG=ADANUC,IGNDIB=YES
```

IGNDTP Parameter: Ignore Two-Phase Commit Area (Work Part 4)

Parameter	Specify . . .	Possible Values	Default
IGNDTP	whether to ignore the two-phase commit area (Work part 4).	YES | NO	NO

Caution: This parameter is for emergency use only; contact your Software GmbH technical support representative before using it. Whenever the information on Work part 4 is ignored, the integrity of the incomplete global transactions that are related to that information cannot be guaranteed.

IGNDTP=YES can be specified with DTP=NO or DTP=RM.

When DTP=NO, the nucleus will not start if there is data in Work part 4 related to the recovery of previously incomplete global transactions. In this situation, specifying IGNDTP=YES forces the nucleus to ignore the data in Work part 4 and start anyway.

When DTP=RM, the nucleus also ignores any data on Work part 4 and reinitializes the area. This can be useful during a recovery when the state and correctness of the incomplete global transactions that are stored on Work part 4 has already been established.

Example

The following example runs an Adabas update session without support for distributed transactions; it ignores data in the Work part 4 area left over from the previous session with support for distributed transaction processing.

```
ADARUN PROGRAM=ADANUC,DTP=NO,LDTP=8,IGNDTP=YES
```

INDEXCROSSCHECK Parameter: Data Storage Record Validation

Parameter	Specify . . .	Possible Values	Default
INDEXCROSSCHECK	whether index cross-checking should be turned on or off.	YES | NO	YES

This parameter controls whether index cross-checking is turned on or off. Index cross-checking is a process Adabas uses to ensure that a data storage record it is reading actually matches the descriptor value in the index by which the record was found. When index cross-checking is turned on, Adabas extracts the descriptor value from the record being read and compares it to the value used for positioning in the index. If there is a mismatch, Adabas returns response code 175 (ADARSP175).

If "YES" is specified (the default value), Adabas extracts the descriptor value from the record being read and compares it to the value used for positioning in the index. If the two values are not equal, Adabas checks for a parallel update command on the same record and retries after that update has finished. If there is no parallel update command, Adabas returns response code 175 (ADARSP175) to indicate an inconsistency between the data storage and the index.

If "NO" is specified, Adabas does not extract the descriptor value from the record, but immediately checks for a parallel update command on the same record. If there is one, even if it does not update the descriptor in question, Adabas waits until the update has finished and then retries. With this parameter setting, Adabas does not validate the index against the data storage record and does not return response code 175 (ADARSP175).

Note: In a cluster nucleus (CLUSTER=SYSPLEX or CLUSTER=LOCAL), Adabas sets INDEXCROSSCHECK to "YES", overriding any user-specified INDEXCROSSCHECK value, and validates the descriptor value from the index against the corresponding value from the data storage.

Setting INDEXCROSSCHECK to "NO" may, under certain circumstances, reduce the CPU time required for ensuring that the Data Storage record being read matches the descriptor value by which the record was found. This applies in particular to cases where neither the descriptor itself nor (if it is a special descriptor) its parent field(s) occur in the format buffer specifying the fields to be returned and where the descriptor or one of its parents were defined near the end of the FDT.

Example

The following example specifies that Adabas should not extract the descriptor value from the record being read. The index is not validated against the data storage record.

```
ADARUN PROGRAM=ADANUC,INDEXCROSSCHECK=NO
```

INDEXUPDATE Parameter: Index Update Processing Control

Parameter	Specify . . .	Possible Values	Default
INDEXUPDATE	whether to use the original or advanced index update processing rules.	ORIGINAL | ADVANCED	ORIGINAL

This parameter controls which index structure rules the Adabas nucleus uses to perform index updates. For each file in the database, the file's index stores the descriptor values from the file's data storage records in a tree structure. Adabas can use two very similar, but slightly different sets of rules that govern the tree structure. The advanced rules are less strict than the original ones and allow Adabas to perform certain types of index updates more efficiently than the original rules do.

When INDEXUPDATE is set to "ORIGINAL", Adabas uses the original, strict rules for the structure of the index trees in the database. This is the default value

When INDEXUPDATE is set to "ADVANCED", Adabas uses modified, less strict rules for the index tree structure for certain types of index updates. For these index updates, efficiency gains are expected and less protection data will be written to the WORK dataset.

The INDEXUPDATE setting cannot be changed at runtime. In a cluster, INDEXUPDATE should (but need not) be set to the same value in all nuclei in the cluster. It is possible to set INDEXUPDATE back to "ORIGINAL" after Adabas has been performing updates with INDEXUPDATE=ADVANCED. In this case, the index structure established with the advanced rules will stay in effect and the nucleus will perform further index updates according to the original index structure rules.

Note: The ICHECK functions of the *ADACHK* utility check that the index of each file adheres to the advanced, less strict rules for the index structure.

INFOBUFFERSIZE Parameter: Information Buffer Pool (Adabas Event Log) Size

Parameter	Specify . . .	Possible Values	Default
INFOBUFFERSIZE	the size of the information buffer pool, in bytes. The information buffer pool is also known as the Adabas event log.	0 - 2147483647	0 (no buffer pool is allocated)

The information buffer pool is a wraparound buffer in memory that represents the size of the new Adabas event log. The INFOBUFFERSIZE parameter identifies the size of the Adabas event log. When the Adabas event log fills up, the oldest entries in the log are overwritten. If a non-zero value is specified then the buffer size will be rounded down to the size of a whole number of event

log entries (each entry currently being 136 bytes in length) with a minimum size sufficient to contain 8 entries.

Setting INFOBUFFERSIZE to zero (0) indicates that no buffer pool (Adabas event log) is allocated.

Note: Each response code (RSP145 or RSP200) event is logged to the command log, if CLOGLAYOUT=8 is specified, regardless of the setting of the INFOBUFFERSIZE parameter.

The information in the Adabas event log can be printed using the ADADBS DEVENTLOG utility function. For more information, refer to the Adabas Utilities Manual.

Example

The following example specifies an information buffer pool (Adabas event log) that is 50,000 bytes in size.

```
ADARUN PROGRAM=ADANUC,INFOBUFFERSIZE=50000
```

INTAUTO Parameter: Time Interval Between Autorestart Progress Messages

Parameter	Specify . . .	Minimum	Maximum	Default
INTAUTO	the time interval (in seconds) between autorestart progress messages.	1	2147483647	0

When performing session autorestart to recover from a preceding failure, Adabas prints autorestart progress messages (in message ADAN56) to indicate how much repair and recovery work has already been done and how much remains to be done. By default, these messages occur when each phase of the autorestart (backward repair, forward repair, autobackout) completes.

The INTAUTO parameter allows you to set the time interval (in seconds) at which autorestart progress messages will be printed, in addition to the standard messages printed at the end of each phase (assuming a phase takes longer than the specified time interval). The default (0 seconds) indicates that autorestart progress messages should only be printed at the completion of each autorestart phase.

Example

The following ADARUN INTAUTO parameter indicates that Adabas should print progress messages every ten seconds when performing session autorestart processing.

```
ADARUN PROGRAM=ADANUC,INTAUTO=10
```

INTNAS Parameter: Time Interval Between SYNS 60 Checkpoints

Parameter	Specify . . .	Minimum	Maximum	Default
INTNAS	the time interval between SYNS 60 checkpoints.	1	2147483647	3600

The SYNS 60 statistic checkpoint is generated by the Adabas nucleus and written to the Adabas Checkpoint system file. It may be displayed using the CPLIST function of the ADAREP utility or Adabas Online System. For more information about checkpoints see the ADAREP utility discussion in the Adabas Utilities Manual.

INTNAS allows you to set the time interval (in units of 1.048576 seconds) between SYNS 60 checkpoints. The default (3600) is approximately 1 hour meaning that nucleus statistics will be recorded once each hour.

Example

The following example sets Adabas so that nucleus statistics are recorded every 30 minutes.

```
ADARUN PROGRAM=ADANUC,INTNAS=1800
```

LAP Parameter: Adabas Auditing Pool Size

Parameter	Specify . . .	Possible Values	Default
LAP	The size of the Adabas Auditing pool.	minimum 20000	100000 (if AUDITING=YES) or 0 (if AUDITING=NO)

Notes:

1. The LAP parameter should be specified for an Adabas nucleus running with ADARUN parameter AUDITING=YES. In addition, the Auditing pool will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters AUDITING=YES and V64BIT=YES.
2. The LAP parameter should be specified for an Adabas Audit Server.
3. The LAP parameter values in the Adabas nucleus and the Adabas Audit Server may be different.
4. we recommend setting LAP to a relatively large value (e.g. LAP=40M or greater). To determine what this setting should be for your site, you could run Adabas in a test environment with auditing turned on, simulate the activity in the production environment on this test system, and use the Adabas DRES operator command to view the current and high-water mark of auditing pool usage.
5. If an Adabas nucleus with AUDITING=YES is run with V64BIT=YES and LARGEPAGE=G64, the allocation request for the Auditing pool will be rounded up to the nearest multiple of two gigabytes.

Any difference between the specified size and the allocated size will remain unused. For this reason, we recommend that if `LARGEPAGE=G64`, you specify the LAP value as a multiple of two gigabytes to avoid leaving a potentially large amount of allocated storage unused. If `LARGEPAGE=G64` is specified and 2GB large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

Example

```
ADARUN PROG=ADANUC,AUDITING=YES,LAP=50000K
```

LARGEPAGE Parameter: Large Page Support Option

Parameter	Specify . . .	Possible Values	Default
LARGEPAGE	whether or not the Adabas nucleus should use page-fixed one-megabyte (1M) large pages of real storage above the two-gigabyte (2G) bar (LARGEPAGE=L64), page-fixed 2G large pages above the 2G bar (LARGEPAGE=G64), or pageable one-megabyte (1M) large pages of 64-bit real storage (LARGEPAGE=P64). A value of "YES" is equivalent to specifying a value of *L64*. Large pages can only be used if the system is configured to provide them.	G64 | L64 | P64 | NO | YES	NO

This parameter indicates whether or not the Adabas nucleus should use large pages. The default is NO.

Value	Meaning
G64	The Adabas nucleus will attempt to acquire virtual storage above the 2G bar backed by page-fixed 2G large pages. The virtual storage will be backed by large pages if the operating system supports them and if sufficient large pages are available in the system. If insufficient large pages are available, the virtual storage above the 2G bar will be backed by pageable four-kilobyte (4K) pages. You can specify LARGEPAGE=G64 only if the ADARUN `V64BIT` parameter is set to "YES". The LARGEPAGE=G64 setting does not apply to the buffer pool, flush I/O pool, Work and PLOG I/O buffers, or cast-out directory buffer (in a cluster environment). At this time, it applies only to ADARUN parameter LRPL (in an Adabas replication environment or the Event Replicator Server) and LAP (in an Adabas Auditing environment or the Adabas Audit Server). **Note:** Allocation requests for storage to be backed by 2G large pages are rounded up to the nearest multiple of two gigabytes. Any difference between the specified size and the allocated size will remain unused. For this reason, we recommend that you specify a multiple of two gigabytes for any storage area to which the LARGEPAGE=G64 setting applies, to avoid leaving a potentially large amount of allocated storage unused. If LARGEPAGE=G64 is specified and 2G large pages

Value	Meaning
	are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.
L64 or YES	The Adabas nucleus will attempt to acquire virtual storage above the 2G bar backed by page-fixed 1M large pages. The virtual storage will be backed by large pages if the operating system supports them and if sufficient large pages are available in the system. If insufficient large pages are available, the virtual storage above the 2G bar will be backed by pageable 4K pages. A value of "YES" is equivalent to specifying a value of "L64". You can specify LARGEPAGE=L64 or LARGEPAGE=YES only if the ADARUN `V64BIT` parameter is set to "YES".
P64	Where appropriate, the Adabas nucleus will attempt to acquire virtual storage above or below the 2G bar backed by pageable 1M large pages in 64-bit real storage. The virtual storage will be backed by large pages if the operating system supports them and if sufficient large pages are available in the system. If insufficient large pages are available, the virtual storage above the 2G bar will be backed by pageable 4K pages. LARGEPAGE=P64 can be specified with the ADARUN `V64BIT` parameter set to "NO" or to "YES".
NO	Virtual storage above the 2G bar will be backed by 4K pages. This is the default value.

The amount of storage in your operating system available for page-fixed 1M or 2G large pages is set using the LFAREA parameter in PARMLIB member IEASYS*xx*. For complete information on the LFAREA parameter, refer to the *IBM z/OS Extended Addressability Guide*.

Example

The following example requests that the Adabas nucleus should make use of virtual storage above the 2G bar and should use 1M large pages.

```
ADARUN PROG=ADANUC,LARGEPAGE=YES,V64BIT=YES
```

LBP Parameter: Length of Buffer Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LBP	the size of the Adabas buffer pool in bytes.	80 000	See text	350 000

This parameter controls the size of the buffer pool in an Adabas session. The buffer pool contains the ASSO and DATA blocks that Adabas reads and updates on behalf of the application programs.

Ensure that sufficient memory is available to accommodate the LBP size specified. Adabas allocates the number of bytes specified to the buffer pool. You can specify that number in units of kilobytes, megabytes, or gigabytes by appending the letter K, M, or G, respectively.

A minimum of 80 000 bytes must be available to successfully initiate a session. For performance reasons, we recommend that you specify at least 50 000 bytes per thread (NT parameter). If you are using an asynchronous buffer flush management, that is, the LFIOP parameter specifies a nonzero value, the LBP value must be greater than the value specified by LFIOP.

If you do not have the `V64BIT` parameter specified or you have it set to "NO", the buffer pool space is allocated in 31-bit-addressable memory (above the 16M line) and the maximum number that can be requested is 2,147,483,647 (2^31−1, or roughly 2 gigabytes). If you specify V64BIT=YES, the buffer pool space is allocated in 64-bit-addressable memory (above the 2G bar) and the maximum number that can be requested is 2^64−1 (roughly 16 exabytes).

In either case, the practical maximum depends on what the operating system allows Adabas to obtain based on the `REGION` and `MEMLIMIT` parameters in the JCL. The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session. A related key performance indicator to watch is the buffer efficiency statistic.

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.
- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can process updates from multiple concurrent users efficiently.

Examples

The following example allocates a maximum of 800000 bytes to the Adabas buffer pool for this session.

```
ADARUN PROG=ADANUC,LBP=800000
```

The following example allocates 4 gigabytes to the buffer pool. V64BIT=YES is required for a buffer pool of such a size.

```
ADARUN PROG=ADANUC,LBP=4G,V64BIT=YES
```

LCP Parameter: Length of Security Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LCP	the maximum size of the Adabas security pool.	2000	2147483647	10000

The maximum number of bytes to be used for the Adabas security pool during a session. The pool is used to keep security information in memory in order to optimize performance.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

Example

The following example allocates a maximum of 50000 bytes to the Adabas security pool for this session.

```
ADARUN PROG=ADANUC,LCP=50000
```

LDEUQP Parameter: Length of Unique Descriptor Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LDEUQP	the size of the unique descriptor pool.	5000	2147483647	5000

The unique (UQ) descriptor pool comprises elements that are the length of a unique descriptor value plus 14 bytes. The UQ descriptor pool should be large enough to hold all unique descriptor values that are being changed by *all* transactions at any point in time.

Adabas will make an entry in this pool every time a unique descriptor value is deleted from or inserted into a file's index. A full update of a unique descriptor value produces two entries: one for the deletion of the old value and one for the insertion of the new value. The entry for an insertion is normally removed as soon as the index update has been performed. The entry for a deletion is removed when the current transaction ends. However, Adabas Cluster Services and Adabas Parallel Services nuclei always keep entries for insertions until the end of the transaction too.

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.
- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

For example, assuming that the average length of unique descriptor values is about 16 bytes, the following Adabas session can process a maximum of about 1000 unique descriptors at one time.

```
ADARUN PROG=ADANUC,LDEUQP=30000
```

LDTP Parameter: Length of Distributed Transaction Processing Area (Work Part 4)

Parameter	Specify . . .	Minimum	Maximum	Default
LDTP	the size (in blocks) of the WORK4 index which is used to administer the data on DDWORKR4 if you have Adabas Transaction Manager installed.	8	one eighth of the WORK4 size	8

If you have Adabas Transaction Manager installed, this parameter specifies the size (in blocks) of the WORK4 index which is used to administer the data on DDWORKR4:

- If DTP=RM and LDTP is either unspecified or specified as "0", a default of 8 blocks is used.
- If specified, the ADARUN LDTP must be at least 8 blocks. The maximum size that can be specified is the size of the DDWORK4 data set divided by 8.

DDWORKR4 is a container data set used for the same purpose as Work part 4 of DDWORKR1 was used previously, with the difference that DDWORKR4 can be used in parallel by all members in a cluster. The DDWORKR4 data set should be allocated and formatted in the normal way, using a block size greater than or equal to DDWORKR1. It should be at least as large as the cluster's LP parameter of the database or cluster.

Example

In the following example, the LP parameter specifies 2000 blocks for the data protection area (Work part 1). This example also uses 2500 blocks for the distributed transaction processing area on the Adabas Work data set (part 4).

```
ADARUN PROG=ADANUC,DTP=RM,LP=2000,LDTP=2500
```

LFIOP Parameter: Asynchronous Buffer Pool Control

Parameter	Specify . . .	Minimum	Maximum	Default
LFIOP	whether to enable asynchronous buffer flush operation and set the I/O pool size.	0 | 1 | 80000	18446744073709551615 (although this is really limited by the available amount of virtual and real memory)	0

The asynchronous buffer flush I/O pool enables asynchronous Adabas buffer flush operation and defines the size of the related buffer. Specifying a non-zero value for LFIOP enables asynchronous buffer flush operation; the value specified determines the buffer size, in bytes.

If LFIOP is not specified or zero is specified (the default), synchronous buffer flushing is performed. Note that flushing is always synchronous in single-user mode (MODE=SINGLE).

Value	Meaning
0	No asynchronous buffer flush (the default); flushing is synchronous.
1	Asynchronous buffer flush operations occur. The flush pool is sized at 25% of the buffer pool (LBP=) size.
n (minimum of 80000)	Asynchronous buffer flush operations occur. The flush pool size is defined by the LFIOP setting (in bytes).

When LFIOP=1 or LFIOP=n and the specified or calculated flush pool size is less than 80000 bytes, Adabas uses a size of 80000 bytes.

In environments running in 31-bit addressing mode, the I/O pool is allocated above the 16-megabyte line.

In environments supporting 64-bit virtual storage, the flush I/O pool is allocated above the 2 gigabyte bar when the V64BIT ADARUN parameter is set to "YES".

If you do not have the V64BIT parameter specified or you have it set to "NO", the flush I/O pool space is allocated in 31-bit-addressable memory above the 16M line) and the maximum number that can be requested is 2,147,483,647 (2^31−1, or roughly 2 gigabytes). If you specify V64BIT=YES, the flush I/O pool space is allocated in 64-bit-addressable memory (above the 2G bar) and the maximum number that can be requested is 2^64−1 (roughly 16 exabytes).

In either case, the practical maximum depends on what the operating system allows Adabas to obtain based on the REGION and MEMLIMIT parameters in the JCL. The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session. Related key performance indicators to watch are the buffer flush statistics and the write I/O statistics.

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server causing replication data to be written and read from the SLOG file.
- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1") consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example enables asynchronous buffer flush operation for this session and allocates an I/O pool of 200000 bytes (25% of the LBP size).

```
ADARUN PROG=ADANUC,LFIOP=1,LBP=800000
```

LFP Parameter: Length of Internal Format Buffer Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LFP	the size of the Adabas internal format buffer pool.	6000	2147483647	12 000

This pool stores internal format buffers used with Adabas read and update commands as described in the *Adabas Command Reference Guide*.

Each internal format buffer pool entry requires approximately 60 + (64 * *n*) bytes, where *n* equals the number of fields referenced in the command.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

In environments running in 31-bit addressing mode, the LFP pool is allocated above the 16-megabyte line.

Specific Product Recommendations

If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example allocates 8000 bytes for the Adabas internal format buffer pool for this session.

```
ADARUN PROG=ADANUC,LFP=8000
```

LI Parameter: Length of Table of ISNs (TBI)

Parameter	Specify . . .	Minimum	Maximum	Default
LI	the size of the TBI.	2000	2147483647	10000

LI specifies the number of bytes allocated for the table of ISNs (TBI), which is used to store ISN lists (overflow ISNs or saved ISN lists). Each ISN list is identified by a command ID (CID) and stored in a TBI element.

You must also specify a value for the NSISN parameter, which sets the maximum number of ISNs per TBI element. Each TBI element has a length of 64 + (NSISN * 4) bytes.

The number of TBI elements required determines the length of the TBI. For complex searches, the Adabas nucleus builds temporary TBI elements in addition to the permanent TBI element; for example, for holding the results of descriptor searches, for reading the result records in Data Storage and collecting matches, or for holding the results of the partial criteria associated with the R operator. Although the temporary TBI elements are deleted at the end of a search, the TBI must be large enough to hold all of the temporary and permanent TBI elements in parallel during search processing.

The Adabas session statistics or Adabas Online System can be used to set this parameter for the next session.

When running with Adabas expanded files, a larger-than-normal LI value is recommended.

In environments running in 31-bit addressing mode, the LI pool is allocated above the 16-megabyte line.

Example

The following example allocates 30000 bytes for the table of ISN list IDs for this session.

```
ADARUN PROG=ADANUC,LI=30000
```

LNKGNAME Parameter: Link Globals Table Name

Parameter	Specify . . .	Possible Values	Default
LNKGNAME	the module name of the link globals table to be employed by an Adabas batch/TSO link routine. The link globals table is used to obtain default information and for invoking any linked exits.	valid module name	LNKGBLS

The parameter name LNKGNAME may be abbreviated to LGN.

Notes:

1. If you specify the LNKGNAME parameter, you must also specify the ADARUN DBID and SVC parameters.
2. To use this and other keywords for a batch or TSO job step, the application must be linked with the ADAUSER load module and call Adabas to invoke the ADAUSER load module.
3. If the job step specifies PROG=USER or PROG=RENTUSER with no LNKGNAME keyword, either the ADALNK or ADALNKR module will be loaded by ADARUN.

Example:

In the following example, the link globals table in module MYGBLS is used.

```
ADARUN PROG=USER,LNKGNAME=MYGBLS
```

LOCAL Parameter: Local Adabas Review Hub

Parameter	Specify . . .	Possible Values	Default
LOCAL	whether an Adabas nucleus or Adabas Review hub is isolated and available for local use only.	YES | NO	YES

Defines an isolated Adabas nucleus or Adabas Review hub that is only available locally. The nucleus or hub is unreachable to Entire Net-Work. A nucleus or Adabas Review hub specifying LOCAL=YES (the default) can have the same database ID or hub ID as another database nucleus or Review hub on another network node.

Value	Meaning
YES	Isolates this nucleus or Adabas Review hub (that is, makes it unaddressable) from other Entire Net-Work nodes.
NO	Allows the nucleus or Adabas Review hub to receive calls from other Entire Net-Work nodes.

Examples

In the following example, the Adabas nucleus is not isolated and can be addressed by other Entire Net-Work nodes.

```
ADARUN PROG=ADANUC,LOCAL=NO
```

In the following example, the Review hub is isolated and cannot be addressed by other Entire Net-Work nodes.

```
ADARUN PROG=ADAREV,LOCAL=YES
```

LOGGING / LOGxxxx Parameters: Command Logging Control

Parameter	Specify . . .	Possible Values	Default
LOGGING	whether to log Adabas commands.	YES | NO	NO

The LOGGING parameter specifies whether to use command logging for the Adabas session. The LOG*xxxx* parameters specify the type of information to be logged. Valid values are YES and NO for all of these parameters.

Note: User exit 4 is still called even if ADARUN LOGGING=NO and REVIEW is specified. User exit 4 will not be invoked if LOGGING=NO and REVIEW is not active. If REVIEW is specified, the only way to disable user exit 4 is to remove the ADARUN UEX4 parameter from the Adabas run.

If you specify LOGGING=YES, you must also specify YES for one or more of the following ADARUN LOG*xxxx* parameters. By default, each type of information is *not* logged.

Parameter	Specify whether to log the ...	Possible Values	Default
LOGABDX	Adabas buffer descriptions (ABDs)	YES | NO	NO
LOGCB	extended Adabas control block	YES | NO	NO
LOGCLEX	command log extension (CLEX) [1]	YES | NO	NO
LOGFB	format buffer	YES | NO	NO
LOGIB	ISN buffer	YES | NO	NO
LOGIO	I/O activity	YES | NO	NO
LOGMB	multifetch buffer	YES | NO	NO

Parameter	Specify whether to log the ...	Possible Values	Default
LOGRB	record buffer	YES | NO	NO
LOGSB	search buffer	YES | NO	NO
LOGUX	user exit B data	YES | NO	NO
LOGVB	value buffer	YES | NO	NO
LOGVOLIO	extended I/O list [2]	YES | NO	NO

[1]LOGCLEX=YES requires that Adabas Review modules must be available. The data provided in command log records with CLEX can be viewed using the batch program REVIEWB. For more information review the *Field Reference* in the Adabas Review Reference Guide.

[2] If the LOGIO parameter is set to "NO", no I/O buffer element is logged at all. If LOGIO is set to "YES" but LOGVOLIO is set to "NO", only the standard I/O buffer element is logged. The extended I/O list is only logged if both the LOGIO and LOGVOLIO parameter are set to "YES".

You can use Adabas operator commands or Adabas Online System to modify logging parameters during session execution.

Example

The following example runs the Adabas nucleus with command logging and logs the Adabas control block.

```
ADARUN PROG=ADANUC,LOGGING=YES,LOGCB=YES
```

LOGSIZE Parameter: Maximum Command Log Size

Parameter	Specify . . .	Minimum	Maximum	Default
LOGSIZE	the maximum block size for the sequential command log DDLOG.	100	32760	value of QBLKSIZE

This parameter controls the maximum block size for the sequential command log DDLOG. The default value is identical to the value of the QBLKSIZE parameter. If dual or multiple command logging is used, this parameter is ignored.

Example

In the following example, the block size of the command log is set as 2000 bytes.

```
ADARUN PROG=ADANUC,LOGSIZE=2000
```

LOGWARN Parameter: CLOG and PLOG Alert Message Interval

Parameter	Specify . . .	Minimum	Maximum	Default
LOGWARN	the frequency, in seconds, at which the PLOG and CLOG status is checked and resulting alert messages are produced.	0	2147483647	0

This parameter is optional, but a valid user exit 2 (ADARUN parameter **UEX2**) or user exit 12 (ADARUN parameter **UEX12**) must also be specified when this parameter is set to any value other than zero (0). A value of zero (0), the default, indicates that no CLOG and PLOG status checks should occur and that no alert messages should be produced. Any other valid LOGWARN value represents the interval (in seconds) at which the PLOG and CLOG status is checked and resulting alert messages are produced.

After checking the status of the CLOG and PLOG data sets, alert messages are written when Adabas detects that a PLOG or CLOG data set has not been copied within a specified time interval. Thereafter, an alert message is written at regular intervals as long as there is a PLOG or CLOG data set waiting to be copied. These alert messages are issued until the PLOG or CLOG data set is copied or until no logs remain to be copied. Your user exits 2 or 12 can be set up so that the messages appear providing a comprehensive trace of activity so you can resolve and problems with the PLOG or CLOG.

This feature helps you prevent the potential problem of Adabas overwriting an older CLOG or PLOG data set without waiting for direction from the user. Ordinarily, when a PLOG or CLOG data set fills up, the Adabas nucleus examines the return code from user exit 2 or user exit 12 to determine whether it should wait before overwriting an older PLOG or CLOG data set with new information. If there is no user exit 2 or 12, Adabas does not wait, but simply overwrites the older log. If this LOGWARN parameter is set to a value other than zero, the statuses of the CLOG and PLOG data sets are analzed at the frequency specified by this parameter, warning about their status and providing you with the information you need to prevent any overwriting of older logs.

Example

In the following example, the CLOG and PLOG statuses are checked every hour.

```
ADARUN PROG=ADANUC,LOGWARN=3600,UEX2=SUBR2
```

LP Parameter: Length of Data Protection Area (Work Part 1)

Parameter	Specify . . .	Minimum	Maximum	Default
LP	the number of blocks to allocate to Work part 1.	200	see text	1000

The LP parameter specifies the number of blocks to allocate to the Work data protection area, which is part 1 of the Work data set. This area must be large enough to accommodate the data protection information for the current transaction for all ET logic users.

The LP parameter is no longer limited to 65535 blocks; however, it is still limited by the size of the Work data set and the sizes of Work parts 2, 3, and 4.

Note: For any pending autorestart processing, Adabas retains and uses the value of the LP parameter from the previous session. If the LP parameter for the new session is specified with a different value than in the previous session, the nucleus will process any autorestart using the old value of LP, and establish the new LP size at the end of the autorestart.

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.
- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example uses 2,000 blocks to store data protection information on the Adabas Work data set.

```
ADARUN PROG=ADANUC,LP=2000
```

LQ Parameter: Length of Sequential Command Table

Parameter	Specify . . .	Minimum	Maximum	Default
LQ	the size of the table of sequential commands.	2000	2147483647	10 000

The table of sequential commands contains internal entries required during the processing of the Adabas read sequential (L2/L5, L3/L6, and L9) commands.

Each sequential pass of an Adabas file requires one entry in this table. Each table entry requires approximately 60-100 bytes, depending on the command type. Additionally, a variable portion (descriptor value) is required in the table entry for the L3/L6/L9 commands. An entry is deleted when an end-of-file condition is detected during sequential processing or when an RC or CL command is issued.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session. If using expanded files, the LQ value may need to be set to a higher-than-normal value.

The LQ value also determines the maximum number of command IDs that each user can specify. See the NQCID parameter description, later in this chapter, for more information.

In environments running in 31-bit addressing mode, the LQ pool is allocated above the 16-megabyte line.

Example

The following example allocates 30000 bytes for the table of sequential commands for this session.

```
ADARUN PROG=ADANUC,LQ=30000
```

LRDP Parameter: Length of the Redo Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LRDP	the size (in bytes) of the redo pool.	0; or 80000	2147483647	For Adabas Parallel Services 8.2 or later, the default is 0 (zero). For Adabas Cluster Services 8.2 or later, the default is either the value of the ADARUN LFIOP parameter or 10 megabytes (10M), whichever is smaller. For earlier releases of Adabas Parallel Services and Adabas Cluster Services, the default is the value of the ADARUN LFIOP parameter.

The LRDP parameter specifies the size of the redo pool in an Adabas Cluster Services or Adabas Parallel Services nucleus. It has an effect only when CLUSTER=SYSPLEX or LOCAL has also been specified. For Adabas Parallel Services 8.2 or later, its default is zero (0). For Adabas Cluster Services 8.2 or later, its default is either 10 megabytes (10M) or the value of the ADARUN LFIOP parameter, whichever is smaller. For earlier releases of Adabas Parallel Services and Adabas Cluster Services, the default is the value of the ADARUN `LFIOP` parameter.

- If LRDP is set to zero, the cluster nucleus performs immediate publishing: Updated database blocks are always written to the global cache at the time each update takes place.
- If LRDP is set to (or defaults to) a nonzero value, the cluster nucleus performs deferred publishing: The writing of updated database blocks to the global cache is deferred until later. However, updates are always published in the global cache before the transactions they belong to are committed (by ET commands). The redo pool is used to maintain descriptions of all unpublished updates, in case they need to be redone due to concurrent updates to the same blocks by other nuclei in the cluster.

By combining multiple updates to the same block into a single write-to-cache operation, deferred publishing tends to result in fewer read and write operations from and to the global cache than immediate publishing. Depending on the type of application workload, this may have a large positive effect on performance. This is especially the case for Adabas Cluster Services, where frequent communication with the cache in the coupling facility may cause a high overhead.

Recommendation: For Adabas Cluster Services, specify a nonzero LRDP parameter (or use the default).

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 another one with LRDP>0.

If the redo pool specification is too small and the pool runs full, the nucleus will write additional updated blocks to the global cache before the surrounding transactions end. Such additional cache writes may hurt system performance. To obtain more information about the current usage of the redo pool, review the redo high-water mark in the output of the DRES operator command and in the cluster nucleus shutdown statistics. This information is also available in corresponding Adabas Online System (AOS) displays and in the output of the ADADBS OPERCOM function. Please note that if a non-zero value is specified, the redo pool size will be rounded down to the nearest full-word boundary.

LRPL Parameter: Adabas or Event Replicator Replication Pool Size

Parameter	Specify . . .	Possible Values	Default
LRPL	The size of the Adabas or Event Replicator replication pools.	minimum 20000	100000 (if REPLICATION=YES) or 0 (if REPLICATION=NO)

Notes:

1. The LRPL parameter should be specified for an Adabas nucleus running with ADARUN parameter REPLICATION=YES. In addition, the replication pool will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters REPLICATION=YES and V64BIT=YES.
2. The LRPL parameter should be specified for an Event Replicator Server.
3. The LRPL parameter values in the Adabas nucleus and the Event Replicator Server may be different.
4. we recommend setting LRPL to a relatively large value (e.g. LRPL=40M or greater). To determine what this setting should be for your site, you could run Adabas in a test environment with replication turned on, simulate the activity in the production environment on this test system, and use the Adabas DRES operator command to view the current and high-water mark of replication pool usage.
5. If an Adabas nucleus with replication or Event Replicator Server is run with V64BIT=YES and LARGEPAGE=G64, the allocation request for the replication pool will be rounded up to the nearest multiple of two gigabytes. Any difference between the specified size and the allocated size will remain unused. For this reason, we recommend that if LARGEPAGE=G64, you specify the LRPL value as a multiple of two gigabytes to avoid leaving a potentially large amount of allocated storage unused. If LARGEPAGE=G64 is specified and 2G large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

Example:

```
ADARUN PROG=ADANUC,REPLICATION=YES,LRPL=50000K
```

LS Parameter: Length of Internal Sort Area

Parameter	Specify . . .	Minimum	Maximum	Default
LS	the maximum number of bytes to be used for internal sort processing.	19968	see note	49920

Note: To ensure that a single search command does not use too much of the work pool space, the maximum LS value must be less than or equal to the LWP (work pool length) divided by two minus the minimum LS value (19968) (`LS = LWP/2 - 19968`). See the ***LWP parameter***.

The LS parameter specifies the maximum number of bytes within the Work pool to be used for sort work space during the processing of a single command of the following type:

- S2 and S9 commands;
- complex search commands, which require the storage of bit strings.

The following formula may be used to estimate the number of Adabas ISNs that may be sorted in the work space provided (S2 or S9 commands):

If the TOPISN in the file is less than 65536:

Using	Formula to determine the number of Adabas ISNs that may be sorted in the work space provided
using 1 descriptor	LS/12
using 2 descriptors	LS/16
using 3 descriptors	LS/20

Notes:

1. TOPISN is the highest ISN currently in use for the file, less (MINISN - 1). TOPISN is not controlled by a parameter.
2. 4-byte ISNs are used for the internal sort area, regardless of the ISNSIZE specified for the file.

If the TOPISN in the file is greater than 65536:

Using	Formula to determine the number of Adabas ISNs that may be sorted in the work space provided
using 1 descriptor	LS/14
using 2 descriptors	LS/20
using 3 descriptors	LS/26

For complex searches, it is desirable for performance reasons to set LS value to the next integer multiple of 256 bytes greater than TOPISN / 8. For example, if TOPISN = 480000, then TOPISN / 8 = 60000, so LS should be set to the next higher multiple of 256, namely 60160.

Example

In the following example, the maximum amount of internal sort space during this session is set at 60160 bytes.

```
ADARUN PROG=ADANUC,LS=60160
```

LSF Parameter: Logical Save Facility

Parameter	Specify . . .	Possible Values	Default
LSF	whether Logical Save Facility should be enabled (YES) or disabled (NO) for the nucleus.	YES | NO	NO

The LSF parameter enables the nucleus to support Logical Save Facility processing. In the majority of sites Logical Save Facility will not be installed and this parameter should be set to NO (or allowed to default).

Value	Meaning
YES	Logical Save Facility processing is to be enabled.
NO	The default setting. Logical Save Facility processing is to be disabled.

LTZ Parameter: Time Zone Pool Size

Parameter	Specify . . .	Minimum	Maximum	Default
LTZ	the size of the time zone pool (in bytes) where time zone information is kept for each time zone used in a user session.	0, 12288	2147483647	32768

Note: You can specify values for this parameter with a "K" to indicate that the number you specify is in kilobytes. For example, a specification of LTZ=32K is a valid specification.

When LTZ=0 is specified, no time zone pool is allocated. This disables the use of time zones for user sessions. The minimum, non-zero allocation is 12288 (12K).

Time zone information varies in size with an average of 1000 bytes. So the default of 32768 (32K) allows for approximately 32 time zones.

Finally, while the maximum value that can be specified for the LTZ parameter is 2147483647 bytes, only 491520 bytes (480K) is required to load all available time zones.

Example

In the following example, the minimum size (12288 bytes) is set for the time zone pool.

```
ADARUN PROG=ADANUC,LTZ=12288
```

LU Parameter: Length of Intermediate User Buffer Area

Parameter	Specify . . .	Minimum	Maximum	Default
LU	the size of the intermediate user buffer area.	none	none (see note 1)	65535 (see note 2)

The LU parameter is optional. The size specified must be large enough to accommodate all Adabas control information (204 bytes), all user buffers (format, record, search, value, ISN) that may be required for any given Adabas command, plus any user information passed from Adabas link routines to nucleus user exits. If the ADALNK has user exits, the addresses of the ACB and each buffer passed will be stored in this space. Also, if the ADALNK has declared LUSIZE and has user exits, this buffer size plus 64 bytes will also be stored in this space.

If the multifetch/prefetch option or a utility that needs large record/ISN buffers is to be used during the session, the setting of LU must be large enough to contain the buffers needed.

If you are using the ADARPD IQETBBROKERID NET communication (used with Event Replicator for Adabas), make sure that the IBM EntireX Broker NET definition parameter IUBL is set to a value as large as the setting of this LU ADARUN parameter.

Notes:

1. An error occurs if the LU parameter specifies a value greater than the byte count implied by the NAB (number of attached buffers) parameter. LU cannot exceed a value greater than that produced by the following calculation: `(NABvalue x 4096)`. For more information about the NAB parameter, read ***NAB: Number of Attached Buffers***.
2. Due to the length of the record buffer of the utilities that need the nucleus, e.g., ADAULD, the default value is set to 65,535. If the value of LU is less than 65,535 for an Adabas session, a response code will occur when such a utility is running.

The LU parameter syntax is:

```
LU={ n | 65535 }
```

Specific Product Recommendations

- For Event Replicator Server databases running with Adabas 8, the LU parameter must be greater than or equal to 167,000.

  In addition, if data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work LU parameter must be greater than or equal to the LU parameter setting for the Event Replicator Server itself (greater than or equal to 167,000).
- The ADACHK utility can use large record buffer lengths when making nucleus calls to verify spanned Data Storage records or an index structure with many levels. If this is the case, the settings of your LU and NAB ADARUN parameters may need to be increased.

- For an Adabas nucleus running with the ADARUN REPLICATION=YES parameter, ensure the LU ADARUN parameter is set to a value greater than or equal to 70000.
- For an Adabas Audit Server, the LU parameter must be greater than or equal to 167,000.

  In addition, if audit data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work LU parameter must be greater than or equal to the LU parameter setting for the Adabas Audit Server itself (greater than or equal to 167,000).

Example

The following example runs the Adabas nucleus with an Adabas intermediate user buffer area of 20,000 bytes.

```
ADARUN PROG=ADANUC,LU=20000
```

LWKP2 Parameter: Length of ISN List Processing Area (Work Part 2)

Parameter	Specify . . .	Minimum	Maximum	Default
LWKP2	the number of Work blocks to use for intermediate ISN lists.	0	16777215	0

This parameter specifies the size (in blocks) of the Adabas Work part 2, which holds intermediate ISN lists during S*x* commands.

If LWKP2=0 (the default), Adabas calculates the Work part 2 size automatically using the following formula:

```
WORK2 = 22 + 2 * (4 * RECORDS / (BLKSIZE - 16))
```

where

WORK2	is the Work part 2 space requirement in blocks.
RECORDS	is the number of records in the file with the most records. This number equals TOPISN - MINISN + 1, where TOPISN is the highest ISN currently used in the file. MINISN is the lowest ISN used in the file. The MINISN value is specified with the ADACMP/ADALOD parameter MINISN; 1 is the default. You can use the ADAREP utility to display the TOPISN and MINISN values for the files in a database.
BLKSIZE	is the block size of the device where the Work data set is stored (see the Adabas Installation documentation).

If you choose not to use LWKP2=0, the minimum value is LWKP2=20 blocks.

Note: An Adabas internal table requires one byte of storage for each Work part 2 block.

Example:

For example, for the following Adabas session, the size of the Adabas Work part 2 is 50000 blocks.

```
ADARUN PROG=ADANUC,LWKP2=50000
```

LWP Parameter: Length of Adabas Work Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LWP	the size of the Adabas work pool.	80000 but not less than NT * 25000	2147483647	150000

Note: The LWP (work pool length) parameter value must be equal to or greater than the LS (internal sort area length) value plus its minimum value (19968) multiplied by 2; that is, `(LS + 19968) * 2`.

The Adabas work pool area is used to store the following:

- descriptor value tables (DVT);
- intermediate user buffer areas (see parameter LU);
- sort work space areas (see parameter LS);
- work space during session and command execution.

In environments running in 31-bit addressing mode, the work pool space is allocated above the 16-megabyte line.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session. If expanded files are being used, the LWP value may need to be increased above the normal setting.

Specific Product Recommendations

If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example runs the Adabas nucleus with a work pool of 900000 bytes.

```
ADARUN PROG=ADANUC,LWP=900000
```

MLWTO Parameter: Multi-Line Console Message Control

Note: At this time, this parameter only affects the appearance of replication-related and cluster-related messages (messages with prefixes ADAF*, ADAX*, PLI*, and PLX*).

Parameter	Specify . . .	Possible Values	Default
MLWTO	Whether multi-line messages that are normally written to the operator console as a series of single-line write-to-operator (WTO) messages should instead appear as multi-line WTO messages.	YES | NO	NO

To activate multi-line console messages, set MLWTO to YES in the nucleus startup job. When MLWTO is set to NO (or is allowed to default), no multi-line console messages will appear.

Example:

```
ADARUN PROG=ADANUC,MLWTO=YES
```

MODE Parameter: Mode of Operation

Parameter	Specify . . .	Possible Values	Default
MODE	the Adabas operating mode.	SINGLE | MULTI	MULTI

Value	Meaning
SINGLE	Run in single-user mode: Load the Adabas nucleus and the user program or Adabas utility into the same region.
MULTI	Run in multiuser mode: Load only the ADALNK routine into the same region as the program/utility. Load the Adabas nucleus into a different region. ADALNK will manage all interregion communication between the program or utility and the nucleus.

Note the following:

1. The MODE parameter does not affect parallel utility execution.
2. The nucleus cannot run in single-user mode (MODE=SINGLE cannot be specified) if it is to run
   - with Adabas Parallel Services or Adabas Cluster Services (NUCID 1)
   - with support for distributed transactions (DTP=RM)
   - as an Adabas Transaction Manager (DTP=TM)

   In addition, a valid Adabas SVC should be specified when running in single-user (MODE=SINGLE) mode.
3. When running several nuclei against the same database, only one update nucleus can be used.

Example

The following example runs the ADAINV utility in single-user mode.

```
ADARUN PROG=ADAINV,MODE=SINGLE,SVC=234
```

MSGBUF Parameter: Size of Message Buffer

Parameter	Specify . . .	Minimum	Maximum	Default
MSGBUF	the size of the message buffer area.	0	128	0

MSGBUF specifies the size (in kilobytes) of the area used to buffer messages for use for performance and tuning and problem analysis. Adabas messages buffered in this area can be reviewed even when online access to the console or DDPRINT messages is unavailable. While the buffer is intended for messages written to the system log or DDPRINT, it can be used for any purpose by any caller.

MSGBUF presupposes that ADARUN SMGT=YES.

MSGBUF=0 (the default) means that message buffering is not active for the session.

To activate message buffering, set MSGBUF to a value greater than 0. we recommend setting MSGBUF to 10 (kilobytes).

Message buffering can be temporarily disabled and subsequently reactivated using the `SMGT` **operator command**.

See the *Adabas DBA Tasks Manual* for more information about message buffering.

Example

The following example activates message buffering when the Adabas nucleus is initialized with a message buffer size of 10 KB.

```
ADARUN PROGRAM=ADANUC
  ADARUN SMGT=YES
  ADARUN MSGBUF=10
```

MSGCONSL/MSGDRUCK/MSGPRINT Parameter: Case of Messages Sent to the Console/DDDRUCK /DDPRINT

Parameter	Specify the case of Adabas . . .	Possible Values	Default
MSGCONSL	messages sent to the console	UPPER | MIXED	MIXED
MSGDRUCK	messages sent to the DDDRUCK data set		
MSGPRINT	messages sent to the DDPRINT data set		

Adabas messages are displayed in mixed upper and lowercase. You can use one or more of these ADARUN parameters to display different message types in all uppercase.

Notes:

1. The first message line in DDPRINT will not be influenced by the value of parameter MSGPRINT.
2. Lowercase fields will not display correctly (they will be converted to uppercase) if you use the ADARUN parameter settings MSGCONSL=U, MSGDRUCK=U, or MSGPRINT=U.

Example

The following example will display messages in the DDDRUCK and DDPRINT data sets in all uppercase:

```
ADARUN MSGDRUCK=U,MSGPRINT=U
```

MXCANCEL Parameter: Timeout Threshold for Cancelled Peer Nucleus

Parameter	Specify . . .	Minimum	Maximum	Default
MXCANCEL	the timeout threshold, in seconds, in which the processing of an intracluster cancellation request must occur.	15	2147483647	MXMSG parameter value

The optional MXCANCEL parameter specifies the maximum number of seconds for which one cluster nucleus (the sending nucleus) waits for the termination of another nucleus (the peer nucleus) in the cluster that it has cancelled for not responding to an internal, intracluster command. If a value for MXCANCEL is not specified, the value specified for the ADARUN MXMSG parameter is used. Values from zero through 14 are not valid MXCANCEL values.

If the sending nucleus does not receive notification that the cancelled nucleus has terminated within the MXCANCEL timeout period, it does one of the following things:

- It returns response code 124 (ADARSP124), subcode 28 (no reply) to the user on whose behalf the intracluster command was issued.
- It terminates itself abnormally if the communication occurred on behalf of an internal process that must not fail.

When a cluster nucleus fails, the other nuclei in the cluster can recover from the failure only once they know for sure that the first nucleus has disconnected from the shared structures and is no longer writing to the database. When an unresponsive cluster nucleus has been cancelled, it is important that it terminate quickly, so that it does not hold up the recovery process performed by the other nuclei in the cluster.

Note: Do not specify an MPMDUMP JCL statement for a cluster nucleus. An MPMDUMP is written before the failing nucleus disconnects from the shared structures and may significantly delay the start of the online recovery process performed by the other nuclei in the cluster. Instead, consider specifying a //SVCDUMP DD DUMMY statement, which usually produces a very quick dump prior to disconnecting. Alternatively, specify a regular system dump statement (e.g., SYSUDUMP) for a dump that is produced after disconnecting.

The value for MXCANCEL can be changed dynamically (for the time the Adabas nucleus is running) using the MXCANCEL operator command.

MXCANCELWARN Parameter: Timeout Threshold for Cancelled Peer Nucleus Warning

Parameter	Specify . . .	Minimum	Maximum	Default
MXCANCELWARN	the timeout threshold, in seconds, for an intracluster cancellation request to occur before an early warning about the failure of the cancellation request is issued.	0 (no warning) or 4	MXCANCEL - 1	MXCANCEL / 4

The optional MXCANCELWARN parameter specifies the number of seconds after an intracluster cancellation request is made when the cluster nucleus requesting the cancellation should issue warnings about the inability of the cancelled peer nucleus to terminate. If no value is specified for the MXCANCELWARN parameter, the value used is calculated as the value of the ADARUN MXCANCEL parameter divided by four. When dividing by four to determine the default value, the quotient is rounded up to the next integer value.

The value of MXCANCELWARN must be less than the value of MXCANCEL. If MXCANCELWARN is set to a value greater than or equal to the value specified for MXCANCEL, the value is rejected. If MXCANCELWARN is set to a value of zero (0), no warnings are issued about cancelled cluster nuclei that are slow to terminate.

If the target nucleus does not terminate within the time period set by MXCANCELWARN, the nucleus that issued the cancellation request writes message ADAX9G to the operator console, identifying the target nucleus by its nucleus ID, job name, and system name. This message indicates that the nucleus writing the message may be in danger of terminating itself if the cancelled peer nucleus does not terminate within the timeout period defined by the MXCANCEL parameter. You can use the ADAX9G message to trigger an alert or take other appropriate action before the MXCANCEL timeout period expires.

If the peer nucleus targeted for cancellation terminates after the ADAX9G message is written, the nucleus that issued the cancellation request writes message ADAX9K to the operator console, indicating that the cancellation occurred, and the requesting nucleus is no longer in danger of terminating itself as a result of this particular incident.

The value for MXCANCELWARN can be changed dynamically (for the time the Adabas nucleus is running) using the MXCANCELWARN operator command.

Note: If the value of the MXCANCEL parameter is dynamically changed using the MXCANCEL operator command while the nucleus is running, but its new value is less than or equal to the current MXCANCELWARN setting, the value of MXCANCELWARN is automatically set to zero. You can then use the MXCANCELWARN operator command to dynamically set the MXCANCELWARN parameter to a new value that is less than the new MXCANCEL value.

MXMSG Parameter: Timeout Threshold for Internucleus Command Processing

Parameter	Specify . . .	Minimum	Maximum	Default
MXMSG	the timeout threshold, in seconds, in which the processing of an intracluster command must occur.	15	32767	300

The optional MXMSG parameter specifies the maximum number of seconds in which one cluster nucleus waits for the response of another nucleus in the cluster to an internal, intracluster command. If a value for MXMSG is not specified, a value of 300 seconds (5 minutes) is used.

If the receiving nucleus does not respond to the intracluster command within the time period set by the MXMSG parameter, the requesting cluster nucleus issues a cancellation request for the unresponsive nucleus. The cancellation is requested to preserve the ability of all of the other active nuclei in the cluster to process user commands.

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 intermittently submit internal, intracluster commands to one another. If a nucleus in the cluster does not respond to these intracluster requests in a timely manner, the other nuclei in the cluster may quickly or gradually become unable to continue processing user commands.

Problems that may prevent a cluster nucleus from responding to intracluster commands fall into two broad categories:

- Problems have occurred in the cluster nucleus, such as a CPU loop or a resource deadlock. In such cases, the nucleus experiencing the problem must be terminated to resolve the problem and the sooner this happens, the sooner the other cluster nuclei can recover from the failure and get back to normal processing.
- Problems have occurred outside the cluster nucleus, such as CPU starvation or a problem in the host system. In such cases, the cause of the delayed response might possibly be removed by system measures unrelated to the cluster database, and letting the other cluster nuclei wait some time for a resolution might prevent an unnecessary abnormal termination of a cluster member.

The value of MXMSG should strike a balance between these two categories, so that faulty cluster nuclei that cannot possibly stay active are canceled reasonably quickly while cluster nuclei that are victims of temporary system problems are not unnecessarily terminated. For assistance in setting the value of the MXMSG parameter, review the statistics provided on the messaging performance reports produced at termination or by the DXMSG operator command.

The value for MXMSG can be changed dynamically (for the time the Adabas nucleus is running) using the MXMSG operator command.

MXMSGWARN Parameter: Timeout Threshold for Internucleus Command Processing Warning

Parameter	Specify . . .	Minimum	Maximum	Default
MXMSGWARN	the timeout threshold, in seconds, for a response to an intracluster command to occur before an early warning about the failure of the intracluster request is issued	0 (no warning) or 4	MXMSG - 1	MXMSG / 4

The optional MXMSGWARN parameter specifies the number of seconds after an intracluster request is made when the cluster nucleus issuing the intracluster request should issue warnings about the failure of the target cluster nucleus to respond. If no value is specified for the MXMSGWARN parameter, the value used is calculated as the value of MXMSG divided by four. When dividing by four to determine the default value, the quotient is rounded up to the next integer value.

The value of MXMSGWARN must be less than the value of MXMSG. If MXMSGWARN is set to a value greater than or equal to the value specified for MXMSG, the value is rejected. If MXMSGWARN is set to zero, no warnings are issued about cluster nuclei that are slow to respond.

If the target nucleus does not respond within the time period set by MXMSGWARN, plus any extra time allowed for some types of requests (such as ET synchronization), the nucleus that issued the request writes message ADAX9C to the operator console, identifying the target nucleus by its nucleus ID, job name, and system name. This message indicates that the target nucleus may be in danger of being canceled by the nucleus that issued the request if it does not respond within the timeout period defined by the MXMSG parameter. You can use the ADAX9C message to trigger an alert or take other appropriate action before the MXMSG timeout period expires.

If the target nucleus responds to the intracluster command after the ADAX9C message is written, the nucleus that issued the intracluster-command writes message ADAX9D to the operator console, indicating that a response was given and the target nucleus is no longer in danger of being canceled as a result of this particular incident.

The value for MXMSGWARN can be changed dynamically (for the time the Adabas nucleus is running) using the MXMSGWARN operator command.

Note: If the value of the MXMSG parameter is dynamically changed using the MXMSG operator command while the nucleus is running, but its new value is less than or equal to the current MXMSGWARN setting, the value of MXMSGWARN is automatically set to zero. You can then use the MXMSGWARN operator command to dynamically set the MXMSGWARN parameter to a new value that is less than the new MXMSG value.

MXSTATUS Parameter: Member-Level XCF Status Monitoring Heartbeat Interval

Parameter	Specify . . .	Minimum	Maximum	Default
MXSTATUS	the interval, in seconds, at which the member-level XCF status monitoring heartbeat should occur.	0 (no member-level status monitoring) or 15	21474836	15

The optional MXSTATUS parameter specifies the number of seconds between member-level XCF status monitoring heartbeats. If no value is specified for the MXSTATUS parameter, a value of "15" is used.

Note: Note: The MXSTATUS parameter is only used by Adabas Cluster Services and not by Adabas Parallel Services. Adabas Parallel Services does not use XCF and ignores this parameter when it is set.

If MXSTATUS is set to zero, member-level status monitoring will not occur. However, system-level status monitoring always occurs.

The value for MXSTATUS can be changed dynamically (for the time the Adabas nucleus is running) using the MXSTATUS operator command.

XCF status monitoring provides a second method by which Adabas Cluster Services can warn you that a cluster nucleus might be unable to respond in a timely way to intracluster commands. The first method is, of course, via the normal intracluster communication that occurs between cluster members. If a nucleus has heartbeat exceptions (as determined by XCF status monitoring), it most likely will be unable to process and respond to an intracluster command; if a nucleus is slow to respond to an intracluster command, it might or might not have a heartbeat monitor exception (a nucleus may appear to be active to XCF but be unable to respond to an intracluster command). If the ADARUN MXMSGWARN parameter for a cluster nucleus is nonzero, it produces warning messages (ADAX9B or ADAX9C) when intracluster communication with other nuclei in the cluster is too slow; likewise, when XCF status monitoring determines that a nucleus is missing its heartbeat updates, it produces warning messages (ADAX22 and ADAX04). You can use an automated mechanism set up at installation to raise an alert or take other appropriate action based on the existence of these messages, as they identify existing or potential problems in the cluster.

MXTNA Parameter: Maximum Non-activity Time

Parameter	Specify . . .	Minimum	Maximum	Default
MXTNA	the maximum non-activity time limit that can be specified in the OP command for an individual user.	1	65535	3600

The TNAx parameter specifies the maximum elapsed time (in units of 1.048576 seconds) that users may be considered active without issuing an Adabas command. Limits may be set for three types of user:

- the TNAA **parameter** for access-only users
- the TNAE **parameter** for ET logic users
- the TNAX **parameter** for exclusive control users

You can override these parameter settings for an individual user by specifying a different value in the Adabas control block of an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.

For these parameters, the MXTNA parameter sets a limit on the amount of time that can be specified in the OP command for an individual user.

If the nonactivity time limit is exceeded, action is taken as described in ***Adabas Timeout Feature***, elsewhere in this guide.

The value of the MXTNA parameter can only be set when the nucleus is started. It cannot be changed using operator commands or the ADADBS OPERCOM utility function.

Example

For example, during the following Adabas session, the nonactivity time limit for exclusive control users is about 5 minutes (300 seconds). This limit cannot be extended for an individual user beyond 8 minutes (480 seconds).

```
ADARUN PROG=ADANUC,TNAX=300,MXTNA=480
```

MXTSX Parameter: Maximum Sx Command Execution Time

Parameter	Specify . . .	Minimum	Maximum	Default
MXTSX	the maximum execution time (in seconds) that can be specified for the **ADARUN** TLSCMD parameter and for any override execution time setting that might be specified in the Adabas control block of an OP command.	1	65535	3600

The TLSCMD parameter specifies the maximum time (in units of 1.048576 seconds) to be used to process a single Adabas S1, S2, or S4 command with a complex search criterion.

If the estimated time to process a command exceeds the TLSCMD time limit, the command is not processed and the user who issued the command receives a nonzero response code.

If the estimated time to process a command does not exceed the time limit, the command is processed to completion, whether the actual time required exceeds the time limit specified by the TLSCMD or not.

You can override the TLSCMD parameter setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.

The MXTSX parameter sets the maximum time that can be specified for the TLSCMD parameter, including any override execution time setting specified in the Adabas control block of an OP command.

The value of the MXTSX parameter can only be set when the nucleus is started. It cannot be changed using operator commands or the ADADBS OPERCOM utility function.

Example

For example, during the following Adabas session, the S*x* command execution time limit is about 1 hour 20 minutes (4800 seconds). This limit cannot be extended for an individual user beyond 2 hours (7200 seconds).

```
ADARUN PROG=ADANUC,TLSCMD=4800,MXTSX=7200
```

MXTT Parameter: Maximum Transaction Time

Parameter	Specify . . .	Minimum	Maximum	Default
MXTT	the maximum transaction time limit that can be specified in the OP command for an individual user.	1	65535	3600

The TT parameter specifies the maximum elapsed time (in units of 1.048576 seconds) permitted for a logical transaction issued by an ET logic user. See ADARUN parameter TT for more information.

You can override this parameter setting for an individual user by specifying a different value in the Adabas control block of an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.

The MXTT parameter sets a limit on the amount of time that can be specified in the OP command for an individual user.

The time measurement for a logical transaction starts when the first command that results in a record being placed in hold status is issued. It ends when an ET, BT, or CL command is issued.

If the transaction time limit is exceeded, action is taken as described in ***Adabas Timeout Feature***, elsewhere in this guide.

The value specified for MXTT directly influences the required size of the Adabas parameter LP (length of the data protection area).

we recommend that you set the MXTT parameter lower than the MXTNA **parameter**.

The value of the MXTT parameter can only be set when the nucleus is started. It cannot be changed using operator commands or the ADADBS OPERCOM utility function.

Example

For example, during the following Adabas session, the transaction time limit for ET logic users is about 3 minutes (180 seconds) and cannot be extended for an individual user beyond 5 minutes (300 seconds).

```
ADARUN PROG=ADANUC,TT=180,MXTT=300
```

MXWTOR Parameter: Self-Termination Operator Query Interval

Parameter	Specify . . .	Minimum	Maximum	Default
MXWTOR	the timeout interval, in seconds, in which an operator response to a self-termination operator query must occur.	0 (no warning) or 15	64800	0

The optional MXWTOR parameter specifies the maximum number of seconds that a cluster nucleus should wait for an operator response to a self-terminating operator query. If no value is specified for the MXWTOR parameter, a value of zero is used. Values from 1 through 14 are not valid MXWTOR values.

A cluster nucleus issues a self-terminating operator query to the console via message ADAX9J if a canceled peer nucleus remains active after the time period defined by the MXCANCEL parameter. The MXWTOR parameter specifies how long the nucleus should wait for a response to this query.

One way that the operator may reply to the ADAX9J message is to make sure that the canceled cluster nucleus mentioned in the message actually terminates; once this happens, the ADAX9J message is retracted automatically, an online recovery process is performed, and the nucleus that issued the ADAX9J message remains active.

If the operator does not respond to the ADAX9J message within the time period set by the MXWTOR parameter and the canceled peer nucleus has not terminated in the meantime, the nucleus that requested the operator response and printed message ADAX9J terminates itself with message ADAX99.

If MXWTOR is set to zero (the default), the operator query does not occur (no ADAX9J messages are written) and no operator intervention is expected. Instead, the nucleus terminates itself immediately with the ADAX99 message and user abend code 79.

Some installations may want to specify MXWTOR value long enough to allow the operator a good opportunity to bring down the canceled peer nucleus. Other installations may prefer not to get such operator queries at all.

The value for MXWTOR can be changed dynamically (for the time the Adabas nucleus is running) using the MXWTOR operator command.

NAB Parameter: Number of Attached Buffers

Parameter	Specify . . .	Minimum	Maximum	Default
NAB	the number of attached buffers to be used.	1	varies, depending on the amount of available virtual storage	16

The NAB parameter defines the number of attached buffers to be used during the session. An attached buffer is an internal buffer used for interregion communication. It is required in all environments. Adabas allocates an attached buffer pool with a size equal to the value of NAB multiplied by 4096 bytes.

Note: The allocation for buffers in the attached buffer pool is done in 256 byte slots; this means that each allocation is rounded to a multiple of 256. For example, if a size of 300 bytes is needed, the allocated space is 512 bytes.

You may specify as many attached buffers as fit into the available virtual storage.

In environments running in 31-bit addressing mode, the attached buffer pool space is allocated above the 16-MB line.

The NAB parameter syntax is:

```
NAB={ n | 16 }
```

Specific Product Recommendations

- For Event Replicator Server databases, set parameter `NAB` to a value greater than or equal to: `41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server`.

  For example, if one Adabas nucleus will be sending data to the Event Replicator Server, set the `NAB` parameter greater than or equal to 410 (for example NAB=420).
- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.
- If data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work NAB parameter must also be set to a value greater than or equal to: `41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server`.
- Users of the Adabas Review hub should read *Storage Requirements* in the *Adabas Review Concepts Manual* for more information about the space requirements of the Command Queue for Adabas Review.

- The ADACHK utility can use large record buffer lengths when making nucleus calls to verify spanned Data Storage records or an index structure with many levels. If this is the case, the settings of your LU and NAB ADARUN parameters may need to be increased.
- For an Adabas Audit Server, set parameter NAB to a value greater than or equal to:

  ```
  41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
  ```

  For example, if one Adabas nucleus will be sending data to the Adabas Audit Server, set the NAB parameter greater than or equal to 410 (e.g. `NAB=420`).

  If data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work NAB parameter must also be set to a value greater than or equal to:

  ```
  41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
  ```

Example

The following example runs the Adabas Review hub nucleus with 50 attached buffers.

```
ADARUN PROG=ADAREV,NAB=50
```

NALOG Parameter: Number of Audit Log Data Sets

Parameter	Specify . . .	Minimum	Maximum	Default
NALOG	the maximum number of audit log data sets to be opened and logged in the PPT.	2	8	0

This parameter specifies the maximum number of audit log data sets to be opened and logged in the parallel participant table (PPT) for the current Audit Server. Valid values are 0 (the default) and 2-8. Other values produce a parameter error. A zero value disables the parameter.

If more than the number of log data sets specified in the NALOG parameter are allocated in the startup JCL, only the number specified in the NALOG parameter are opened and logged; any additional allocated data sets are activated using the dynamic allocation process.

NALOG must be specified with a nonzero value in conjunction with the ALOGDEV, ALOGSIZE, and UEX12 parameters.

NALOG is not valid if DUALALD, DUALALS, or UEX2 are specified.

UEX2 and UEX12 are mutually exclusive.

If a nonzero NALOG, ALOGDEV, and ALOGSIZE are specified, but `LOGGING=NO`, Adabas attempts to open and log the number of data sets specified by NALOG; however, no logging occurs until

the LOGGING parameter has been set to YES using either an operator command or Adabas Online System.

Example

The following example runs the Adabas Audit Server with audit logging. It uses six audit log data sets and a device type 3390 for audit logging. The size of each audit log data set is 1500 blocks, based on an allocation of 10 cylinders. Each cylinder of a 3390 device has 15 tracks, and each track can contain 10 blocks of audit log data. Thus, 10 cylinders = 10 x 15 x 10 = 1500 blocks.

```
ADARUN PROG=ADANUC,LOGGING=YES,NALOG=6,ALOGDEV=3390,ALOGSIZE=1500
```

NC Parameter: Number of Command Queue Elements

Parameter	Specify . . .	Minimum	Maximum	Default
NC	the maximum number of command queue elements.	20	32767	200

The number of command queue elements (CQEs) established for the Adabas or Review hub session determines the maximum number of Adabas commands that may be queued or be in process at any one time during the session.

Each call from the Adabas nucleus is assigned a CQE. The CQE is released when the user receives the results of the command, the Adabas Review hub has processed the command, or the user has been timed out..

192 bytes are required for each CQE.

we recommend that you set NC high enough to allow one command per active user for possible synchronization during execution of the online SAVE database function of the ADASAV utility.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

For more information about the space requirements of the Command Queue for Adabas Review, refer to *Storage Requirements* in the *Adabas Review Concepts Manual*.

Specific Product Recommendations

- For Event Replicator Server databases, set parameter `NC` to a value greater than or equal to: `10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server`. For example, if one Adabas nucleus will be sending data to the Event Replicator Server, set the `NC` parameter greater than or equal to 10 (for example NC=20).
- If data is sent through Entire Net-Work from one or more Adabas nuclei to an Event Replicator Server, the Entire Net-Work NC parameter must also be set to a value greater than or equal to: `10 * the-number-of-Adabas-nuclei-sending-data-to-the-Event-Replicator-Server`.

- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.
- For an Adabas Audit Server, set parameter NC to a value greater than or equal to:

  ```
  10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
  ```

  For example, if one Adabas nucleus will be sending data to the Adabas Audit Server, set the NC parameter greater than or equal to 10 (e.g. `NC=20`).

  If data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work NC parameter must also be set to a value greater than or equal to:

  ```
  10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
  ```

Example:

Run the Adabas nucleus with a maximum of 500 elements in the command queue.

```
ADARUN PROG=ADANUC,NC=500
```

The following example runs the Adabas Review hub nucleus with a maximum of 500 elements in the command queue.

```
ADARUN PROG=ADAREV,NC=500
```

NCLOG Parameter: Number of Command Log Data Sets

Parameter	Specify . . .	Minimum	Maximum	Default
NCLOG	the maximum number of command log data sets to be opened and logged in the PPT.	0, 2	8	0

This parameter specifies the maximum number of command log data sets to be opened and logged in the parallel participant table (PPT) for the current nucleus. Valid values are 0 (the default) and 2-8. Other values produce a parameter error. A zero value disables the parameter.

If more than the number of log data sets specified in the NCLOG parameter are allocated in the startup JCL, only the number specified in the NCLOG parameter are opened and logged; any additional allocated data sets are activated using the dynamic allocation process.

NCLOG must be specified with a nonzero value in conjunction with the CLOGDEV, CLOGSIZE, and UEX12 parameters.

NCLOG is not valid if DUALCLD, DUALCLS, DUALPLD, DUALPLS, or UEX2 are specified.

UEX2 and UEX12 are mutually exclusive.

If a nonzero NCLOG, CLOGDEV, and CLOGSIZE are specified, but LOGGING=NO, Adabas attempts to open and log the number of data sets specified by NCLOG; however, no logging occurs until the LOGGING parameter has been set to YES using either an operator command or Adabas Online System.

Example

The following example runs the Adabas nucleus with command logging, uses six command log data sets, and uses a model 3390 device for command logging. The size of each command log data set is 1500 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3390 device has 15 tracks, and each track can contain 10 blocks of command log data. Thus, 10 cylinders = 10 x 15 x 10 = 1500 blocks).

```
ADARUN PROG=ADANUC,LOGGING=YES,NCLOG=6,CLOGDEV=3390,CLOGSIZE=1500
```

NH Parameter: Number of Hold Queue Elements

Parameter	Specify . . .	Minimum	Maximum	Default
NH	the maximum number of hold queue elements.	20	16777215	500

One hold queue element is required for each record (ISN) placed in hold status. A hold queue element is released when the record is released from hold status.

The hold queue is normally used to hold ISNs for the duration of a transaction. Therefore NH should be large enough to contain at least the average number of ISNs held per transaction multiplied by the maximum number of transactions running in parallel.

One hold queue element takes 52 bytes of storage. In environments running in 31-bit addressing mode, the hold queue is allocated above the 16-megabyte line.

Each shared hold request with a different command ID (CID), as well as a (shared or exclusive) hold request without a CID, is counted against the NH limit.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

Specific Product Recommendations

- Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.
- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"),

consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example runs the Adabas nucleus with a maximum of 2000 elements in the hold queue.

```
ADARUN PROG=ADANUC,NH=2000
```

NISNHQ Parameter: Number of ISNs in Hold Queue per User

Parameter	Specify . . .	Minimum	Maximum	Default
NISNHQ	the maximum number of records that can be placed in hold status at the same time by one user.	1	16777215 (NH parameter maximum)	(NH / NU) * 2 or 20, whichever is larger

NISNHQ determines the maximum number of held ISNs the hold queue (see the NH parameter description) can contain from one user. The maximum value allowed for this parameter is the value set for the NH ADARUN parameter (which has a maximum value of 16777215). The default is the greater of either 20 or the number of ISNs the hold queue can contain divided by the number of users, multiplied by two.

A user who attempts to place more records in hold status than permitted receives response code 47 (ADARSP047), subcode 1.

Each shared hold request with a different command ID (CID), as well as a (shared or exclusive) hold request without a CID, is counted against the NISNHQ limit.

Specific Product Recommendations

Consider adjusting the value of this parameter when the SLOG file is defined for use on the Event Replicator Server, causing replication data to be written and read from the SLOG file.

Overriding the Parameter Setting

You can override this parameter setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.

Examples

In the following example, the maximum number of records that may be in hold status for a single user is set at 100.

```
ADARUN PROG=ADANUC,NISNHQ=100,NH=450...
```

In the following example, NISNHQ is not specified. The maximum number of records that may be in hold status for a single user is 80 (the NISNHQ default of `(NH / NU) * 2`, which is greater than the alternative default of 20).

```
ADARUN PROG=ADANUC,NU=50,NH=2000...
```

NONDES Parameter: Nondescriptor Searches

Parameter	Specify . . .	Possible Values	Default
NONDES	whether to allow (YES) or disallow (NO) searches using nondescriptors in search buffers.	YES | NO	YES

By default, you can perform searches on Adabas fields that have not been defined as descriptors. However, nondescriptor searches carry significant overhead.

NONDES=NO prevents searches on nondescriptors. An attempt to search on a nondescriptor field produces response code 68 (ADARSP068).

Note: The behavior of nondescriptor searches in Adabas databases differs between mainframe and open systems in regards to null suppression in the fields. In open systems, nondescriptor searches do not return records with null values in a field if the field is null-suppressed (NU); on mainframe systems, the null-suppression (NU) of fields is ignored during nondescriptor searches. At this time, to resolve this problem, we recommend that you remove the null suppression option (NU) for open systems fields, if the fields must be used for a nondescriptor search.

Example

The following example disallows searches using nondescriptors.

```
ADARUN PROG=ADANUC,NONDES=NO
```

NPLOG Parameter: Number of Protection Log Data Sets

Parameter	Specify . . .	Minimum	Maximum	Default
NPLOG	the maximum number of protection log data sets to be opened and logged in the PPT.	0, 2	8	0

This parameter specifies the maximum number of protection log data sets to be opened and logged in the parallel participant table (PPT) for the current nucleus. Valid values are 0 (the default) and 2-8. Other values produce a parameter error. A zero value disables the parameter.

If more than the number of log data sets specified in the NPLOG parameter are allocated in the startup JCL, only the number specified in the NPLOG parameter are opened and logged; any additional allocated data sets are activated using the dynamic allocation process.

NPLOG must be specified with a nonzero value in conjunction with the PLOGDEV, PLOGSIZE, and UEX12 parameters.

NPLOG is not valid if DUALCLD, DUALCLS, DUALPLD, DUALPLS, or UEX2 are specified.

UEX2 and UEX12 are mutually exclusive.

Example

The following example runs the Adabas nucleus with six data protection log data sets and uses a model 3390 device. The size of each log data set is 1350 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3390 device has 15 tracks, and each track contains 9 blocks of protection log data. Thus, 10 cylinders = 10 * 15 * 9 = 1350 blocks).

```
ADARUN PROG=ADANUC,NPLOG=6,PLOGDEV=3390,PLOGSIZE=1350
```

NPLOGBUFFERS Parameter: PLOG Protection I/O Buffer Control

Parameter	Specify . . .	Minimum	Maximum	Default
NPLOGBUFFERS	the number of PLOG protection I/O buffers allocated in the Adabas nucleus session	1	2147483687	1

PLOG protection I/O buffers are only allocated when ADARUN parameter LFIOP is set to a nonzero value. The realistic maximum setting of the NPLOGBUFFERS parameter is limited by the available amount of virtual and real memory. The number of protection I/O buffers allocated is limited to the number of PLOG blocks available on the dual/OCTO PLOG data sets (see the related ADARUN parameters DUALPLS and PLOGSIZE).

To determine the setting for NPLOGBUFFERS, look at the related high-water mark shown in the output of the DRES operator command or in the nucleus shutdown statistics. If the related high-water mark hits 100%, consider increasing the size of NPLOGBUFFERS

The use of the NPLOGBUFFERS parameter may considerably improve performance in update intensive applications that were formerly constrained by writes to the PLOG. When NPLOGBUFFERS is greater than 1, multiple PLOG blocks may be written in one write I/O to the PLOG.

In environments running in 31-bit addressing mode, the PLOG protection I/O buffers are allocated above the 16-megabyte line.

In environments supporting 64-bit virtual storage, the PLOG protection I/O buffers are allocated above the 2 gigabyte bar when the V64BIT ADARUN parameter is set to "YES".

Example

The following example sets the number of PLOG protection I/O buffers to 5.

```
ADARUN PROG=ADANUC,NPLOGBUFFERS=5
```

NQCID Parameter: Number of Active Command IDs per User

Parameter	Specify . . .	Minimum	Maximum	Default
NQCID	the maximum number of active command IDs per user.	1	65535	20

The table of sequential commands contains the control blocks for sequential read operations. Each operation is identified by a command ID (CID). NQCID defines the total number of CIDs allowed per user, and therefore the maximum number of table elements the user has available at any point in time.

Note: The NQCID value specified does not ensure that all users have the specified count of CIDs available at all times. The actual number of CIDs available to a single user depends on the totals of table elements available and sequential operations active at a given time. Also note that NQCID does not pertain to internal CIDs.

Overriding the Parameter Setting

You can override this parameter setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.

Example

The following example sets the maximum number of parallel sequential processes per user to 25.

```
ADARUN PROG=ADANUC,NQCID=25,NU=200,LQ=3000
```

Since one element in the sequential command table occupies approximately 100 bytes (varies depending on the command type; see the description of the LQ parameter), NQCID=25 limits each user to approximately 2500 bytes of the table.

NSISN Parameter: Number of ISNs per ISN Table Element

Parameter	Specify . . .	Minimum	Maximum	Default
NSISN	the maximum number of ISNs per element in the table of ISNs (TBI).	7	see text	51

The parameter NSISN specifies the maximum number of ISNs kept in a TBI element in memory (list of resulting ISN lists). The more ISNs that are kept in memory, the fewer the Work I/Os needed to read additional resulting ISN lists from the Work data set. This parameter influences the performance of the L1/4 command with the N (GET NEXT) option.

The highest value that can be specified for NSISN depends on the Work device. For Adabas-specified Work block sizes, the maximum values are shown in the device tables provided later in this section. However, the tables do not apply to *user-defined* Work block sizes; in this case, the maximum is determined as follows, where WORK is the block size of the Work data set:

- (WORK - 6) / 4
- The TBI element has a length of 64 + (NSISN * 4) bytes in the TBI pool (see the LI parameter).

IBM and Compatible Devices

Device Type	Max. Value
3380	1371
3390	1429
8380	2267
8381	2867
8385	5865
8390	2674
8391	3418
8392	4611
8393	6995

Overriding the Parameter Setting

You can override this parameter setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.

Example

In the following example, the maximum number of ISNs per TBI element during the Adabas session is set to 400.

```
ADARUN PROG=ADANUC,NSISN=400
```

NT Parameter: Number of Threads

Parameter	Specify . . .	Minimum	Maximum	Default
NT	the number of user threads to be used during the Adabas session.	4	250	5

A thread is used to execute a single command for any user (access or update). Each thread requires 8192 bytes.

In environments running in 31-bit addressing mode, the threads are allocated above the 16-megabyte line.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

For Event Replicator Server databases, this parameter must be greater than or equal to 15.

Specific Product Recommendations

- For Event Replicator Server databases, the NT parameter must be greater than or equal to 15.
- If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.
- For an Adabas Audit Server, the NT parameter must be greater than or equal to 15.

Example

The following example runs the Adabas nucleus with six (6) user threads.

```
ADARUN PROG=ADANUC,NT=6
```

NU Parameter: Number of User Queue Elements

Parameter	Specify . . .	Minimum	Maximum	Default
NU	the maximum number of user queue elements.	20	16777215	200

Adabas assigns a user queue element (UQE) to each active user when the user issues an OP command or (if the user did not issue an OP command) the first Adabas command.

Adabas releases the UQE when the user issues a CL command or when there is a timeout for an ET logic user who did not specify a restart ID in the control block (Additions 1 field) for the OP command.

Currently, 272 * (NU + 4) bytes are allocated to the user queue. The length of each UQE is 288 bytes.

In environments running in 31-bit addressing mode, the user queue is allocated above the 16-megabyte line.

Because each user's file list can be a different size, the number of users who can be active at the same time may be less than the NU value. Therefore, you may need to specify an NU value that is greater than the maximum number of users desired.

The Adabas session statistics or Adabas Online System can be used to tune this parameter for the next session.

Specific Product Recommendations

If the Event Replicator Server is set to support updates by multiple concurrent users to Adabas targets (when the NPADACALLS initialization parameter is set to any value greater than "1"), consider adjusting the value of this parameter in the target Adabas nucleus to ensure the target nucleus can handle updates from multiple concurrent users.

Example

The following example runs the Adabas nucleus with a maximum of 500 elements in the user queue.

```
ADARUN PROG=ADANUC,NU=500
```

NUCID Parameter: Adabas Cluster Nucleus ID

Parameter	Specify . . .	Possible Values	Default
NUCID	the ID for a nucleus in an Adabas cluster.	0-65000	0

NUCID identifies an Adabas cluster nucleus. The cluster may span operating system images under Adabas Cluster Services, or it may be confined to a single operating system image under Adabas Parallel Services. A value greater than 0 (the default) identifies this nucleus as an Adabas cluster nucleus.

Value	Meaning
0	The default setting. This is not an Adabas cluster nucleus.
1-65000	This is an Adabas cluster nucleus. This value must be unique among all target IDs declared to the Adabas router (ADASVC): Adabas databases (DBID), cluster nuclei (NUCID), Natural buffer pools and other MPM servers, Entire Net-Work, and internal targets used by Adabas Cluster Services and Adabas Parallel Services (restricted range 65001-65479). Nucleus IDs for the same cluster need not be specified contiguously.

Guidelines

- Each nucleus being run must have its own Work data set.
- All nuclei in a cluster must be run against the same database.
- we strongly recommend that you run utilities or user applications against a nucleus in the local z/OS image. Although a remote open nucleus may be used, performance degradation due to network overhead is likely.

Example

The following example starts an Adabas cluster session with the specified cluster nuclei:

```
ADARUN PROG=ADANUC,NUCID=1, ...
```

NWORK1BUFFERS Parameter: Work Part 1 Protection I/O Buffer Control

Parameter	Specify . . .	Minimum	Maximum	Default
NWORK1BUFFERS	the number of Work part 1 protection I/O buffers allocated in the Adabas nucleus session.	1	2147483687	1

Work part 1 protection I/O buffers are only allocated when ADARUN parameter LFIOP is set to a nonzero value. The realistic maximum setting of the NWORK1BUFFERS parameter is limited by the available amount of virtual and real memory. The number of protection I/O buffers allocated is limited to the number of Work part 1 blocks available (see the related ADARUN LP parameter).

To determine the setting for NWORK1BUFFERS, look at the related high-water mark shown in the output of the DRES operator command or in the nucleus shutdown statistics. If the related high-water mark hits 100%, consider increasing the size of NWORK1BUFFERS

The use of the NWORK1BUFFERS parameter may considerably improve performance in update intensive applications that were formerly constrained by writes to Work part 1. When NWORK1BUFFERS is greater than 1, multiple Work part 1 blocks may be written in one write I/O to the Work data set.

In environments running in 31-bit addressing mode, the Work part 1 protection I/O buffers are allocated above the 16-megabyte line.

In environments supporting 64-bit virtual storage, the Work part 1 protection I/O buffers are allocated above the 2 gigabyte bar when the V64BIT ADARUN parameter is set to "YES".

Example

The following example sets the number of Work part 1 protection I/O buffers to 5.

```
ADARUN PROG=ADANUC,NWORK1BUFFERS=5
```

OPENRQ Parameter: Open Command Required

Parameter	Specify . . .	Possible Values	Default
OPENRQ	whether an open command is required.	YES | NO	YES

This parameter may be used to indicate whether an open (OP) command must be issued as the first command of a user session.

Value	Meaning
YES	The default setting. Any user attempting to issue commands to Adabas without previously issuing an OP command receives a response code.
NO	Users are not required to issue an OP as the first command of the session.

Example

The following example specifies that the first session command issued by users does not have to be an OP command.

```
ADARUN PROG=ADANUC,OPENRQ=NO
```

PGFIX Parameter: EXCPVR Page Fixing Control

Parameter	Specify . . .	Possible Values	Default
PGFIX	whether pages containing I/O control blocks are released after I/O processing is completed or after the job has ended when using EXCPVR.	YES | NO	NO

This parameter is only valid for users using EXCPVR (for example, when running the Adabas nucleus or utility jobs APF-authorized). It allows you to control how page-fixing occurs when EXCPVR is in use. When PGFIX=NO is specified, pages containing the I/O control blocks are fixed only for the duration of the I/O processing. When PGFIX=YES is specified, pages containing the I/O control blocks are fixed for the duration of the job.

PLOGDEV/PLOGSIZE Parameter: Multiple Protection Log Device Type / Size

Parameter	Specify . . .	Possible Values	Default
PLOGDEV	the device type to be used for multiple protection logs.	see text	value of DEVICE parameter
PLOGSIZE	the number of blocks available for each protection log data set.	Min 16, Max 16777215	none

The PLOGDEV and PLOGSIZE parameters specify the type of external storage device used for multiple data protection log data sets and the number of blocks available for each log data set. These parameters must be used *together*; if you specify one parameter, you must also specify the other parameter.

You must also specify the NPLOG parameter to use multiple protection logs. If you specify a user exit, it must by UEX12. See ***Adabas Restart and Recovery***, elsewhere in this guide, for information about multiple data set protection logging.

The Adabas Installation documentation specifies the device types that Adabas supports.

All protection log data sets must be formatted using the ADAFRM utility before they are used. The appropriate JCL for the data sets is required.

Example

The following example runs the Adabas nucleus with six (6) data protection log data sets and uses a model 3390 device. The size of each log data set is 1350 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3390 device has 15 tracks, and each track contains 9 blocks of protection log data. Thus, 10 cylinders = 10 * 15 * 9 = 1350 blocks).

```
ADARUN PROG=ADANUC,NPLOG=6,PLOGDEV=3390,PLOGSIZE=1350
```

PLOGRQ Parameter: Protection Log Required

Parameter	Specify . . .	Possible Values	Default
PLOGRQ	whether a protection log is required for this session.	YES | NO | SEL | FORCE	YES

The valid values are as follows:

Value	Meaning
YES	The default setting. Any attempt to start an Adabas nucleus without a protection log causes the Adabas initialization to terminate with an error message.
NO	In this Adabas session: ■ multiple PLOG data sets are used if a PLOGSIZE parameter is provided; ■ dual PLOG data sets are used if a DUALPLS parameter is provided; ■ a SIBA data set is used if JCL for a SIBA data set is provided; ■ otherwise, no protection log data set is used.
SEL	Selective protection logging is switched on only when running an ADASAV SAVE (database) or SAVE FILE= ... utility operation with an active nucleus. Protection logging is switched off when the ADASAV SAVE operation completes successfully.
FORCE	Any attempt to start an Adabas nucleus without UEX2 or UEX12, or without a protection log causes the Adabas initialization to terminate with an error message. Additionally, an error occurs if the initialization process determines that different PLOG data sets have been specified for the session and the old PLOG data sets have not yet been copied.

Example

The following example indicates that a data protection log (SIBA or PLOG) is not mandatory during the Adabas session but will be used if provided.

```
ADARUN PROG=ADANUC,PLOGRQ=NO
```

PREFETCH Parameter: Activate Prefetch Feature

Parameter	Specify . . .	Possible Values	Default
PREFETCH	whether to enable a prefetch option automatically.	YES | OLD | NO	NO

This parameter determines whether the Adabas prefetch feature is enabled automatically; if it is, execution is transparent to the user program.

There are two Adabas prefetch options: the multifetch option is supported across Adabas platforms; the prefetch option is available only on mainframe Adabas. For information about these two options, see the *Adabas Command Reference Guide*.

Records can be prefetched and stored in a prefetch buffer for the following commands:

- L1/L4 with I or N option (read by ISN, find with GETNEXT)
- L2/L5 (read physical)
- L3/L6 (read logical by descriptor)
- L9 (histogram)

Sequential commands of the types listed above can retrieve the next record from the prefetch buffer, which saves cross-region/cross-network calls to the nucleus.

The PREFETCH parameter has the following values:

Value	Meaning
YES	Adabas enables the command-level multifetch (M) option.
OLD	Adabas enables the command-level prefetch (P) option.
NO	Adabas does not enable the prefetch feature.

Note: Specify PREFETCH=YES or PREFETCH=OLD only when MODE=MULTI (multiuser mode). Do not specify these settings when running Adabas utilities. Dynamic reversal of read direction for L3/L6/L9 is not supported with these settings.

Examples

The following example runs a user program using multifetch.

```
ADARUN PROG=USER,PREFETCH=YES
```

The following example runs a user program using prefetch.

```
ADARUN PROG=USER,PREFETCH=OLD
```

PREFICMD/PREFIFIL Parameter: Include Commands/Files in Prefetch/Multi-fetch Processing

Parameter	Specify . . .	Possible Values	Default
PREFICMD	the commands to include in prefetch processing.	L1-L9	none
PREFIFIL	the files to include in prefetch processing.	1-5000	none

Together, PREFIFIL and PREFICMD specify one or more files to be included in prefetch/multifetch processing. PREFIFIL specifies one or more files by file number; PREFICMD specifies a command for each file number.

You can either include list of files using PREFIFIL/PREFICMD or exclude lists of files using the parameters PREFXFIL/PREFXCMD), depending on which list is shorter. You cannot both include and exclude files at the same time.

If prefetching/multifetching has been specified and

- files/commands have been neither included nor excluded, the prefetching/multifetching applies to all files by default.
- files/commands have been included using the ADARUN parameters PREFIFIL/PREFICMD, the prefetching/multifetching applies only to the files on the inclusion list.
- both PREFIFIL/PREFICMD and PREFXFIL/PREFXCMD are specified, Adabas recognizes the exclusion parameters PREFXFIL/PREFXCMD and ignores the inclusion parameters PREFIFIL/PREFICMD.

Multiple Inclusions

Specify multiple file numbers in multiple PREFIFIL statements or in a list, separated by commas. Specify multiple commands in multiple PREFICMD statements or in a list, separated by slashes (/). The maximum number of entries for each is 50.

If you use lists, they must have the same number of entries. There is a one-to-one correspondence, by position, between a file number in the PREFIFIL list and a command in the PREFICMD list.

Eligible Commands and Command Pairs

The following commands can be included in prefetch/multifetch processing for a file:

- L1/L4
- L2/L5

- L3/L6
- L9

If either command in one of the pairs above is specified in a PREFICMD statement, *both* commands are included. For example, if PREFICMD=L1 is specified, both L1 and L4 are included.

Examples

The following example includes the L3 command for file 8 in prefetch processing.

```
ADARUN PREFIFIL=8,PREFICMD=L3
```

The following example includes the L3 command for file 8 and the L3 and L9 commands for file 12 in prefetch/multifetch processing. As this example shows, you can specify the same file number several times, with a different command each time. Likewise, you can specify the same command code several times, with a different file number each time.

```
ADARUN PREFIFIL=8,12,12,PREFICMD=L3/L3/L9
```

or

```
ADARUN PREFIFIL=8,PREFICMD=L3
  ADARUN PREFIFIL=12,PREFICMD=L3
  ADARUN PREFIFIL=12,PREFICMD=L9
```

The following example executes a user program with multifetch and allocates 20000 bytes for the total prefetch buffer area. It includes the L3 and L6 commands for files 8 and 12 in multifetch processing and it includes *all* commands for file 7. No PREFSBL value is specified, so the buffer length for a single prefetch sequence defaults to 3000 bytes; up to 6 prefetch sequences can be handled.

```
ADARUN PROG=USER,PREFETCH=YES,PREFTBL=20000
  ADARUN PREFIFIL=8,12,PREFICMD=L3/L3
  ADARUN PREFIFIL=7,7,7,7,PREFICMD=L1/L2/L3/L9
```

PREFNREC Parameter: Multifetch Record Count

Parameter	Specify . . .	Minimum	Maximum	Default
PREFNREC	the maximum number of records that can be prefetched.	0	see text	0

This parameter specifies the maximum number of records that the multifetch feature (specified by the PREFETCH=YES parameter) obtains.

In mode PREFETCH=YES the maximum that can be specified depends on the maximum that can be held by the record and multifetch buffers.

In mode PREFETCH=OLD the maximum that can be specified depends on the maximum that can be held by the record and ISN buffers, each of which is assumed to be less than 32 kilobytes.

If no value or zero (the default) is specified, the sizes of the ISN and record buffers determine the number of records that are prefetched. Specifying zero (0) *does not* disable multifetch operation.

Example

The following example enables the multifetch feature and allows a maximum of 32 records to be prefetched.

```
ADARUN PROG=USER,PREFETCH=YES,PREFNREC=32
```

PREFSBL Parameter: Prefetch Buffer Length per Call

Parameter	Specify . . .	Minimum	Maximum	Default
PREFSBL	the prefetch/multifetch buffer length (in bytes) for one prefetch.	1024	<PREFTBL value>	3000

The PREFSBL parameter specifies the size of an internal buffer that the multifetch/prefetch feature uses to prefetch data for a single sequential read command. By setting of this buffer size you control the maximum amount record data that is prefetched. See also the description of the **PREFTBL parameter**, which specifies the total buffer space used by prefetch/multifetch.

When prefetch or multifetch is active, Adabas makes a buffer available to the user program for storing prefetched records on the client side. If a required record is already in a prefetch buffer, the record is taken directly from the buffer.

Note: Do not specify this operand unless PREFETCH=YES or PREFETCH=OLD also is specified.

The following formula can be used to estimate the approximate maximum number of records (n) that will be prefetched.

In mode PREFETCH=OLD:

```
600 + FBL + ((RBL * 3) * n) = PREFSBL
```

In this calculation, *FBL* is the length of the format buffer and *RBL* is the length of the record buffer.

The 32K buffer size limit will limit the maximum number of prefetched records to:

```
MAX(n) = 32768 / RBL ↩
```

In mode PREFETCH=YES:

```
1400 + FBL + ((RBL * 3) * n) = PREFSBL
```

In this calculation, FBL is the length of the format buffer and RBL is the length of the record buffer.

For long sequential processes, this parameter should be as large as possible unless the sequential processes are frequently interrupted with value start (L3/L6) or RC commands; in that case, a large value often causes records to be prefetched needlessly.

Example

The following example executes a user program using multifetch and uses a 15000-byte prefetch buffer.

```
ADARUN PROG=USER,PREFETCH=YES,PREFSBL=15000
```

PREFSTDD Parameter: Multifetch Statistics DD Name

Parameter	Specify . . .	Valid Values	Default
PREFSTDD	the job control statement label (DD name) in the JCL that defines the location to which multifetch statistics should be written when a CL command is issued.	DD name or other job statement label	none

The PREFSTDD parameter specifies the job control statement label (DD name) in the nucleus startup job that defines the location to which multifetch statistics should be written when multifetch statistics are printed. The length of the job statement label (DD name) is limited to seven (7) characters.

When PREFSTDD is specified, the ADARUN PREFETCH parameter must be set to "YES".

Note: Any data set defined by the job statement label referenced by PREFSTDD must have an LRECL of at least 80 bytes and must be a fixed block data set. If it does not, the PREFSTDD setting is ignored and not statistics are printed.

Multifetch statistics are written when a CL command is issued. If no CL command is issued, no multifetch statistics are produced. If multiple CL commands are issued, multiple multifetch statistics are produced. The first multifetch statistics contains the values cumulated since start of job. The second and following multifetch statistics contain the values cumulated since start of job and the interval from the last Close command.

Example 1

The following example executes a user program using multifetch, uses a 15,000-byte prefetch buffer, and writes multifetch statistics to the output location identified by the STATDD job control statement:

```
ADARUN PROG=USER,PREFETCH=YES,PREFSBL=15000,PREFSTDD=STATDD
```

The STATDD job control statement that accompanies these ADARUN parameters might simply indicate that multifetch statistics should be written to SYSOUT:

```
//STATDD    DD SYSOUT=*
```

Alternatively, the STATDD job control statement that accompanies these ADARUN parameters might specify a specific data set:

```
//STATDD    DD DISP=(NEW,CATLG),DSN=ADABAS.MLF.STATS,
//             UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE),
//             DCB=(RECFM=FB,LRECL=121,BLKSIZE=1210)
```

The following is an example of the multifetch statistics output produced:

```
ADAML1 +-------------------------------------------------------+
ADAML1 +
ADAML1 + A D A M L F   V833   S T A T I S T I C S
ADAML1 +
ADAML1 + Cumulated since start of Job
ADAML1 +-------------------------------------
ADAML1 + Percentage of PREFTBL used          :        50.00
ADAML1 + Number of calls to ADAMLF           :         2007
ADAML1 + Number of cmds to ADABAS            :           43
ADAML1 + Number of cmds returned from ADAMLF :         1998
ADAML1 + Number of over reads                :            0
ADAML1 +
ADAML1 +-------------------------------------------------------+
```

Example 2

The following example executes a user program issuing three Close commands, using multifetch, uses a 15,000-byte prefetch buffer, and writes multifetch statistics to the output location identified by the STATDD job control statement (like Example 1):

```
ADARUN PROG=USER,PREFETCH=YES,PREFSBL=15000,PREFSTDD=STATDD
```

Note: When the STATDD job control statement specifies a data set, it must be declared with the DISP=MOD parameter as shown in the following syntax example.

```
//STATDD    DD DISP=MOD,DSN=ADABAS.MLF.STATS
```

The following is an example of the multifetch statistics output produced:

```
ADAML1 +-------------------------------------------------------+
ADAML1 +
ADAML1 + A D A M L F   V833   S T A T I S T I C S
ADAML1 +
ADAML1 + Cumulated since start of Job
ADAML1 +--------------------------------------
ADAML1 + Percentage of PREFTBL used          :        50.00
ADAML1 + Number of calls to ADAMLF           :         2007
ADAML1 + Number of cmds to ADABAS            :           43
ADAML1 + Number of cmds returned from ADAMLF :         1998
ADAML1 + Number of over reads                :            0
ADAML1 +
ADAML1 +-------------------------------------------------------+
ADAML1 +-------------------------------------------------------+
ADAML1 +
ADAML1 + A D A M L F   V833   S T A T I S T I C S
ADAML1 +
ADAML1 + Cumulated since start of Job
ADAML1 +--------------------------------------
ADAML1 + Percentage of PREFTBL used          :        50.00
ADAML1 + Number of calls to ADAMLF           :         4015
ADAML1 + Number of cmds to ADABAS            :           86
ADAML1 + Number of cmds returned from ADAMLF :         3996
ADAML1 + Number of over reads                :            0
ADAML1 +
ADAML1 + Cumulated since last Close cmd (1)
ADAML1 +--------------------------------------
ADAML1 + Percentage of PREFTBL used          :        50.00
ADAML1 + Number of calls to ADAMLF           :         2008
ADAML1 + Number of cmds to ADABAS            :           43
ADAML1 + Number of cmds returned from ADAMLF :         1998
ADAML1 + Number of over reads                :            0
ADAML1 +
ADAML1 +-------------------------------------------------------+
ADAML1 +-------------------------------------------------------+
ADAML1 +
ADAML1 + A D A M L F   V833   S T A T I S T I C S
ADAML1 +
ADAML1 + Cumulated since start of Job
ADAML1 +--------------------------------------
ADAML1 + Percentage of PREFTBL used          :        50.00
ADAML1 + Number of calls to ADAMLF           :         6023
ADAML1 + Number of cmds to ADABAS            :          122
ADAML1 + Number of cmds returned from ADAMLF :         5994
ADAML1 + Number of over reads                :            0
ADAML1 +
ADAML1 + Cumulated since last Close cmd (2)
ADAML1 +--------------------------------------
ADAML1 + Percentage of PREFTBL used          :        50.00
ADAML1 + Number of calls to ADAMLF           :         2008
ADAML1 + Number of cmds to ADABAS            :           36
ADAML1 + Number of cmds returned from ADAMLF :         1998
```

```
ADAML1 + Number of over reads                :            0
ADAML1 +
ADAML1 +-------------------------------------------------------+
```

The following table describes these statistics:

Statistic	Meaning
Percentage of PREFTBL used	The percentage of the ADARUN parameter PREFTBL used for this execution.
Number of calls to ADAMLF	The number of times that ADAMLF was entered from the client program.
Number of cmds to ADABAS	The total number of commands that were passed to Adabas for this execution.
Number of cmds returned from ADAMLF	The number of commands that were returned from the ADAMLF buffers.
Number of over reads	The number of records that were prefetched that were not passed back to the caller.

The version is adapted with every change of the release.

PREFTBL Parameter: Total Size of Prefetch Buffer

Parameter	Specify . . .	Minimum	Maximum	Default
PREFTBL	the size of the total prefetch buffer area.	6144	2147483647	30000

PREFTBL specifies the size of the total prefetch buffer area for this operation. This buffer length should be large enough to hold the PREFSBL-specified value, multiplied by the number of command-ID-controlled sequential prefetches for this operation. This allows a user program to perform multiple sequential operations requiring prefetches, each with a prefetch buffer of the size specified in the PREFSBL parameter.

If this user program has a single prefetch sequence only, the PREFTBL value must be equal to or greater than the PREFSBL value. The PREFTBL-specified space is always allocated to the program.

Note: Do not specify this operand unless PREFETCH=YES or PREFETCH=OLD is also specified.

Example

The following example executes a user program using multifetch. No PREFSBL value is specified, so this example uses the default buffer length for a single prefetch (3,000 bytes). It also allocates 30000 bytes for the total prefetch buffer area; this size provides buffer space for up to 10 prefetch sequences.

```
ADARUN PROG=USER,PREFETCH=YES,PREFTBL=30000
```

PREFXCMD (PREFXFIL) Parameters: Exclude Commands (Files) from Prefetch/Multifetch Processing

Parameter	Specify . . .	Possible Values	Default
PREFXCMD	the commands to exclude from prefetch processing.	L1-L9	none
PREFXFIL	the files to exclude from prefetch processing.	1-5000	none

Together, PREFXFIL and PREFXCMD specify one or more files to be excluded from prefetch/multifetch processing. PREFXFIL specifies one or more files by file number; PREFXCMD specifies a command for each file number.

You can either include list of files using PREFIFIL/PREFICMD) or exclude lists of files using the parameters PREFXFIL/PREFXCMD, depending on which list is shorter. You cannot both include and exclude files at the same time.

If prefetching/multifetching has been specified and

- files/commands have been neither included nor excluded, the prefetching/multifetching applies to all files by default.
- files/commands have been excluded using PREFXFIL/PREFXCMD, the prefetching/ multifetching applies only to the files on the exclusion list.
- both PREFIFIL/PREFICMD and PREFXFIL/PREFXCMD are specified, Adabas recognizes the exclusion parameters PREFXFIL/PREFXCMD and ignores the inclusion parameters PREFI-FIL/PREFICMD.

Multiple Exclusions

Specify multiple file numbers in multiple PREFXFIL statements or in a list, separated by commas. Specify multiple commands in multiple PREFXCMD statements or in a list, separated by slashes (/). The maximum number of entries for each is 50.

If you use lists, they must have the same number of entries. There is a one-to-one correspondence, by position, between a file number in the PREFXFIL list and a command in the PREFXCMD list.

Eligible Commands and Command Pairs

The following commands can be excluded from prefetch/multifetch processing for a file:

- L1/L4
- L2/L5

- L3/L6
- L9

If either command in one of the pairs above is specified in a PREFXCMD statement, *both* commands are excluded. For example, if PREFXCMD=L1 is specified, both L1 and L4 are excluded.

Examples

The following example excludes the L3 command for file 8 from prefetch processing.

```
ADARUN PREFXFIL=8,PREFXCMD=L3
```

The following example excludes the L3 command for file 8 and the L3 and L9 commands for file 12 from prefetch/multifetch processing. As this example shows, you can specify the same file number several times, with a different command each time. Likewise, you can specify the same command code several times, with a different file number each time.

```
ADARUN PREFXFIL=8,12,12,PREFXCMD=L3/L3/L9
```

or

```
ADARUN PREFXFIL=8,PREFXCMD=L3
  ADARUN PREFXFIL=12,PREFXCMD=L3
  ADARUN PREFXFIL=12,PREFXCMD=L9
```

The following example executes a user program with multifetch and allocates 20000 bytes for the total prefetch buffer area. It excludes the L3 and L6 commands for files 8 and 12 from multifetch processing and it excludes *all* commands for file 7. No PREFSBL value is specified, so the buffer length for a single prefetch sequence defaults to 3000 bytes; up to 6 prefetch sequences can be handled.

```
ADARUN PROG=USER,PREFETCH=YES,PREFTBL=20000
  ADARUN PREFXFIL=8,12,PREFXCMD=L3/L3
  ADARUN PREFXFIL=7,7,7,7,PREFXCMD=L1/L2/L3/L9
```

PROGRAM Parameter: Program to Run

Parameter	Specify:	Possible Values	Default
PROGRAM	the program to be executed.	see table below	USER

This parameter specifies what to execute. The possible values are described in the following table:

Specify:	To start:
ADACOM	an ADACOM task (used in Adabas Cluster Services and Adabas Parallel Services environments) For more information, refer to your Adabas Cluster Services and Adabas Parallel Services documentation.
ADANUC	an Adabas nucleus For more information about executing an Adabas nucleus, read *Adabas Session Execution*, in the *Adabas Operations Manual*.
ADAREV	an Adabas Review hub. Specify this in conjunction with the ADARUN `REVIEW` parameter. For more information, refer to your Adabas Review documentation.
NETWRK	an Entire Net-Work node For more information, refer to your Entire Net-Work documentation.
RENTUSER	a user program to be run using a reentrant Adabas batch/TSO link routine. For more information, refer to description of the Adabas TP monitor installation in your Adabas installation documentation.
USER	a user program to be run using a non-reentrant Adabas batch/TSO link routine. For more information, read *Linking Applications to Adabas*, in the *Adabas Operations Manual*
`utility-name`	an Adabas utility Specify an Adabas utility for *`utility-name`*. For more information, refer to the *Adabas Utilities Manual*.

Examples

The following example specifies that an Adabas nucleus is running.

```
ADARUN PROGRAM=ADANUC
```

The following example specifies that an Adabas Review hub is running.

```
ADARUN PROGRAM=ADAREV, REVIEW=202
```

The following example specifies that an Entire Net-Work node is running.

```
ADARUN PROGRAM=NETWRK
```

QBLKSIZE Parameter: Block Size for Sequential Data Sets

Parameter	Specify . . .	Minimum	Maximum	Default
QBLKSIZE	the block size for sequential data sets used by ADAIOR.	0	262144	0 (see text)

QBLKSIZE specifies the block size, in bytes, for the *sequential data sets* used by ADAIOR, the Adabas I/O component. Use this parameter to override the default block sizes. For more information about sequential files used by Adabas utilities and their operating system dependencies, read *Adabas Sequential Files* in the *Adabas Utilities Manual*.

Caution: If you choose to write tape files with large block sizes, for example for database backups, these files will not be transportable to systems where support for large blocks is not available. This might include a site being used as a backup facility for disaster recovery.

If QBLKSIZE=0 (the default), the following default block sizes are used:

Data Set	Block Size
Tape	the optimum selected for the device by the operating system
ECKD-Disk	track-size / k, where k = smallest value such that track-size / k <= 32760

For ECKD disks, the block size selected will typically result in 2 blocks per track.

This block size may be provided in the JCL or data set label.

Examples

For example, a block size of 65536 bytes for the Adabas sequential data sets is used by the ADAULD utility:

```
ADARUN PROG=ADAULD,QBLKSIZE=65536
```

In the following example, a block size of 32760 bytes for the Adabas sequential data sets is used by the ADAORD utility:

```
ADARUN PROG=ADAORD,QBLKSIZE=32760
```

READONLY Parameter: Read-Only Session Control

Parameter	Specify . . .	Possible Values	Default
READONLY	whether to permit database updates during this session.	YES | NO	NO

READONLY determines whether the Adabas session is a read-only or a read/write session.

Value	Meaning
YES	All database update commands issued by this nucleus are rejected.
NO	The default setting. The session allows both reading and updating of the database.

Guidelines

- When executing the ADAREP (database report) utility, you can specify READONLY=YES to issue an access OPEN instead of an update OPEN.
- When using READONLY=YES, ensure that the Work data set does not contain a pending autorestart; otherwise, the nucleus cannot start.
- Each nucleus being run must have its own Work data set.
- When a nucleus is started with READONLY=YES, it is *not* possible to switch the nucleus to READONLY=NO mode later in the session. When a nucleus is started with READONLY=NO, it *can* be switched back and forth between the READONLY=YES and READONLY=NO modes.
- A database save performed while the nucleus is in read-only mode is equivalent to an offline save (which is taken while the nucleus is down). You can avoid shutting down the nucleus for an offline save by switching the nucleus into read-only mode before the save and back to update mode after the save.

Example

The following example starts an Adabas session as read-only.

```
ADARUN PROG=ADANUC,READONLY=YES
```

REFSTPRT Parameter: Print Statistics before Refresh Statistics

Parameter	Specify . . .	Possible Values	Default
REFSTPRT	whether statistics should be printed before they are refreshed by an ADADBS REFRESHSTATS utility function run or before interval statistics have been refreshed using Adabas Online System (option **R** in **Session Monitoring**).	YES | NO	YES

REFSTPRT determines whether statistics are printed before they are refreshed.

Value	Meaning
YES	Adabas prints the statistics before an ADADBS REFRESHSTATS utility function run or before interval statistics have been refreshed using Adabas Online System. This is the default.
NO	Adabas does *not* print the statistics before an ADADBS REFRESHSTATS utility function run or before interval statistics have been refreshed using Adabas Online System.

Example

The following example indicates that Adabas should not print the statistics before an ADADBS REFRESHSTATS utility function run or before interval statistics have been refreshed using Adabas Online System.

```
ADARUN PROG=ADANUC,REFSTPRT=NO
```

REPLICATION Parameter: Replication Control

Parameter	Specify . . .	Possible Values	Default
REPLICATION	Whether to activate replication for the Adabas nucleus.	YES | NO	NO

Valid values are described in the following table.

Value	Meaning
YES	Activate replication for the Adabas nucleus.
NO	The default setting. Do not activate replication for the Adabas nucleus. When REPLICATION=NO is specified and there are replicated files on the database, replication will be turned off for all replicated files. A message is issued during nucleus initialization indicating which files were defined with replication.

The REPLICATION parameter should not be specified for the Event Replicator Server. REPLICATION=NO must be explicitly specified if:

- You want to start the Adabas nucleus without replication active, *and*
- `REPLICATION=YES` was specified in the previous Adabas nucleus session, *and*
- Replication has been defined for one or more files on the database.

`REPLICATION=NO` must be explicitly specified for utility execution if:

- A utility requiring an active nucleus is executed with `MODE=SINGLE`, *and*
- `REPLICATION=YES` was specified in the previous Adabas nucleus session, *and*
- Replication has been defined for one or more files on the database.

Note: The replication pool will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters REPLICATION=YES and V64BIT=YES.

Example

The following example activates replication for an Adabas session.

```
ADARUN PROG=ADANUC,REPLICATION=YES
```

REVBUFMB Parameter: Buffer Pool Size in Megabytes

Parameter	Use	Possible Values	Default
REVBUFMB	Specify the size of the Review Buffer Pool used as temporary storage when sending the CLOG records via Network (ADALNK) to the Review Hub.	1-1024	10

Use the `REVBUFMB` parameter only when `REVSTHMV` is set to `V53` and `REVLOGNR` is set to a value higher than `0`.

The `REVBUFMB` parameter may be activated during the Adabas nucleus session.

The possible values for this parameter are:

Value	Meaning
N	A temporary buffer with a size of *N* megabytes is allocated.

Example

The following example temporarily saves CLOG records in a buffer pool with a size of 32MB and then sends the records in packages of maximum 12 records via Network to the Review hub.

```
ADARUN PROG=ADANUC,REVSTHMV=V53,REVLOGNR=12,REVBUFMB=32
```

REVFILTER Parameter: Review Record Filtering Control

Parameter	Specify . . .	Possible Values	Default
REVFILTER	whether to allow Adabas Review record filtering during the session.	YES | NO	YES

REVFILTER determines whether command log record filtering may be activated. Filtering can decrease the number of command log records passed to Review for report processing.

Value	Meaning
YES	The default setting. Database command log records may be filtered from report processing, depending upon Review report rules.
NO	All command log records will be passed to Review for report processing.

Examples

In the following example, Adabas Review's record filtering may be activated during the Adabas nucleus session.

```
ADARUN PROG=ADANUC,REVFILTER=YES
```

In the following example, Adabas Review's record filtering will not be in effect for the Adabas nucleus session.

```
ADARUN PROG=ADANUC,REVFILTER=NO
```

REVIEW Parameter: Adabas Review Control

Parameter	Specify . . .	Possible Values	Default
REVIEW	whether to run Adabas Review in local or hub mode specifying the hub ID, or not at all.	NO | LOCAL | dbid	NO

Note: The parameter name REVIEWHUBID is a synonym for REVIEW, provided to ensure downward compatibility with past Adabas releases. We recommend that you use the parameter name REVIEW instead, wherever possible.

REVIEW controls the use of the Adabas Review product:

Value	Meaning
NO	The default setting. Adabas Review is not started.
LOCAL	Adabas Review is started in local mode running as an extension to ADALOG. In local mode, Adabas Review job control statements should be added to the Adabas nucleus startup JCL. **Note:** If an Adabas Review load library is not included in the startup JCL, the REVIEW parameter is automatically changed from LOCAL to NO.
dbid	Adabas Review is started in hub mode. The physical database ID that you specify for the hub identifies ■ the hub (server) itself (with PROGRAM=ADAREV) that is being started; or ■ from an Adabas nucleus (client), the hub that is the target for Adabas Review processing for that nucleus (with PROGRAM=ADANUC). In hub mode, Adabas Review job control statements should be added to the Adabas Review hub startup JCL.

Dynamic Modification

The setting of the ADARUN PROG=ADANUC,REVIEW=dbid parameter can be changed dynamically using the REVIEWHUBID command from the operator console, the ADADBS OPERCOM REVIEWHUBID function, or the Modify Parameter function of Adabas Online System.

Note: The value of the REVIEW parameter is not changed in the Adabas ID table. A REVIEW hub coming up with TARGET=ALL checks the ID table but does not find the dynamic changes of the REVIEW parameter.

Examples

The following example starts hub 202 for the Adabas Review hub (server) installation.

```
ADARUN PROGRAM=ADAREV,REVIEW=202
```

The following example starts the Adabas nucleus that will log to Adabas Review hub 202 for the Adabas Review (client) installation.

```
ADARUN PROGRAM=ADANUC,REVIEW=202
```

REVLOGBMAX Parameter: Logged Buffer Size Limit for Review

Parameter	Use	Values	Default
REVLOGBMAX	Specify the maximum allowable number of bytes of a logged buffer for Review.	Any integer ranging from 0 to 30000	5120

If a buffer is longer than this value, the logged buffer is truncated from the point at which its size exceeds the setting of the REVLOGBMAX parameter. The REVLOGBMAX setting affects the ADARUN LOGGING parameter specifications only for both CLOGLAYOUT=8. The minimum value (368) is the size of the length of the CLOGLAYOUT=8 basic record plus the length of the extended Adabas control block (ACBX).

Example

The following example runs the Adabas nucleus using a logged buffer size limit of 512. Individual logged buffers, such as the format buffer for an Adabas command, will be truncated if they exceed 512 bytes.

```
ADARUN PROG=ADAREV,LOGBMAX=512
```

REVLOGMAX Parameter: Total Logged Buffer Size Limit for a Review Command

Parameter	Use	Values	Default
REVLOGMAX	Specify the maximum size of all of the logged buffers allowed for an Adabas Review command.	Any integer ranging from 2000 to 32764 (32K - 4).	16384

When the sum of sizes of the logged buffers for an Adabas Review command reaches the value of the REVLOGMAX parameter, the buffer exceeding the limit is truncated and all following buffers are omitted. The size of REVLOGMAX must at least be as large as the REVLOGBMAX + 2000 in order to accommodate the largest buffer of the Adabas command. The minimum value (2000) is the size of the length of the CLOGLAYOUT=8 basic record plus the length of the extended Adabas control block (ACBX) and the CLEX information.

The CLOGMAX setting affects the ADARUN LOGGING parameter specifications only for CLOGLAYOUT=8.

Example

In the following example, the sum of all logged buffers for an Adabas Review command to the Adabas nucleus cannot exceed 10000 bytes.

```
ADARUN PROG=ADAREV,REVLOGMAX=10000
```

REVLOGNR Parameter: Maximum Number of Records per Buffer

Parameter	Use	Possible Values	Default
REVLOGNR	Hub mode: Specify whether the CLOG records are sent as buffered records or as single records via Network (ADALNK) to the Review Hub.	0-64	0
	Local mode: Specify the CLOG record after which the REVIEW subtask is posted for processing.	0-64	16

Use the `REVLOGNR` parameter in Hub mode only when `REVSTHMV` is set to `V53`.

The `REVLOGNR` parameter may be activated during the Adabas nucleus session.

The possible values for this parameter in Hub mode are:

Value	Meaning
0	The CLOG records are sent as single records.
N, where 1<=N<=64	The CLOG records are buffered with a maximum of *N* records per buffer. The size of this buffer is 32K-1 bytes.

The possible values for this parameter in Local mode are:

Value	Meaning
0	*N* is set to the default (16).
N, where 1<=N<=64	After the *N*th CLOG record is sent, post the REVIEW subtask for processing.

Examples

The following example temporarily saves CLOG records in a buffer pool with a size of 32MB and then sends the records in packages of maximum 12 records via Network to the Review hub.

```
ADARUN PROG=ADANUC,REVSTHMV=V53,REVLOGNR=12,REVBUFMB=32
```

The following example sends single CLOG records via Network to the Review hub immediately after the records are created.

```
ADARUN PROG=ADANUC,REVSTHMV=V53,REVLOGNR=0
```

REVSTHMV Parameter: Send to HUB Mode Version

Parameter	Use	Possible Values	Default
REVSTHMV	Specify whether the ADARMT/ADRST interface or the ADARMX/ADARSX interface is used to send data via Network (ADALNK) to the Review Hub.	V52 V53	V52

The `REVSTHMV` parameter may be activated during the Adabas nucleus session.

The possible values for this parameter are:

Value	Meaning
V52	The CLOG records are sent via the ADARMT/ADARST interface.
V53	The CLOG records are sent via the ADARMX/ADARSX interface.

Examples

See the examples of the REVBUFMB, REVLOGNR and REVTIMER parameters.

REVTIMER Parameter: Send Interval in Seconds

Parameter	Use	Possible Values	Default
REVTIMER	Hub mode: Specify the largest delay for sending buffered CLOG records via Network (ADALNK) in case of low database activity.	1-60	1
	Local mode: Send the CLOG records to the Review hub after no later than *N* seconds.		

Use the `REVTIMER` parameter in Hub mode only when `REVSTHMV` is set to `V53` and `REVLOGNR` is set to a value higher than `0`..

The `REVTIMER` parameter may be activated during the Adabas nucleus session.

The possible values for this parameter are:

Value	Meaning
N	The maximum delay in seconds after which to send the CLOG records to the Review hub.

Example

The following example sends the CLOG records in packages of maximum 12 records via Network to the Review hub. In case of low activity, send the records with a maximum delay of 5 seconds (even if the buffer is not filled completely).

```
ADARUN PROG=ADANUC,REVSTHMV=V53,REVLOGNR=12,REVTIMER=5
```

RIAFTERUPDATE Parameter: Controlling RI Command Behavior

Parameter	Use	Values	Default
RIAFTERUPDATE	Indicate whether Adabas should suppress the response code 113 (ADARSP113) subcode 5 and response code 2 (ADARSP002) subcode 5 that are returned when an application issues an RI command for a record that has been updated in the current transaction.	YES | NO	NO

Normally, the RI command releases ISNs for records being held by the issuing user. The selected ISN for a single database file, or all ISNs held by the issuing user in all files can be released.

However, the records are not released unconditionally. If your application issues an RI command for a record that has been updated in the current transaction, Adabas normally returns response code 113 (ADARSP113), or if ISN=0 was specified, response code 2 (ADARSP002). With the RI-AFTERUPDATE parameter set to "YES", these response codes are suppressed. Instead, a response code of zero (0) is returned, and only records that have not been updated in the current transaction are released from hold. Updated records continue to be protected from concurrent updates by other users that might otherwise introduce inconsistencies.

This logic is made configurable for customers with old COBOL applications that are exposed to response code 113 (subcode 5) responses.

Example

The following example indicates that any RI commands for records that have been updated in the current transaction return should not be released from hold and a response code of zero should be returned.

```
ADARUN PROG=ADANUC,RIAFTERUPDATE=YES
```

RPLCONNECTCOUNT Parameter: Replication Connection Attempt Count

Parameter	Specify . . .	Possible Values	Default
RPLCONNECTCOUNT	the number of connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails with an unexpected response code.	0 - 2147483647	0

A value of zero indicates that no connection attempts should occur; a value of zero makes the most sense in situations where the Adabas database and the Event Replicator Server execute together on the same logical partition (LPAR). If the Adabas database and the Event Replicator Server execute on different LPARs, however, setting a real value using this command helps avoid errors that might arise if network problems occur because the network is not started or a network connection between the Adabas database and the Event Replicator Server is lost.

RPLCONNECTINTERVAL Parameter: Replication Connection Attempt Interval

Parameter	Specify . . .	Possible Values	Default
RPLCONNECTINTERVAL	the interval (in seconds) between connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails with an unexpected response code.	0 - 2147483647	0

A value of zero indicates that no connection attempts should occur; a value of zero makes the most sense in situations where the Adabas database and the Event Replicator Server execute together on the same logical partition (LPAR). If the Adabas database and the Event Replicator Server execute on different LPARs, however, setting a real value using this command helps avoid errors that might arise if network problems occur because the network is not started or a network connection between the Adabas database and the Event Replicator Server is lost.

RPLPARMS Parameter: Replication Parameter Location

Parameter	Specify . . .	Possible Values	Default
RPLPARMS	Where your replication definitions (initialization parameters) should be read from.	BOTH | FILE |NONE | PARMS	See below.

This parameter is only used when starting the Event Replicator database.

If the RPLPARMS parameter is not specified, the following logic is used to determine where to read the replication definitions:

- If the Replicator system file is loaded on the Event Replicator database, the replication definitions are read from that file.
- If the Replicator system file is not loaded on the Event Replicator database, the replication definitions are read from DDKARTE.

Valid values are described in the following table.

Value	Meaning
BOTH	The replication definitions are read first from the Replicator system file and secondly from DDKARTE. Separate entries are built for each and errors are generated if there are duplicate entries.
FILE	The replication definitions are read from the Replicator system file, which is maintained using the Adabas Event Replicator Subsystem in Adabas Online System (AOS).
NONE	No replication definitions are read. The Event Replicator Server will establish itself with a null set of replication definitions and will not communicate or handshake with related Adabas nuclei. This mode allows maintenance to occur to the Replicator system file, without the Event Replicator Server communicating with related Adabas nuclei.
PARMS	The replication definitions are read from DDKARTE.

RPLSORT Parameter: Event Replicator for Adabas Transaction Data Sorting Control

Parameter	Specify . . .	Possible Values	Default
`RPLSORT`	Whether Event Replicator for Adabas transaction data sorting should occur or not.	YES | NO	YES

This parameter is only used for the Adabas database from which transactions will be replicated. It should not be used for the Event Replicator Server database.

By default (`RPLSORT=YES`), Event Replicator for Adabas sorts transaction data by a key consisting of the file number, ISN, and relative update number. If you do not want this sort to occur, specify `RPLSORT=NO`.

It is important to note that the RPLSORT setting will affect how the Event Replicator consolidates transaction modifications and the order in which they are replicated. When `RPLSORT=YES`, all modifications for a given ISN in a file are consolidated before replication occurs. For complete information on regular modification consolidation, read *Detailed Adabas Nucleus Processing* in the *Event Replicator for Adabas Concepts Manual*. When `RPLSORT=NO` is specified, the modifications are still consolidated, but their referential integrity is maintained. In other words, the chronological order of the updates is maintained for each ISN in a file.

Note: If one or more source files with unique descriptors are replicated to an Adabas target file also with unique descriptors, you may need to run with RPLSORT set to "NO" to prevent an invalid response code198 (ADARSP198) from occurring when replicating the data.

Example

For example, assume the following modifications occur, in the following order:

```
Insert File 77 ISN 80 Seq 1
Update File 77 ISN 80 Seq 2
Insert File 77 ISN 1010 Seq 3
Update File 77 ISN 80 Seq 4
Update File 77 ISN 1010 Seq 5
```

If `RPLSORT=YES`, these modifications are consolidated and replicated as follows:

```
Insert File 77 ISN 80 Seq 1 + Seq 2 + Seq 4
Insert File 77 ISN 1010 Seq 3 + Seq 5
```

If `RPLSORT=NO`, these modifications are consolidated and replicated as follows:

```
Insert File 77 ISN 80 Seq 1 + Seq 2
Insert File 77 ISN 1010 Seq 3
Update File 77 ISN 80 Seq 4
Update File 77 ISN 1010 Seq 5
```

The end result is the same, regardless of whether RPLSORT is set to YES or NO, but multiple modifications to the same ISN in a file may be necessary when RPLSORT is set to NO.

RPWARNINCREMENT Parameter: Replication Pool Warning Message Interval

Parameter	Specify . . .	Possible Values	Default
`RPWARNINCREMENT`	The interval (in percent of `LRPL` usage) at which warning messages should be sent that indicate that the replication pool usage has exceeded the threshold set by the `RPWARNPERCENT` **parameter**.	1-99	10

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

RPWARNINTERVAL Parameter: Replication Pool Usage Interval

Parameter	Specify . . .	Possible Values	Default
RPWARNINTERVAL	The interval, in seconds, during which replication pool usage warning messages are suppressed on the console.	1-2147483647	60

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

When the number of replication pool usage warning messages sent to the console exceeds the limit specified by the RPWARNMESSAGELIMIT parameter, all such messages are suppressed for the interval of time specified by this parameter. When the interval specified by the RPWARNINTERVAL parameter has passed, warning messages resume.

This parameter, along with the RPWARNMESSAGELIMIT parameter, is provided to avoid flooding the console with replication pool usage warning messages.

RPWARNMESSAGELIMIT Parameter: Replication Pool Usage Warning Message Count

Parameter	Specify . . .	Possible Values	Default
RPWARNMESSAGELIMIT	The number of replication pool usage warning messages that can be issued to the console before message suppression begins.	1-2147483647	5

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

When the threshold specified by this parameter is reached, all replication pool usage warning messages are suppressed until the message interval specified by the RPWARNINTERVAL parameter has passed.

This parameter, along with the RPWARNINTERVAL parameter, is provided to avoid flooding the console with replication pool usage warning messages.

RPWARNPERCENT Parameter: Replication Pool Usage Threshold

Parameter	Specify . . .	Possible Values	Default
RPWARNPERCENT	The threshold for replication pool usage (expressed as a percentage of LRPL) at which warning messages should be sent. If "0" is specified or taken as the default there are no warning messages	0-99	0

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

The usage level should be expressed as a percentage. Warning messages will be printed when usage of the replication pool exceeds this percentage (threshold).

RVCLIENT Parameter: Adabas Review Client Reporting Activation

Parameter	Specify . . .	Possible Values	Default
RVclient	whether Adabas Review client reporting should be activated in batch environments.	ACTIVE | INACTIVE	INACTIVE

This ADARUN parameter allows you to activate Adabas Review client reporting when you want to run client reports in batch environments. Specify "ACTIVE" to activate it; specify "INACTIVE" (or specify no setting, since INACTIVE is the default) if you do not want client reporting activated.

Note: This ADARUN parameter is valid only when ADARUN PROGRAM=USER and LGBLSET RVCLNT=YES. For more information, read about the RVCLNT parameter in your Adabas or Adabas Review installation documentation.

Example

In the following example, client reporting is activated.

```
ADARUN PROGRAM=USER,RVCLIENT=ACTIVE
```

SECUID Parameter: Security System User ID Requirement Level

Parameter	Specify . . .	Possible Values	Default
SEcuid	the requirement level of security system user IDs for a database. Using this parameter, you can indicate how Adabas handles calls from a user without a security system user ID or with a security system user ID that has changed during the Adabas session.	ALLOW | REQUIRE | WARN	ALLOW

Valid values are ALLOW, REQUIRE, and WARN; ALLOW is the default.

Value	Description	Recommended for installations that...
ALLOW	Indicates that the user can continue working with the database when they have no security system user ID or when the security system user ID changes during their Adabas session. This is the default. **Note:** This is the same as working with Adabas in past releases.	will not make use of the security system user ID support provided with Adabas and that want to limit the possibility of any failures occurring with existing application programs due to the implementation of security system user IDs.
REQUIRE	Indicates that the user *must* have a nonblank security system user ID and that it must remain unchanged for the entire Adabas session. If the Adabas caller's security system user ID is blank or invalid, Adabas will reject the call and issue response code 200, subcode 21; if the Adabas caller's security system user ID changes during the Adabas session, Adabas will reject the call and issue response code 200, subcode 22.	make use of the security system user ID support provided with Adabas (for example, store it in a system field) and require that it be consistent for every user.
WARN	Indicates that a warning message will be issued but that the user can continue working with the database when they have no security system user ID or when their security system user ID changes during their Adabas session. These users are reported on, but Adabas allows their calls to be processed normally.	plan to require that all Adabas users have a consistent security system user ID, but that first want to verify that implementing security system user IDs will not break their existing applications.

Note: Various reports or displays about Adabas settings (such as the output of the DPARM operator command) include the current SECUID setting.

This parameter can be changed at runtime via operator command or corresponding Adabas Online System (AOS) and ADADBS utility functions. In a cluster, this is a global parameter; its setting is the same for all nuclei in the cluster.

Example

The following example indicates that Adabas will issue a warning message when a call is received from a user without a security system user ID or with a security system user ID that changed during the user's Adabas session.

```
ADARUN PROG=ADANUC,SECUID=WARN
```

SMF Parameter: Enable Adabas SMF Recording

Parameter	Specify . . .	Valid Values	Default
SMF	whether or not Adabas SMF recording should be enabled.	YES | NO	NO

This parameter allows you to enable and disable Adabas SMF recording.

- NO is the default and disables all SMF processing for the duration of the nucleus.
- YES enables all SMF processing for the duration of the nucleus. Adabas SMF records will be generated and written when the SMF parameter is set to YES.

This parameter *cannot* be altered by operator commands while the nucleus is running. If you specify SMF=YES, note that APF authorization is required to write to system SMF files.

Example

In the following example, the Adabas SMF records are generated and written.

```
ADARUN PROG=ADANUC,SMF=YES
```

SMF89 Parameter: Enable SMF Type 89 Records

Parameter	Specify . . .	Valid Values	Default
SMF89	whether or not the Adabas nucleus should register with z/OS for type 89 SMF records.	YES | NO	NO

This allows you to have the Adabas nucleus register with z/OS for type 89 SMF records.

- NO is the default and prevents registration.
- YES causes the nucleus to register with z/OS for type 89 SMF records..

This parameter *cannot* be altered by operator commands while the nucleus is running.

Example

In the following example, the nucleus will register with z/OS for type 89 SMF records.

```
ADARUN PROG=ADANUC,SMF89=YES
```

SMFDETAIL Parameter: Selecting Detail SMF Record Sections

Parameter	Specify . . .	Valid Values	Default
SMFDETAIL	the type of detail sections in SMF Interval and Termination records (subtypes 2 and 3) that should be included in Adabas SMF records. All specifications (regardless of how many) for this parameter should be enclosed in parentheses.	NONE ALL comma-delimited list of detail section names	NONE

Adabas SMF records can contain a variable set of detail sections in Interval and Termination records (subtypes 2 and 3). This parameter allows you to select the detail section types in Interval and Termination records that should be included in the Adabas SMF records. The value or values you specify for this parameter must be enclosed in parentheses. If you specify NONE or ALL, they should be specified alone. However, you can specify one or more of the other detail section names (CMD, CSHB, CSHF, CSHG, CSHP, FILE, IODD, LOCK, MSGB, MSGC, MSGH, PARM, STG, THRD, ZIIP or USER) in one SMFDETAIL parameter, separating each value with a comma.

Note: The CSHB, CSHF, CSHG, LOCK, MSGB, MSGC, and MSGH detail section names apply only to cluster environments (Adabas Cluster Services or Adabas Parallel Services must be installed). In addition, the CSHP detail section name only applies in cluster environments with Adabas Parallel Services installed.

The following table describes the meaning of the possible detail section names that can be used in the SMFDETAIL parameter:

Detail Section Name	Description
ALL	Generate all possible detail sections in the Adabas SMF records. If this value is specified, no others can be specified on the SMFDETAIL parameter.
CMD	Generate Adabas command activity detail sections in the Adabas SMF records.
CSHB[1]	Generate global cache activity by block detail sections in the Adabas SMF records.
CSHF[1]	Generate global cache activity by Adabas file number detail sections in the Adabas SMF records.
CSHG[1]	Generate global cache activity detail sections in the Adabas SMF records.
CSHP[2]	Generate Adabas Parallel Services cache activity detail sections in the Adabas SMF records.
FILE	Generate Adabas file activity detail sections in the Adabas SMF records.

Detail Section Name	Description
IODD	Generate I/O activity by DD statement detail sections in the Adabas SMF records.
LOCK[1]	Generate global lock activity detail sections in the Adabas SMF records.
MSGB[1]	Generate internucleus messaging control block activity detail sections in the Adabas SMF records.
MSGC[1]	Generate internucleus messaging count detail sections in the Adabas SMF records.
MSGH[1]	Generate internucleus messaging service time histogram detail sections in the Adabas SMF records.
NONE	Generate no detail sections in the Adabas SMF records. If this value is specified, no others can be specified on the SMFDETAIL parameter.
PARM	Generate ADARUN parameter value detail sections in the Adabas SMF records.
SESS	Generate Adabas session statistics sections in the Adabas SMF records.
STG	Generate Adabas storage pool detail sections in the Adabas SMF records.
THRD	Generate thread activity detail sections in the Adabas SMF records.
ZIIP	Generate zIIP statistics detail sections in the Adabas SMF records.
USER	Generate user-defined details sections in the Adabas SMF records. If USER is specified, a value for the UEXSMF parameter should also be specified to identify the user exit to be invoked to generate the user-defined detail section.

[1] *Available only in cluster environments (Adabas Cluster Services or Adabas Parallel Services must be installed).*

[2] *Available only in cluster environments with Adabas Parallel Services installed.*

This parameter can be altered by operator commands while the nucleus is running.

Example

In the following example, detail sections for Adabas command, file, and thread activity are generated and written in the Adabas SMF records.

```
ADARUN PROG=ADANUC,SMFDETAIL=(CMD,FILE,THRD)
```

In the following example, all detail sections are generated and written in the Adabas SMF records.

```
ADARUN PROG=ADANUC,SMFDETAIL=(ALL)
```

SMFINTERVAL Parameter: Enabling Adabas SMF Interval Records

Parameter	Specify . . .	Valid Values	Default
SMFINTERVAL	whether SMF interval records should be generated and the interval at which they should be generated.	NONE GLOBAL SUBSYS minutes	NONE

Adabas SMF records can be generated at specific intervals. This parameter enables and disables interval recording and specifies the interval or the source from which the interval can be derived.

The interval can be one of several values from z/OS specifications or an explicit interval in minutes. Interval records contain a product detail section and whatever detail sections are selected in the SMFDETAIL parameter, except for the ADARUN parameter detail section which are not included in SMF interval records.

The following table describes the meaning of the specifictions that can be used in the SMFINTERVAL parameter:

Valid Value	Description
GLOBAL	Generate all interval SMF records at the rate established by the INTVL and SYNCVAL SMF parameters in PARMLIB member SMFPRM*xx*.
NONE	Generate no interval SMF records. This is the default.
SUBSYS	Generate interval SMF records at the rate specified in PARMLIB member SMFPRM*xx* for the subsystem identified in the ADARUN SMFSUBSYS parameter.
minutes	Generate interval SMF records at the specified interval, in minutes. Valid minute values can range from 1 through 9999..

This parameter can be altered by operator commands while the nucleus is running.

Example

In the following example, interval SMF records are generated at the rate specified in PARMLIB member SMFPRM*xx* for the subsystem identified in the ADARUN SMFSUBSYS parameter.

```
ADARUN PROG=ADANUC,SMFINTERVAL=SUBSYS
```

In the following example, interval SMF records are generated every 6 hours (3600 minutes)

```
ADARUN PROG=ADANUC,SMFINTERVAL=3600
```

SMFRECNO Parameter: Selecting User-Defined Adabas SMF Record Number

Parameter	Specify . . .	Valid Values	Default
SMFRECNO	the Adabas SMF record number used for user-defined SMF records.	128 - 255	255

IBM designates the range of SMF numbers from 128 through 255 for user-defined records.

This parameter can be altered by operator commands while the nucleus is running.

Example

In the following example, user-defined SMF records will be assigned number 200.

```
ADARUN PROG=ADANUC,SMFRECNO=200
```

SMFSUBSYS Parameter: Specifying SMF Subsystem Name

Parameter	Specify . . .	Valid Values	Default
SMFSUBSYS	the name of the IBM or user-defined SMF subsystem from PARMLIB member SMFPRM*xx*.	CURRENT | *subsystem-name*	CURRENT

This parameter associates the Adabas SMF records with an IBM or user-defined subsystem. The z/OS systems staff can provide different sets of SMF system parameters for IBM-defined subsystems. Up to eight user-specified subsystems can be defined, each with their own set of system parameters. This parameter allows you to optionally identify one of these subsystem names to be used with Adabas SMF recording, if the one you want is different from the current one.

The possible values for this parameter are:

- CURRENT: This is the default value and associates the Adabas SMF record with the IBM-defined subsystem under which the Adabas nucleus is active. The subsystem may be TSO for a nucleus running in a TSO session, STC for a starting task or the name of the job entry subsystem under which a batch job is running, JES2 or JES3.
- An explicit subsystem name can be specified. It may be either an IBM-defined name or a user-defined name from PARMLIB member SMFPRM*xx*. The subsystem name is one to four characters long. The first character must be alphabetic or national (#, @, or $) and the remaining characters can be either alphanumeric or national characters. Contact your z/OS administrator for more information about the subsystem names available in your z/OS environment.

This parameter can be altered by operator commands while the nucleus is running.

SMGT Parameter: Error Handling and Message Buffering Facility

Parameter	Specify . . .	Possible Values	Default
SMGT	whether to enable the error handling and message buffering facility.	YES | NO	NO

The meanings of the possible values are described below:

Value	Meaning
YES	ADARUN loads the module ADAMXI during initialization. If you want to use the message buffering functions, you must also specify the ADARUN MSGBUF parameter with a value greater than zero.
NO	(default) Error handling and message buffering is not enabled for the session.

This parameter can only be used when starting the nucleus. It cannot be used in utilities or user programs.

For more information about the error handling and message buffering facility, see the *Adabas DBA Tasks Manual*.

For information about temporarily turning the facility off and back on, see the `SMGT` **operator command**.

Example

The following example activates the error handling and message buffering facility with a buffer size of 36 kilobytes when the Adabas nucleus is initialized.

```
ADARUN PROGRAM=ADANUC
  ADARUN SMGT=YES
  ADARUN MSGBUF=36
```

SORTCACHE Parameter: Sort Area Controller Caching

Parameter	Specify . . .	Possible Values	Default
SORTCACHE	whether to enable (YES) or disable (NO) controller caching for the Adabas Sort data set.	YES | NO	YES

SORTCACHE controls caching for the sort area component of Adabas.

Value	Meaning
YES	The default setting. Sort blocks are preread and maintained as long as possible in the caching facilities of the disk controller.
NO	Sort blocks are not preread or stored.

Guidelines

- If sort blocks are highly scattered or the update frequency is low, controller caching can cause longer I/O response times.
- ADARUN caching control parameters also exist for the Associator (ASSOCACHE), Data Storage (DATACACHE), the work area (WORKCACHE), and the temp area (TEMPCACHE).

Example

The following example runs the nucleus without controller caching for the sort data set.

```
ADARUN PROG=ADANUC,SORTCACHE=NO
```

SPT Parameter: Activate Adabas Triggers and Stored Procedures

Parameter	Specify . . .	Possible Values	Default
SPT	whether to enable use of triggers and stored procedures.	YES | NO	NO

When the Adabas nucleus starts and SPT=YES has been specified, it passes control to the Adabas trigger driver (ADATSP) to allow it to initialize.

This parameter can only be used when starting the nucleus and only in multiuser mode. It cannot be used for utilities or user programs and it cannot be used in single-user mode.

For more information, see the Adabas Triggers and Stored Procedures Manual.

Example

The following example sets Adabas to use the triggers and stored procedures functions.

```
ADARUN PROGRAM=ADANUC,MODE=MULTI,SPT=YES
```

SRLOG Parameter: Controlling Spanned Record Logging

Parameter	Specify . . .	Possible Values	Default
SRLOG	how spanned records should be logged to the protection logs.	ALL | UPD | PART	UPD

When SRLOG=ALL is specified, the entire spanned record is written to the PLOG. When SRLOG=UPD is specified, only the modified spanned record segments are written to the PLOG. When SRLOG=PART is specified, all spanned records up to and including the modified segments are written to the PLOG; the remainder of the spanned record segments are not written to the PLOG.

Spanned records are logical records that are segmented into two to five data storage records and chained together. Spanned records may be segmented at the field or byte level.

Note: Standalone secondary spanned records encountered in the PLOG will be rejected from further processing. A warning message is issued. Likewise, decompression of a partial field at the end of a spanned record will be skipped. All fields up to the partial field will be decompressed and the remaining fields on the spanned record are not available.

Examples

The following example requests complete spanned records should be logged to the PLOGs.

```
ADARUN PROGRAM=ADANUC,SRLOG=ALL
```

The following example requests that the complete spanned record, up to and including the modified segments, should be logged to the PLOGs.

```
ADARUN PROGRAM=ADANUC,SRLOG=UPD
```

SVC Parameter: SVC Number

Parameter	Specify . . .	Possible Values	Default
SVC	the Adabas SVC number or Adabas Review hub SVC number to be used for the session.	see text	249

The SVC number is specified as an integer. It must correspond to the number used for the Adabas SVC at your installation.

The Adabas SVC or Adabas Review hub SVC are used to perform various Adabas internal functions under z/OS.

Valid SVC values are between 200 and 255.

Example

The following example runs an Adabas session using SVC 202 for the Adabas SVC.

```
ADARUN PROG=ADANUC,SVC=202
```

TARGETID Parameter: Entire Net-Work Target ID

Parameter	Specify . . .	Minimum	Maximum	Default
TARGETID	the unique Entire Net-Work target ID for this node.	1	65535	1

The TARGETID parameter is an optional ADARUN parameter that specifies the unique Entire Net-Work target ID of a node. It is synonymous with the Adabas ADARUN DBID parameter.

All target IDs used by Entire Net-Work, Adabas (database IDs), Natural global buffer pools, etc., must be unique throughout all Entire Net-Work nodes. In particular, the Entire Net-Work target ID must not coincide with any database ID used in the network, with the exception of isolated databases that are defined for local availability only and are therefore unknown to Entire Net-Work. For more information about global target IDs, read *Target ID Handling in the Network*, in the *Entire Net-Work Administration Guide*.

Example

The following example specifies an Entire Net-Work target ID of 12 for a node.

```
ADARUN TARGETID=12
```

TCPIP Parameter: TCP/IP Access Control

Parameter	Specify . . .	Possible Values	Default
TCPIP	whether to activate a direct TCP/IP link to the Adabas nucleus.	YES | NO	NO

For UES-enabled databases, Adabas supports a direct TCP/IP link to the Adabas nucleus from web-based applications or from PC-based applications such as Natural for Windows.

See the Adabas Installation documentation for information about installing direct TCP/IP database access.

The TCPIP parameter is used to activate a direct TCP/IP link to the Adabas nucleus.

If TCPIP=YES, you can specify the universal resource locator (URL) required to activate the direct TCP/IP link in the ADARUN parameter TCPURL.

Example

The following example runs Adabas with a direct TCP/IP link to the nucleus.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=HPS://TCPLPAR1:12345
```

TCPURL Parameter: TCP/IP Universal Resource Locator

Parameter	Specify . . .	Possible Values	Default
TCPURL	the universal resource locator (URL) for the TCP/IP link.	(see text)	none

If TCPIP=YES, you can specify the information required to activate the direct TCP/IP link to the Adabas nucleus. The parameter value is a 20-byte address that conforms to the RFC specification for universal resource locators (URLs):

```
TCPURL=api-name:[//]stackid:port-number[:logging-setting[:allowipv6-setting]]
```

where:

api-name	A required three-character value identifying the application programming interface (API) to use. The APIs HPS and OES are currently supported.
stackid	A one to eight-character value identifying the stack to use: ■ For HPS, specify the name of the TCP/IP started task or job. ■ For OES on systems running a single TCP/IP stack, no value is needed. On systems running multiple TCP/IP stacks, specify the name of the TCP/IP started task or job. (No value is needed on systems with only one TCP/IP stack.)
port-number	A one to five-digit number in decimal notation.
logging-setting	A one-character setting indicating whether or not logging should be performed. Valid values are "Y" or "N"; a setting of "Y" turns logging on.
allowipv6-setting	A one-character setting indicating whether or not IPv6 addresses can be used. Valid values are "Y" or "N"; a setting of "Y" indicates that IPv6 addresses can be used.

Note: The forward slashes ("//") are optional.

Example 1

The following examples run Adabas with a direct TCP/IP link to the nucleus.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://:12216
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=HPS://TCPLPAR1:12213
```

Example 2

The following example runs Adabas with a direct TCP/IP link to the nucleus and with logging turned on.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://:12216:Y ↩
```

Example 3

The following example runs Adabas with a direct TCP/IP link to the nucleus and with logging turned on for a system with multiple TCP/IP stacks. In this example, the TCP/IP stack with the started task or job name of TCPIPMVS is requested.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://TCPIPMVS:12216 ↩
```

Example 4

The following example runs Adabas with a direct TCP/IP link to the nucleus and allows IPv6 addresses to be used.

Note: Note if you choose not to specify the logging setting (*logging-setting*), but to specify the IPv6 setting (*allowipv6-setting*), you must still specify the colons for both the logging and IPv6 settings. This is why there are two colons between "12216" and "Y" in the following example.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://:12216::Y
```

TEMPCACHE Parameter: Temp Area Controller Caching

Parameter	Specify . . .	Possible Values	Default
TEMPCACHE	whether to enable (YES) or disable (NO) controller caching for the Adabas Temp data set.	YES | NO	YES

Value	Meaning
YES	The default setting. Temp blocks are preread and maintained as long as possible in the caching facilities of the disk controller.
NO	Temp blocks are not preread or stored.

Guidelines

- If temp blocks are highly scattered or the update frequency is low, controller caching can cause longer I/O response times.
- ADARUN caching control parameters also exist for the Associator (ASSOCACHE), Data Storage (DATACACHE), the work area (WORKCACHE), and the sort area (SORTCACHE).

Example

The following example runs the nucleus without controller caching for the temp data set.

```
ADARUN PROG=ADANUC,TEMPCACHE=NO
```

TFLUSH Parameter: Synchronous Buffer Flush Duration

Parameter	Specify . . .	Minimum	Maximum	Default
TFLUSH	the time allowed for a synchronous buffer flush.	1	30	1

The TFLUSH value specifies the time allowed for a synchronous buffer flush, in seconds.

If the LFIOP parameter specifies a non-zero value, the TFLUSH value has no effect because asynchronous buffer flush control is active.

If LFIOP=0 (the default) and if TFLUSH is not specified or has the value 1, one second is allowed for each synchronous buffer flush.

Example

For example, during the following Adabas session, 6 seconds are allowed for each synchronous buffer flush.

```
ADARUN PROG=ADANUC,TFLUSH=6,LFIOP=0, ...
```

TLSCMD Parameter: Time Limit for S1, S2, and S4 Complex Searches

Parameter	Specify . . .	Minimum	Maximum	Default
TLSCMD	the time limit (in seconds) for executing a single S1, S2, or S4 command with complex search criteria.	1	the setting of the **ADARUN** MXTSX parameter	300

The TLSCMD parameter specifies the maximum time (in units of 1.048576 seconds) to be used to process a single Adabas S1, S2, or S4 command with complex search criteria.

If the estimated time to process a command exceeds the TLSCMD time limit, the command is not processed and the user who issued the command receives a nonzero response code.

If the estimated time to process a command does not exceed the time limit, the command is processed to completion, whether the actual time required exceeds the time limit specified by the TLSCMD or not.

If the TLSCMD setting exceeds the maximum time permitted by the **ADARUN** `MXTSX` parameter, the maximum `MXTSX` time is used.

Overriding the Parameter Setting

You can override this parameter setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.

Example

For example, during the following Adabas session, the time limit for an S1, S2, or S4 command with a complex search criterion is about 50 minutes.

```
ADARUN PROG=ADANUC,TLSCMD=3000
```

TMDRQ Parameter: Size of ATM Request Queue

Parameter Type	Use	Possible Values	Default
Runtime	Sets the number of entries allowed in the transaction manager's internal request queue.	10 - 32767	10

The ATM transaction manager uses an internal request queue for scheduling some of the tasks it needs to perform. The `TMDRQ` parameter determines the total number of entries in the queue that is used to service certain incoming, asynchronous requests. This queue is currently used only when the ATM RRMS interface is active.

RRMS Considerations

When the RRMS interface is in use, the ATM dynamic request queue (DRQ) is used for communication between ATM RRMS exits and the ATM manager. The value specified for the `TMDRQ` parameter should therefore be increased if RRMS is to be used. Consider the number of global transactions that may reach a syncpoint at the same time, and increase `TMDRQ` by this number. You can specify a higher value, for safety, without incurring any performance degradation.

You can use Online Services to monitor usage of the DRQ. If you specify too small a value and the DRQ becomes full, backouts resulting in response code 9 (ADARSP009) are likely to occur.

TMETDATA Parameter: ET Data Control

Parameter Type	Use	Possible Values	Default
Runtime	Location of ET data	ATM | TARGETS	TARGETS

The TMETDATA parameter determines the database or databases that will store ET data. The same TMETDATA parameter value must be specified for all Adabas Transaction Manager (ATM) instances on the network.

Possible values:

Value	Explanation
ATM	ET data is always stored in and read from the ATM transaction manager's recovery file, without regard to the database ID indicated by the Adabas command. ■ treats ET data as belonging to a global transaction rather than to a database; ■ there is just one copy of a client's ET data in a system; ■ eliminates confusion resulting from the existence of different ET data in different databases under the same ETID; ■ overrides the Natural ETDB parameter; ■ should not be used if the EmergencySerialETs runtime control will be set to YES or FORCE for any client environment in which ET data is stored or read.
TARGETS	ET data is stored during the commit process in all changed databases that are running with DTP=RM. ET data that is stored by an ET or CL command is always stored in the database that is the target of the command, whether or not that database runs with DTP=RM. It the database runs with DTP=NO, the ET data is written to the database only after any open global transaction has been committed. If the client storing ET data is at global transaction status, ATM is not required to carry out any commit processing; the data is simply stored in the database indicated by the Adabas command. ET data is always read from the database specified in the Adabas command. ■ treats ET data as belonging to a database or a local database transaction rather than to a global transaction; ■ a client's ET data might exist in several databases in a system; ■ a client can have several different ET data values in different databases at the same time; ■ means that 3GL application programs using ET data must have knowledge of the database that holds ET data for a given transaction; ■ honors the Natural ETDB parameter.

Notes:

1. The recommended setting for this parameter is the default, TARGETS. This setting eliminates the dangers associated with running in serial mode when the transaction manager is unavailable.
2. See also the section entitled *ET Data Storage* in your Adabas Transaction Manager operations documentation.

TMGTT Parameter: Global Transaction Time Limit

Parameter Type	Use	Possible Values	Default
Runtime	Sets time limit during which a global transaction can be open without being prepared.	1 - 16777215	720

The `TMGTT` parameter approximately specifies the maximum elapsed time, in units of 1.048576 seconds, that a global transaction can remain open without being prepared. If the limit is exceeded, ATM automatically backs out the global transaction from all affected databases. The next time the client issues a transactional Adabas command, a response code 9 (ADARSP009) is returned.

It is recommended that `TMGTT` be smaller than the lowest `TT` parameter of any database running with `DTP=RM`.

The `TMGTT` parameter does not supersede the `TT` parameter settings of other databases. If a database times out a client who has an unprepared global transaction open, the global transaction is subsequently backed out. A response code 9 (ADARSP009) is returned if the client then issues a transactional command.

Note: Transaction timeout can happen when a client session stops issuing commands for a period of time. In these circumstances the transaction manager has no opportunity to give a response code to the client, to indicate that the timeout has taken place. In such cases, the pending response code will be preserved and returned to the client at the first possible opportunity. For further details, please refer to *Pending Response Codes* in the Adabas Transaction Manager introduction.

TMLOG Parameter: ATM Manager Logging Control

Parameter Type	Use	Possible Values	Default
Runtime	Controls the logging option for the current execution of the ATM manager.	YES | NO | NEVER	NO

The ATM manager can log internal events for diagnostic purposes using a dual logging system. It can

- maintain a log buffer area in memory that contains details of the latest logged events;
- maintain transaction-related event details in storage areas related to specific global transactions;
- write every log record to dual sequential ATMLOG data sets.

Possible values:

Value	Description
YES	The ATM manager writes every record to an ATMLOG data set. The ATM manager keeps a record of the latest internal events in its global log buffer area. It also maintains the contents of log buffer areas related to open global transactions. Logging to the ATMLOG data sets can be deactivated at any time using the operator command `ATM NOLOG`.
NO	The ATM manager will not write log records to the ATMLOG data sets. Logging to the ATMLOG data sets can be activated at any time using the operator command `ATM LOG`. The ATM manager keeps a record of the latest internal events in its global log buffer area; it also maintains the contents of log buffer areas related to open global transactions.
NEVER	ATM logging is disabled for the duration of this execution of the ATM manager.

If the ATM manager writes log records to the ATMLOG (`TMLOG=YES` or `ATM LOG` issued) and the data set currently in use becomes full, the manager closes that data set and switches to the other data set. A system message might appear in the console log, indicating that the first log data set has become full. You may also switch between the two ATMLOG data sets by using the operator command `ATM FEOFLOG`. Whenever the ATMLOG data sets are switched, a message is displayed on the console indicating the data set that is currently in use.

TMMSGSEV Parameter: Severity Threshold for Warning Messages

Parameter Type	Use	Possible Values	Default
Runtime	Sets severity threshold for the suppression of warning messages.	0 | 4 | 8	0

Every message that the ATM manager sends to the console has a severity level. Message severity levels are explained in the Adabas Transaction Manager *Messages and Codes*.

Using the `TMMSGSEV` parameter, you can prevent messages with low severity levels from being written to the console by the ATM manager. This parameter has no effect on messages that are issued by the ATM client proxy, or during the early part of the transaction manager's initialization.

Possible values:

Value	Description
0	No messages are suppressed.
4	Messages of severity less than 4 are not sent to the console.
8	Messages of severity less than 8 are not sent to the console.

TMRESTART Parameter: ATM Restart Control

Parameter Type	Use	Possible Values	Default
Runtime	Controls restart handling of problematic transactions.	NORMAL | FORCE | FORCEALL	NORMAL

When an ATM manager restarts, it rebuilds its global transaction list as nearly as possible to its latest state when it was last executing. It then attempts to complete or back out any incomplete global transactions. The `TMRESTART` parameter determines the manager's restart action with regard to global transactions and transaction branches that remain incomplete at the end of this process.

Possible values:

Value	Description
NORMAL	Any incomplete transactions remain in an incomplete state until such time as they can be completed.
FORCE	ATM transfers to the suspect transaction journal (STJ) the details of every incomplete global transaction that has its root local to this ATM manager. The details of these transactions are deleted from ATM's recovery information, the originating clients are closed, and any related internal resources are freed. ATM can no longer guarantee integrity for such global transactions.

Value	Description
FORCEALL	Details of all unresolved global transactions and transaction branches are transferred to the STJ. The details are deleted from ATM's recovery information, the originating clients are closed, and any related internal resources are freed. ATM can no longer guarantee integrity for the affected global transactions.

TMSYNCMGR Parameter: Support for External Transaction Coordinators

Parameter Type	Use	Possible Values	Default
Runtime	Indicates whether or not the ATM manager is to interact with an external transaction coordinator.	NONE | RRMS	NONE

The `TMSYNCMGR` parameter determines whether or not the ATM manager registers and interacts with an external transaction coordinator.

Possible values:

Value	Description
NONE	The ATM manager will not interact proactively with an external transaction coordinator. This setting does not preclude the use of the CICS Syncpoint Manager, which interfaces directly with the ATM's client proxy component and not with the ATM manager.
RRMS	The ATM manager will register with the IBM Recoverable Resource Management Services so that it can participate in transactions that involve other RRMS-enabled resource managers. This parameter value is valid only for z/OS systems in which RRS (a component of RRMS) is active. The current version of ATM offers participation in RRMS-coordinated two-phase commit for single-user, single-TCB batch applications and for applications running under Com-plete or IMS TM.

TMTCIDPREF Parameter: Dynamic Client ID Prefix

Parameter Type	Use	Possible Values	Default
Runtime	Defines the first one or two characters of dynamically allocated Client IDs.	up to 2 alphanumeric characters	TM

To aid interpretation of Adabas Online Services displays and diagnostic logs, the ATM manager dynamically allocates an 8-byte client identifier (TCID) for each client session that engages in transactional activity. These client identifiers are simply labels, and have no operational significance.

The format of a dynamically allocated TCID is

```
ppnnxxxx
```

—where

pp is the value of the `TMTCIDPREF` parameter, left-justified and with trailing blanks if required.

nn is the 2-byte binary Node ID of the Adabas System Coordinator daemon within which the ATM transaction manager is running as a service.

xxxx is a 4-byte binary suffix. Suffixes are allocated in sequence and are reused. The first TCID to be allocated has a suffix of zeros.

TNAA Parameter: Nonactivity Time Limit for Access-Only Users

Parameter	Specify . . .	Minimum	Maximum	Default
`TNAA`	the nonactivity time limit for access-only users.	1	2147483647	900

The TNAA parameter specifies the maximum elapsed time (in units of 1.048576 seconds) that an access-only user may be considered active without issuing an Adabas command.

If an access-only user exceeds the nonactivity time limit, action is taken as described in ***Adabas Timeout Feature***, elsewhere in this guide.

If the NUCID parameter specifies a value greater than zero (that is, a multiprocessing nucleus), you may have to specify a larger TNAA value than for a normal nucleus. The chances of data contention are greater, and therefore more time should be allowed.

Overriding the Parameter Setting

You can override this parameter setting for an individual user or for all users:

- Override the setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.
- Override the setting for *all* users with an Adabas `TNAA` operator command or with the OPERCOM function of the ADADBS utility (described in the *Adabas Utilities Manual*).

Example

For example, during the following Adabas session, the nonactivity time limit for access-only users is about 30 minutes.

```
ADARUN PROG=ADANUC,TNAA=1800
```

TNAE Parameter: Nonactivity Time Limit for ET Logic Users

Parameter	Specify . . .	Minimum	Maximum	Default
TNAE	the nonactivity time limit for ET logic users.	1	2147483647	900

The TNAE parameter specifies the maximum elapsed time (in units of 1.048576 seconds) an ET Logic user may be considered active without issuing an Adabas command.

If an ET logic user exceeds the nonactivity time limit, action is taken as described in ***Adabas Timeout Feature***, elsewhere in this guide.

If the NUCID parameter specifies a value greater than zero (that is, a multiprocessing nucleus), you may have to specify a larger TNAE value than for a normal nucleus. The chances of data contention are greater, and therefore more time should be allowed.

Overriding the Parameter Setting

You can override this parameter setting for an individual user or for all users:

- Override the setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.
- Override the setting for all users with an Adabas `TNAE` operator command or with the OPERCOM function of the ADADBS utility (described in the *Adabas Utilities Manual*).

Example

For example, during the following Adabas session, the nonactivity time limit for ET logic users is about 30 minutes.

```
ADARUN PROG=ADANUC,TNAE=1800
```

TNAX Parameter: Nonactivity Time Limit for Exclusive Control Users

Parameter	Specify . . .	Minimum	Maximum	Default
TNAX	the nonactivity time limit for exclusive control users.	1	2147483647	900

The TNAX parameter specifies the maximum elapsed time (in units of 1.048576 seconds) an exclusive control user (EXU) may be considered active without issuing an Adabas command.

If an exclusive control user exceeds the nonactivity time limit, action is taken as described in ***Adabas Timeout Feature***, elsewhere in this guide.

If the NUCID parameter specifies a value greater than zero (that is, a multiprocessing nucleus), you may have to specify a larger TNAX value than for a normal nucleus. The chances of data contention are greater, and therefore more time should be allowed.

Overriding the Parameter Setting

You can override this parameter setting for an individual user or for all users:

- Override the setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.
- Override the setting for all users with an Adabas `TNAX` **operator command** or with the OPERCOM function of the ADADBS utility (described in the *Adabas Utilities Manual*).

Example

For example, during the following Adabas session, the nonactivity time limit for exclusive control users is about 30 minutes.

```
ADARUN PROG=ADANUC,TNAX=1800
```

TT Parameter: Transaction Time Limit

Parameter	Specify . . .	Minimum	Maximum	Default
TT	the transaction time limit for ET logic users.	1	2147483647	900

The TT parameter specifies the maximum elapsed time (in units of 1.048576 seconds) permitted for a logical transaction issued by an ET logic user.

The time measurement for a logical transaction starts when the first command that results in a record being placed in hold status is issued. It ends when an ET, BT, or CL command is issued.

If the transaction time limit is exceeded, action is taken as described in ***Adabas Timeout Feature***, elsewhere in this guide.

The value specified for TT directly influences the required size of the Adabas parameter LP (length of the data protection area).

we recommend that you set the TT parameter lower than the `TNAE` **parameter**.

Overriding the Parameter Setting

You can override this parameter setting for an individual user or for all users:

- Override the setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas Command Reference Guide*.
- Override the setting for all users with an Adabas `TT` **operator command** or with the OPERCOM function of the ADADBS utility (described in the *Adabas Utilities Manual*).

Example

For example, during the following Adabas session, the transaction time limit for ET logic users is about 5 minutes (300 seconds).

```
ADARUN PROG=ADANUC,TT=300
```

UEXn Parameter: User Exit

Parameter	Specify . . .	Possible Values	Default
`UEXn`	the user exit and the user routine to be given control.	see text	none

This parameter is used in conjunction with the user exit facility. It specifies one or more optional user exits and their names in the following format:

```
UEXn=exit-name
```

where

n	is a digit in the range 1-12 inclusive
exit-name	is the name of a user routine that gets control at the user exit; the name can be up to 8 characters long.

Note: User exit 2 and user exit 12 are mutually exclusive; if you specify one, you may not specify the other. Either UEX2 or UEX12 is required if the ADARUN **LOGWARN** parameter is specified.

The specified user exit routine(s) must be loadable at execution time.

The Adabas user exits discussion in the *Adabas User, Hyperdescriptor, Collation Descriptor, and SMF Exits Manual* describes the purpose and function of each user exit as well as the calling sequence.

Examples

In the following example, control is given to the user routine SECURE at user exit 11 during the Adabas session.

```
ADARUN PROG=ADANUC,UEX11=SECURE
```

During the following Adabas session, control is given to the user routine SUBR2 at user exit 2 and to the user routine SUBR4 at user exit 4.

```
ADARUN PROG=ADANUC,UEX2=SUBR2,UEX4=SUBR4
```

The following example runs the ADACMP utility and gives control to the user routine SUBR6 at user exit 6.

```
ADARUN PROG=ADACMP,UEX6=SUBR6
```

UEXSMF Parameter: Specifying SMF User Exit Module Name

Parameter	Specify . . .	Valid Values	Default
UEXSMF	the name of the SMF user exit module.	*module-name*	---

If user details sections are to be generated in the Adabas SMF record (if the ADARUN parameter SMFDETAIL is set to USER or if USER is added as a detail section via the SMFDETAIL operator command), this parameter can be used to identify the name of the SMF user exit that should be used to generate the user detail sections. This exit will be invoked for each Adabas SMF record.

This parameter *cannot* be altered by operator commands while the nucleus is running.

Example

In the following example, the SMF user exit name is SMFEXIT

```
ADARUN PROG=ADANUC,UEXSMF=SMFEXIT
```

UPDATECONTROL Parameter: Control Scheduling of Update Commands During Buffer Flush

Parameter	Specify . . .	Possible Values	Default
UPDATECONTROL	whether or not to shortly delay the scheduling of update commands at the end of buffer flushes.	DELAY | NODELAY	DELAY

The UPDATECONTROL parameter controls how the nucleus schedules update commands at the end of a buffer flush. Its setting influences the techniques that Adabas uses to recover the database in a session autorestart, in the case that this nucleus session terminates abnormally.

When UPDATECONTROL is set to DELAY, at the end of every buffer flush the nucleus delays the start of new update commands until all active update commands have finished. In a cluster, this happens in all nuclei in the cluster at the same time. Normal update command scheduling resumes once

all previously active update commands have ended. In the case that the nucleus terminates abnormally, the following session autorestart recovers the database by making use of the presence of a point in time before the failure where no update processing was in progress.

When UPDATECONTROL is set to NODELAY, the nucleus does not delay the start of new update commands at the end of buffer flushes. The buffer flushes do not impact normal update command scheduling. In the case that the nucleus terminates abnormally, the following session autorestart recovers the database even if update commands were in progress continually before the failure. Omitting the suspension of update commands after buffer flushes avoids the related short delays in update processing and, in a cluster, eliminates a category of periodic processes where all nuclei in the cluster must collaborate.

The default value of UPDATECONTROL is DELAY. The UPDATECONTROL setting cannot be changed at runtime. In a cluster, UPDATECONTROL must be set to the same value in all nuclei in the cluster. If the UPDATECONTROL setting is to be changed, all nuclei in the cluster must be shut down before the first one can be restarted with the new setting.

If UPDATECONTROL is set to NODELAY, the INDEXUPDATE parameter must be set to ADVANCED.

Note: If an Adabas nucleus running with UPDATECONTROL=NODELAY terminates abnormally, the UPDATECONTROL setting in the restarted nucleus does not matter for the autorestart.

UTIONLY Parameter: Utilities-Only Session Control

Parameter	Specify . . .	Possible Values	Default
UTIONLY	whether to restrict the session to Adabas utilities.	YES | NO	NO

Use this parameter to restrict an Adabas session to the execution of Adabas utilities.

Value	Meaning
YES	Any attempt to execute a program other than an Adabas utility is rejected.
NO	(default) The session is available for both utilities and user programs.

If you specify UTIONLY=YES to reserve the session for a utility program, you can use the operator command UTIONLY=NO to make the nucleus available for user programs again when the utility is completed.

Example

In the following example, only utilities can be run during this Adabas session.

```
ADARUN PROG=ADANUC,UTIONLY=YES
```

V64BIT Parameter: 64-Bit Virtual Storage Option

Parameter	Specify . . .	Possible Values	Default
V64bit	whether or not the Adabas nucleus should use virtual storage above the two gigabyte bar (64-bit virtual storage).	YES | NO	NO

This parameter indicates whether or not the Adabas nucleus should use 64-bit virtual storage. The default is NO.

Currently, the following storage areas can be allocated above the two gigabyte bar in an Adabas nucleus:

- The buffer pool (see ADARUN parameter LBP)
- The flush I/O pool (see ADARUN parameter LFIOP)
- The PLOG protection I/O buffers (see ADARUN parameter NPLOGBUFFERS)
- The Work part 1 protection I/O buffers (see ADARUN parameter NWORK1BUFFERS)
- The cast-out directory buffer, in cluster environments (automatically derived from the attributes of the global cache space for Adabas Cluster or Adabas Parallel Services).

The use of 64-bit virtual storage will help alleviate virtual storage constraints below the two gigabyte bar in installations that use large amounts of virtual storage in Adabas nucleus sessions (for example, nucleus sessions with ADARUN parameter settings of LBP=1200M and LFIOP=200M).

Value	Meaning
YES	The Adabas nucleus will use 64-bit virtual storage (storage above the two gigabyte bar).
NO	The default setting. The Adabas nucleus will not use 64-bit virtual storage.

Users should note that virtual storage above the two gigabyte bar (64-bit virtual storage) is shown in dumps to SVCDUMP or SYSMDUMP but not in dumps to MPMDUMP, SYSABEND, or SYSUDUMP. Therefore, we recommend that you use the SVCDUMP in the JCL of an Adabas nucleus running with ADARUN V64BIT=YES to ensure that 64-bit virtual storage is shown in any SVC dumps that are produced. If, for some reason, you do not want to produce SVC dumps, we recommend that you use SYSMDUMP to ensure 64-bit virtual storage is shown in a dump of the Adabas nucleus address space.

Specific Product Recommendations

- The replication pool (LRPL) is allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters REPLICATION=YES and V64BIT=YES.
- The replication pool (LRPL) is allocated in 64-bit virtual storage when the Event Replicator Server runs with ADARUN parameter V64BIT=YES.

- The Auditing pool (LAP) is allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters `AUDITING=YES` and `V64BIT=YES`.
- The Auditing pool (LAP) is allocated in 64-bit virtual storage when the Audit Server runs with ADARUN parameter `V64BIT=YES`.

Example

The following example indicates that the Adabas nucleus will make use of virtual storage above the two-gigabyte bar and should use 1M large pages.

```
ADARUN PROG=ADANUC,LARGEPAGE=YES,V64BIT=YES
```

VISTA Parameter: Adabas Vista Control

Parameter	Specify . . .	Possible Values	Default
VISTA	whether to enable support for Adabas Vista.	YES | NO	NO

The VISTA parameter enables the nucleus to support Adabas Vista processing. If Adabas Vista is not installed on your system, specifying this parameter invokes the AVILOOK analysis tool delivered with Adabas. For complete information about AVILOOK, read *AVILOOK* in the *Adabas DBA Tasks Manual*.

Example

The following example runs Adabas with Adabas Vista.

```
ADARUN PROG=ADANUC,VISTA=YES
```

WORKCACHE Parameter: Work Area Controller Caching

Parameter	Specify . . .	Possible Values	Default
WORKCACHE	whether to enable (YES) or disable (NO) controller caching for the Adabas Work data set.	YES | NO	YES

Value	Meaning
YES	The default setting. Work blocks are preread and maintained as long as possible in the caching facilities of the disk controller.
NO	Work blocks are not preread or stored.

Guidelines

- If Work blocks are highly scattered or the update frequency is low, controller caching can cause longer I/O response times.
- ADARUN caching control parameters also exist for the Associator (ASSOCACHE), Data Storage (DATACACHE), the sort area (SORTCACHE), and the temp area (TEMPCACHE).

Example

The following example runs the nucleus without controller caching for the Work data set.

```
ADARUN PROG=ADANUC,WORKCACHE=NO
```

ZIIP Parameter: Activate Usage of Adabas for zIIP

The ZIIP parameter controls whether Adabas activates Adabas for zIIP (AZPAD).

Note: Adabas for zIIP requires its own license file (AZPAD), in addition to the standard Adabas license file. If the AZPAD license file is missing or found invalid, Adabas will start but will run as if ZIIP=NO had been specified.

Parameter	Specify . . .	Possible Values	Default
ZIIP	whether or not to activate Adabas for zIIP.	YES | NO	NO

Value	Meaning
YES	Usage of Adabas for zIIP is enabled. Adabas offloads CPU time from the general processors to System z Integrated Information Processors (zIIP). The purpose of this configuration is to reduce the CPU consumption on the general processors.
NO	The default setting. Usage of Adabas for zIIP is disabled. Adabas runs without the option to offload CPU time to zIIPs.

Dynamic Modification

If Adabas was started with ZIIP=YES, the setting of the ZIIP parameter can be changed at runtime - that is, set to NO and subsequently back to YES - using the ZIIP command from the operator console, the ADADBS OPERCOM ZIIP function, or the Modify Parameters menu of Adabas Online System.

If Adabas was started with ZIIP=NO, the setting of the ZIIP parameter cannot be changed to YES later in this session.

ZIIP may be set differently for different nuclei in a cluster (it is a local, modifiable parameter).

Specific Product Recommendations

- Event Replicator Server databases and Adabas nuclei running with ADARUN `REPLICATION=YES` support the use of Adabas for zIIP subject to the availability of the appropriate license file (AZPAD).
- The Adabas Audit Server and Adabas nuclei running with `ADARUN AUDITING=YES` support the use of Adabas for zIIP subject to the availability of the appropriate license file (AZPAD).

Example

The following example activates the use of Adabas for zIIP.

```
ADARUN PROG=ADANUC,ZIIP=YES
```

6 Setting Adabas Session Parameters

- General Guidelines 228
- Using Session Statistics to Determine Parameter Settings 232
- Parameters for Operating Systems 232

This chapter describes how to set Adabas session parameters on the ADARUN control statement.

General Guidelines

Use the following guidelines when setting the various Adabas session parameters:

1. Ensure that the following are specified correctly:
   - program to be executed (see PROGRAM parameter).
   - mode of operation (MODE parameter).
   - database ID (see DBID parameter).
2. If the session is to be restricted to read only (no updating permitted), use the READONLY parameter.
3. If the session is to be restricted to Adabas utilities only, use the UTIONLY parameter.
4. Review buffer and table sizes to be in effect for the session:

Parameter	Description
ASYTVS, LFIOP, TFLUSH	Buffer flush control
LBP	Length of the buffer pool
LCP	Length of security pool
LDEUQP	Length of unique (UQ) descriptor pool
LDTP	If you have Adabas Transaction Manager installed, this parameter defines the size (in blocks) of the WORK4 index which is used to administer the data on DDWORKR4.
LFP	Length of internal format buffer pool
LI	Length of table for ISN lists
LP	Length of data protection area (Work part 1)
LQ	Length of table of sequential commands
LRDP	Length of the cluster redo pool
LS	Length of sort area
LU	Length of intermediate user buffer
LWKP2	Length of Work part 2
LWP	Length of Adabas work pool
MSGBUF	Length of message buffer for use with SMGT=YES
NAB	Number of attached buffers
NC	Number of elements in command queue
NH	Number of elements in hold queue
NISNHQ	Maximum number of records in hold queue per user

Parameter	Description
NQCID	Maximum number of active command IDs per user
NSISN	Maximum number of ISNs per TBI element
NT	Number of threads
NU	Number of elements in user queue

5. Review the time limits to be in effect for the session:

Parameter	Description
CT	Command timeout limit
MXTNA	Maximum inactivity limit using OP command for individual user
MXTSX	Maximum execution limit for S*x* command for individual user
MXTT	Maximum transaction time limit using OP command for individual user
TLSCMD	S*x* command time limit
TNAA	Inactivity limit for access only users
TNAE	Inactivity limit for ET logic users
TNAX	Inactivity limit for EXU users
TT	Transaction time limit

6. If dual protection logging is to be used, the DUALPLD and DUALPLS parameters apply.
7. If multiple (2-8) protection log data sets are to be used, the NPLOG, PLOGDEV, and PLOGSIZE parameters apply.
8. If command logging is to be used, the following parameters are applicable:

   CLOGMAX
   CLOGBMAX
   CLOGLAYOUT
   LOGGING
   LOGABDX, LOGCB, LOGCLEX, LOGFB, LOGIB, LOGIO, LOGMB, LOGRB, LOGSB, LOGUB, LOGUX, LOGVB
9. If a sequential command log is to be used, the LOGGING parameter applies.
10. If dual command log data sets are to be used, the DUALCLD and DUALCLS parameters apply.
11. If multiple (2-8) command log data sets are to be used, the NCLOG, CLOGDEV, and CLOGSIZE parameters apply.
12. If the multifetch or prefetch feature is to be used, the following parameters are applicable:

    PREFETCH
    PREFICMD
    PREFIFIL

PREFNREC
PREFSBL
PREFTBL
PREFXCMD
PREFXFIL

13. Determine if any of the following user exits are to be activated:

User Exit	Description
CDXnn	Collation (sort) descriptor processing
DSFEX1	Delta Save Facility user exit for use with DSF=YES
HEXnn	User hyperdescriptor processing
UEX1	User processing before command execution (retired)
UEX2	Dual log copy
UEX3	User phonetic processing
UEX4	User processing after command processing and before command logging
UEX5	Adabas Review hub event handler
UEX6	User processing before data compression
UEX8	Message/utility user processing
UEX9	User processing during file unload
UEX11	User processing before command execution
UEX12	Multiple log data set copy

14. Enable the nucleus to run with specific subsystems and associated products or in certain environments:

Parameter	Description
CACHE	Adabas Caching Facility (ACF)
CLUSTER	Adabas cluster nucleus session control (ALS or ASM)
DSF	Adabas Delta Save Facility Facility (ADE)
DTP	Enable the nucleus for distributed transaction processing as a resource or transaction manager (ATM)
FASTPATH	Adabas Fastpath (AFP)
REVIEW	Adabas Review (REV)
SMGT	Enhanced error handling (PINs) and message buffering subsystem
STP	Triggers and Stored Procedures Facility
VISTA	Adabas Vista (AVI)

15. If CLUSTER is either LOCAL or SYSPLEX, determine the cluster environment settings:

Parameter	Description
CLOGMRG	Merge cluster command logs automatically
CLUCACHENAME	Custer cache structure/area name
CLUCACHESIZE	Parallel cluster global cache area size
CLUCACHETYPE	Parallel cluster global cache area construct type
CLUGROUPNAME	Cluster name
CLULOCKNAME	Custer lock structure/area name
CLULOCKSIZE	Parallel cluster global lock area size
DIRRATIO/ELEMENTRATIO	Together define the ratio of directory entries to data elements
LRDP	Length of the cluster redo pool
MXCANCEL	Time limit for a canceled peer nucleus to terminate
MXMSG	Time limit for an inter-nucleus command to be processed
NUCID	Cluster nucleus ID

16. Determine the status of the nucleus in a network:

Parameter	Description
LOCAL	Enable nucleus for network connections
TCPIP	TCP/IP direct link to the nucleus
TCPURL	Identify the location of the TCP/IP link

17. Determine if any of the following parameters are applicable for the session:

Parameter	Description
DEVICE	Device type for first block of Associator
FORCE	Ignore active database ID setting in the ID table
IGNDIB	Ignore active nucleus entry in data integrity block (DIB)
IGNDTP	Ignore data in Work part 4 on startup for use with DTP=RM
NONDES	Non-descriptor searches
OPENRQ	Open command required
PLOGRQ	Selective/normal protection log required
QBLKSIZE	Sequential block size

18. When performing an autorestart following an abend, use the optional AREXCLUDE parameter to exclude any problem files from processing.

Using Session Statistics to Determine Parameter Settings

The Adabas session statistics are printed automatically at the end of a session and may be displayed during the session. These statistics can help you determine the optimum settings for each parameter. Adabas Online System may also be used to obtain session statistics.

Parameters for Operating Systems

Some ADARUN parameters are operating-system dependent, and must be set according to the particular operating environment.

z/OS Systems

Parameter	Description
SVC	z/OS SVC assignment for Adabas

z/OS Systems

Parameter	Description
*xxxx*CACHE	Disable controller caching for Adabas components, as desired. Caching is active if not explicitly disabled.

7 ADARUN Parameters Pertinent to Adabas Add-on Products

- Adabas Caching Facility ADARUN Parameters 234
- Adabas Fastpath ADARUN Parameters 235
- Adabas Cluster Services ADARUN Parameters 235
- Adabas Delta Save Facility ADARUN Parameters 236
- Adabas Parallel Services ADARUN Parameters 236
- Adabas Review ADARUN Parameters 237
- Adabas Transaction Manager ADARUN Parameters 238
- Adabas Vista ADARUN Parameters 239
- Event Replicator for Adabas ADARUN Parameters 239
- Adabas Auditing ADARUN Parameters 240

This chapter describes the ADARUN parameters related to specific Adabas add-on products.

Adabas Caching Facility ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Adabas Caching Facility:

Parameter	Usage
CACHE	Adabas Caching Facility control
CACTIVATE	RABN activation
CASSODSP	Associator RABNs or RABN range for data space cache
CASSOEXT	Associator RABNs or RABN range for extended memory cache
CASSOHSP	Associator RABNs or RABN range for hiperspace cache
CASSOL64	Associator RABNs or RABN range for virtual 64 backed by large pages
CASSOV64	Associator RABNs or RABN range for virtual 64 storage cache
CASSOMAXS	Associator cache space unit size
CBUFNO	Number of read-ahead buffers for concurrent I/O processing.
CCTIMEOUT	Cache space area or RABN inactivity time
CDATADSP	Data Storage RABNs or RABN range for data space cache
CDATAEXT	Data Storage RABNs or RABN range for extended memory cache
CDATAHSP	Data Storage RABNs or RABN range for hiperspace cache
CDATAL64	Data Storage RABNs or RABN range for hiperspace backed by large pages
CDATAV64	Data Storage RABNs or RABN range for virtual 64 storage cache
CDATAMAXS	Data Storage cache space unit size
CDEMAND	Lowest Adabas buffer efficiency for caching
CDISPSTAT	RABN range statistics display control
CEXCLUDE	Excluded command types
CFILE	File range
CMAXCSPS	Storage area quantities
CRETRY	Retry time
CSTORAGE	Override cache type
CWORKSTORAGE	Work area cache type
CWORK2FAC	Work part 2 percentage cached
CWORK3FAC	Work part 3 percentage cached
CXFILE	Excluded files

Adabas Fastpath ADARUN Parameters

The FASTPATH parameter provides Adabas Fastpath control.

Adabas Cluster Services ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Adabas Cluster Services:

Parameter	Usage
CLOGMRG	Merge cluster command logs automatically
CLUCACHEEXTRA	The additional vector size, in blocks, required to allow Adabas Cluster Services and Adabas Parallel Services to track the blocks held by Adabas Caching Facility in the different cluster nuclei.
CLUCACHENAME	Adabas cluster cache structure/area name
CLUGROUPNAME	Adabas cluster group name
CLULOCKNAME	Adabas cluster lock structure/area name
CLUPUBLPROT	Adabas cluster publishing of protection data
CLUSTER	Adabas cluster nucleus session control
DIRRATIO	Together with ELEMENTRATIO defines the ratio of directory entries to data elements in Adabas cluster environments
ELEMENTRATIO	Together with DIRRATIO defines the ratio of directory entries to data elements in Adabas cluster environments
LDEUQP	The number of bytes to be used for the unique (UQ) descriptor pool during a session.
LRDP	Length of the cluster redo pool
MLWTO	Whether multi-line messages that are normally written to the operator console as a series of single-line write-to-operator (WTO) messages should instead appear as multi-line WTO messages.
MXCANCEL	Time limit for a canceled peer nucleus to terminate
MXCANCELWARN	Cluster nucleus timeout threshold for canceled peer nucleus warning
MXMSG	Time limit for an inter-nucleus command to be processed
MXMSGWARN	Cluster nucleus timeout threshold for internucleus command
MXSTATUS	Member-level XCF status monitoring heartbeat
MXWTOR	Cluster nucleus self-termination operator query interval
NUCID	Adabas cluster nucleus ID
PROGRAM	Program to run
SMFDETAIL	The type of detail sections in SMF Interval and Termination records (subtypes 2 and 3) that should be included in Adabas SMF records.

Parameter	Usage
V64BIT	Whether the Adabas nucleus should use virtual storage above the 2 gigabyte bar. In order for the nucleus to use 64-bit virtual storage, the operating system must also support 64-bit virtual storage.

Adabas Delta Save Facility ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Adabas Delta Save Facility:

Parameter	Usage
DSF	Delta Save Facility control
DSFEX1	Delta Save Facility user exit

Adabas Parallel Services ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Adabas Parallel Services:

Parameter	Usage
CLOGMRG	Merge cluster command logs automatically
CLUCACHEEXTRA	The additional vector size, in blocks, required to allow Adabas Cluster Services and Adabas Parallel Services to track the blocks held by Adabas Caching Facility in the different cluster nuclei.
CLUCACHESIZE	Adabas cluster global cache area size
CLUCACHETYPE	Adabas cluster global cache area storage type
CLUCACHEUNCHANGED	Adabas cluster global cache unchanged block option
CLUGROUPNAME	Adabas cluster group name
CLULOCKSIZE	Adabas cluster global lock area size
CLUPUBLPROT	Adabas cluster publishing of protection data
CLUSTER	Adabas cluster nucleus session control
DIRRATIO	Together with ELEMENTRATIO defines the ratio of directory entries to data elements in Adabas cluster environments
ELEMENTRATIO	Together with DIRRATIO defines the ratio of directory entries to data elements in Adabas cluster environments
LDEUQP	The number of bytes to be used for the unique (UQ) descriptor pool during a session.
LRDP	Length of the cluster redo pool

Parameter	Usage
MLWTO	Whether multi-line messages that are normally written to the operator console as a series of single-line write-to-operator (WTO) messages should instead appear as multi-line WTO messages.
MXCANCEL	Time limit for a canceled peer nucleus to terminate
MXCANCELWARN	Cluster nucleus timeout threshold for canceled peer nucleus warning
MXMSG	Time limit for an inter-nucleus command to be processed
MXMSGWARN	Cluster nucleus timeout threshold for internucleus command
MXWTOR	Cluster nucleus self-termination operator query interval
NUCID	Adabas cluster nucleus ID
PROGRAM	Program to run
SMFDETAIL	The type of detail sections in SMF Interval and Termination records (subtypes 2 and 3) that should be included in Adabas SMF records.
V64BIT	Whether the Adabas nucleus should use virtual storage above the 2 gigabyte bar. In order for the nucleus to use 64-bit virtual storage, the operating system must also support 64-bit virtual storage.

Adabas Review ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Adabas Review:

Parameter	Usage
CT	Command time limit (seconds)
FORCE	Overwrite database ID
LOCAL	Nucleus for local use only; unknown to network
NAB	Number of attached buffers
NC	Number of command queue elements
PROGRAM	Program to run
REVBUFMB	Specify the size of the Review Buffer Pool used as temporary storage when sending the CLOG records via Network (ADALNK) to the Review Hub. (Adabas Review)
REVFILTER	Allow Adabas Review record filtering during the session
REVIEW	Adabas Review control
REVLOGBMAX	The maximum allowable number of bytes of a logged buffer for Adabas Review
REVLOGMAX	The maximum size of all of the logged buffers allowed for a Adabas Review command
REVLOGNR	Hub mode: Specify whether the CLOG records are sent as buffered records or as single records via Network (ADALNK) to the Review Hub.

Parameter	Usage
	Local mode: Specify the CLOG record after which the REVIEW subtask is posted for processing.
REVSTHMV	Specify whether the ADARMT/ADRST interface or the ADARMX/ADARSX interface is used to send data via Network (ADALNK) to the Review Hub.
REVTIMER	Hub mode: Specify the largest delay for sending buffered CLOG records via Network (ADALNK) in case of low database activity. Local mode: Send the CLOG records to the Review hub after no later than *N* seconds.
RVCLIENT	Whether Adabas Review client reporting should be activated when you want to run client reports in batch environments.
SVC	SVC number

Adabas Transaction Manager ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Adabas Transaction Manager:

Parameter	Usage
DTP	Distributed transaction processing control
IGNDTP	Ignore distributed transaction processing area (Work part 4)
LDTP	If you have Adabas Transaction Manager installed, this parameter defines the size (in blocks) of the WORK4 index which is used to administer the data on DDWORKR4.
TMDRQ	Size of ATM request queue
TMETDATA	ATM ET data control
TMGTT	Adabas Transaction Manager global transaction time limit
TMLOG	ATM manager logging control
TMMSGSEV	Adabas Transaction Manager severity threshold for warning messages
TMRESTART	ATM manager restart control
TMSYNCMGR	ATM manager external transaction coordinator
TMTCIDPREF	Adabas Transaction Manager dynamic client ID (TCID) prefix

Adabas Vista ADARUN Parameters

The VISTA parameter provides Adabas Vista control.

Event Replicator for Adabas ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Event Replicator for Adabas:

Parameter	Usage
ASYTVS	Asynchronous buffer flush based on volume serial number
FMXIO	Limit parallel I/O operations by LFIOP flush processing
LBP	Length of buffer pool
LDEUQP	Length of unique (UQ) descriptor pool
LFIOP	Length of asynchronous flush pool
LFP	The size (in bytes) of the Adabas internal format buffer pool.
LP	Length of data protection area (Work part 1)
LRPL	Adabas or Event Replicator replication pool size
LU	Length of intermediate user buffer
LWP	The size (in bytes) of the Adabas work pool.
MLWTO	Whether multi-line messages that are normally written to the operator console as a series of single-line write-to-operator (WTO) messages should instead appear as multi-line WTO messages.
NAB	Number of attached buffers
NC	The maximum number of command log data sets to be opened and logged in the PPT for the session.
NH	Number of hold queue elements
NISNHQ	Number of ISNs in hold queue for user
NT	Number of threads
NU	The maximum number of user queue elements for the Adabas session.
REPLICATION	Activate replication for the Adabas nucleus (Event Replicator for Adabas)
RPLCONNECTCOUNT	The number of connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails
RPLCONNECTINTERVAL	The interval (in seconds) between connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails
RPLPARMS	Location where your replication definitions (initialization parameters) should be read from (Event Replicator for Adabas)
RPLSORT	Event Replicator for Adabas transaction data sorting

Parameter	Usage
RPWARNINCREMENT	Replication pool usage threshold problem warning message interval
RPWARNINTERVAL	Replication pool usage warning messages suppression interval
RPWARNMESSAGELIMIT	Replication pool usage warning message count before message suppression begins
RPWARNPERCENT	Replication pool usage threshold

Adabas Auditing ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Adabas Auditing:

Parameter	Usage
ALOGDEV	Multiple audit logs device type
ALOGSIZE	Multiple audit log size
ASYTVS	Asynchronous buffer flush based on volume serial number
AUDITING	Enable nucleus auditing
DUALALD	Dual audit log device type
DUALALS	Dual audit log device size
FMXIO	Limit parallel I/O operations by LFIOP flush processing
LAP	Adabas auditing pool size
LARGEPAGE	Large page support option
LBP	Length of buffer pool
LDEUQP	Length of unique (UQ) descriptor pool
LFIOP	Length of asynchronous flush pool
LP	Length of data protection area (Work part 1)
LU	Length of intermediate user buffer
NAB	Number of attached buffers
NALOG	Number of audit log data sets
NC	Number of command queue elements
NH	Number of hold queue elements
NISNHQ	Number of ISNs in hold queue for user
NT	Number of threads
V64BIT	64-bit virtual storage option
ZIIP	Activate usage of Adabas for zIIP

III Other Operations

The following topics are covered:

Operator Commands
Adabas Session Execution
Adabas Utility Execution
Linking Applications to Adabas
Adabas Restart and Recovery
Local Time Zone Change Support
Adabas Timeout Feature

8 Operator Commands

- Entering Operator Commands on z/OS Systems 244
- Operator Command Groupings 245
- Nucleus Commands 245
- Adabas Utility Operator Command DSTAT 317
- SMGT Operator Command 318
- SMGT Display Output Samples 323

Adabas operator commands are entered during an Adabas session or during utility operation to

- terminate an Adabas or user session;
- display nucleus or utility information;
- log commands into CLOG;
- change Adabas operating parameters or conditions.

In this chapter, the commands are listed alphabetically. One command, DSTAT, is listed twice: once as a command for displaying nucleus status, and separately as a command to display current Adabas utility operating status.

- For Adabas Caching Facility operator commands, see the Adabas Caching Facility documentation.
- For Adabas Parallel Services operator commands, see the Adabas Parallel Services documentation.
- For operator commands in a sysplex cluster environment, see the Adabas Cluster Services documentation.
- For Adabas Auditing operator commands, see the Adabas Auditing documentation.

Entering Operator Commands on z/OS Systems

To enter operator commands in z/OS environments, use the z/OS MODIFY (F) command as shown below:

```
F { jobname,command [/* comment */]
  { jobname,command[ , comment ]
```

where *jobname* is the name specified by the EXEC job control statement (usually ADARUN), and *command* is the command you wish to run.

You can also add *comment*s to your operator commands when you issue them. These comments can be added using either of the following methods:

- After the command, specify the comment within a slash-asterisk (/*) and asterisk-slash (*/) pair. For example: `F jobname,DUQ /*comment*/`
- After the command, specify a blank, then a comma, then another blank and then the comment. For example: `F jobname,DUQ , comment`

The comment is included when the operator command is written to DDPRINT in message ADAI29.

Operator Command Groupings

The Adabas operator commands are grouped as follows:

- *Nucleus Commands*
- *Adabas Utility Operator Command DSTAT*
- *SMGT Operator Command*

Nucleus Commands

The following operator commands can be entered from the console to monitor and control Adabas nucleus operation.

- ADAEND Command
- ALOCKF Command
- AOSLOG Command
- ASSOSPACEWARN Command
- ASYTVS Command
- AUDCONNECT Command
- AUTOINCASSOSIZE Command
- AUTOINCASSOTHRESHOLD Command
- AUTOINCASSOTOTAL Command
- AUTOINCDATASIZE Command
- AUTOINCDATATHRESHOLD Command
- AUTOINCDATATOTAL Command
- CANCEL Command
- CLOGMRG Command
- CLUFREEUSER Command
- CT Command
- DATASPACEWARN Command
- DAUQ Command
- DCMDSTAT Command
- DCQ Command
- DDIB Command
- DDSF Command
- DELCQ Command
- DELUF Command
- DELUI Command
- DFILES Command
- DFILESTAT Command
- DFILUSE Command

- DHQ Command
- DHQA Command
- DLOCKF Command
- DMEMTB Command
- DNC Command
- DNFV Command
- DNH Command
- DNU Command
- DONLSTAT Command
- DPARM Command
- DPPT Command
- DRES Command
- DRPLSTAT Command
- DSPACE Command
- DSTAT Command
- DTARG Command
- DTH Command
- DUQ Command
- DUQA Command
- DUQE Command
- DUUQE Command
- DVOLIO Command
- DXCACHE Command
- DXFILE Command
- DXLOCK Command
- DXMSG Command
- DXSTAT Command
- DZSTAT Command
- FEOFAL Command
- FEOFCL Command
- FEOFPL Command
- FMXIO Command
- HALT Command
- INDEXCROSSCHECK Command
- LICREFRESH Command
- LOCKF Command
- LOCKU Command
- LOCKX Command
- LOGGING Command
- LOGCB Command
- LOGFB Command
- LOGIB Command
- LOGIO Command
- LOGRB Command
- LOGSB Command

- LOGUX Command
- LOGVB Command
- LOGVOLIO Command
- LOGWARN Command
- MXCANCEL Command
- MXCANCELWARN Command
- MXMSG Command
- MXMSGWARN Command
- MXSTATUS Command
- MXWTOR Command
- NOLOGGING Command
- NOLOGCB Command
- NOLOGFB Command
- NOLOGIB Command
- NOLOGIO Command
- NOLOGRB Command
- NOLOGSB Command
- NOLOGUX Command
- NOLOGVB Command
- NOLOGVOLIO Command
- NWCONNECT Command
- ONLRESUME Command
- ONLSTOP Command
- ONLSUSPEND Command>
- RALOCKF Command
- RALOCKFA Command
- RDUMPST Command
- READONLY Command
- REFSTPRT Command
- REVIEW Command
- RPLCHECK Command
- RPLCONNECT Command
- RPLCONNECTCOUNT Command
- RPLCONNECTINTERVAL Command
- SECUID Command
- SMFDETAIL Command
- SMFDETAILADD Command
- SMFDETAILDEL Command
- SMFINTERVAL Command
- SMFRECNO Command
- SMFSUBSYS Command
- STOPF Command
- STOPI Command
- STOPSU Command
- STOPSUR Command

- STOPU Command
- STOPUR Command
- SYNCC Command
- TARGET Command
- TCPIP Command
- TNAA Command
- TNAE Command
- TNAX Command
- TT Command
- UNLOCKF Command
- UNLOCKU Command
- UNLOCKX Command
- UTIONLY Command
- ZIIP Command

Note: The DUMP command, which stopped nucleus operation and issued a dump, is no longer supported. To stop operation, use an operating system-dependent command such as a z/OS Cancel.

ADAEND Command

```
ADAEND [,GLOBAL]
```

Use the `ADAEND` command to terminate Adabas session normally. No new users are accepted after this command has been issued. ET logic updating is continued until the end of the current logical transaction for each user. After all activity has been completed as described above, the Adabas session is terminated. In nucleus cluster environments, the GLOBAL option can be used to terminate the Adabas session in all active cluster nuclei.

ALOCKF Command

```
ALOCKF=n
```

Use the `ALOCKF` command to lock a file in advance to ensure that an EXU, EXF, or UTI user will be able to obtain exclusive control of the specified file. The advance-lock prevents new transactions from using the file. Once all current users have stopped using the file, the exclusive-control user will get the lock. Until then, Adabas keeps the exclusive-control user waiting.

To remove the advance lock without running the utility, see the `RALOCKF` command.

This command is not available in single user mode or for a read-only nucleus. It is available in cluster and non-cluster environments.

The following key points should be noted about advance-locks on files:

1. An advance-lock can be set while a file is being used.
2. A command requesting exclusive control (UTI, EXF, or EXU) over an advance-locked file will wait in the command queue until all other users stop using the file before it starts processing the file.
3. Advance-locks are automatically removed when a user gets exclusive control over the file. However, if a file is locked (via the LOCKF, LOCKU, or LOCKX commands), the locks are not removed when a user gets exclusive control over the file. (Locks must be explicitly removed, whereas advance-locks are automatically removed.)
4. Adabas will reject an advance-lock on a file that is already locked (via the LOCKF, LOCKU, LOCKX or ALOCKF commands) but will accept a lock request on an advance-locked file.
5. To ensure you have uninterrupted exclusive control over a file in a situation where you have multiple steps to run that require uninterrupted exclusive control while all steps have been processed, use a combination of advance-locking the file (ALOCKF), stopping all users of the file (STOPF), and locking the file (LOCKU).
6. In the case of expanded files, an ALOCKF command is applied to the anchor file (representing the entire expanded file chain).
7. In a cluster environment, advance-locks are effective in all nuclei of the cluster.

Simple Example

In the following example, issuing the ALOCKF request to advance-lock file 32 ensures that file 32 will be available so the ADALOD UPDATE function can take exclusive control (via a UTI request) of the file for its processing:

```
ADADBS OPERCOM ALOCKF=32
ADALOD UPDATE FILE=32
```

Adabas processing proceeds in the following manner for these utility functions:

1. When the ADADBS OPERCOM ALOCKF request is submitted, file 32 is marked as advance-locked.
2. If there are any active users of file 32, the ADALOD UTI request cannot be granted immediately and will wait for the active users to end their transactions or sessions. Active users continue to issue commands against file 32. However, requests by new users for file 32 are rejected because of the advance-lock on the file.
3. When all active users of file 32 have ended their transactions or sessions, the ADALOD UTI request for exclusive control can be granted. Once exclusive control is established, ADALOD UPDATE processing can occur.

As part of the successful execution of the ADALOD UTI request, the advance-lock is removed from the file. However, because ADALOD processing now has exclusive control of file 32, other users still cannot access it.

To accelerate the process and limit the wait time for the ADALOD UTI request, you can simply stop all active users of the file by force using the STOPF operator command:

```
ADADBS OPERCOM ALOCKF=32
ADADBS OPERCOM STOPF=32
ADALOD UPDATE FILE=32
```

In this case, the STOPF command will cause the nucleus to back out and stop users of file 32 before the ADALOD UTI request is granted. In addition, the advance-lock request specified by the ALOCKF command will prevent new users from accessing the file until the ADALOD UTI request for exclusive control is processed.

More Complex Example

One limitation of ALOCKF processing is that it cannot ensure that a job gets uninterrupted exclusive control if the job consists of multiple job steps in a series. Suppose an installation wants to run the following utility sequence:

```
ADAULD UNLOAD FILE=45
ADADBS REFRESH FILE=45
ADALOD UPDATE FILE=45
```

An ALOCKF request to advance-lock file 45 in this case would only work for the ADAULD UNLOAD function, because the ADAULD EXU request for exclusive control of file 45 would remove the advance-lock. If there are active users who try to issue commands against file 45, there is a chance that one of them will execute a command between the UNLOAD and REFRESH steps, or between the REFRESH and UPDATE steps. Such a user may also prevent the REFRESH or UPDATE step from obtaining exclusive control of file 45.

To ensure you have uninterrupted exclusive control over the file in this situation, use a combination of advance-locking the file (ALOCKF), stopping the file (STOPF), and locking the file (LOCKU):

```
ADADBS OPERCOM ALOCKF=45
ADADBS OPERCOM STOPF=45
ADADBS OPERCOM LOCKU=45
ADAULD UNLOAD FILE=45
ADADBS REFRESH FILE=45
ADALOD UPDATE FILE=45
ADADBS OPERCOM UNLOCKU=45
```

In this example, Adabas processing proceeds in the following manner:

1. When the ADADBS OPERCOM ALOCKF request is submitted, file 45 is marked as advance-locked.

The ADADBS OPERCOM STOPF request causes the nucleus to back out and stop users of file 45. (This step is optional.)

The ADADBS OPERCOM LOCKU request locks the file more permanently than the ALOCKF request since the LOCKU lock will stay in effect until it is explicitly released.

2. If there are any active users updating file 45, the ADAULD EXU request cannot be granted immediately and will wait for the update users to end their transactions or sessions. Active users may continue to issue commands against file 45. However, requests by new users for file 45 are rejected because of the advance-lock on the file.
3. When all active users of file 45 have ended their transactions or sessions, the ADAULD EXU request for exclusive control can be granted. Once exclusive control is established, ADAULD UNLOAD processing can occur.

   As part of the successful execution of the ADAULD UTI request, the advance-lock is removed from the file. However, because ADAULD processing now has exclusive control of file 45, other users still cannot access it.

   When ADAULD processing completes, it releases exclusive control of file 45. However, the LOCKU lock ensures that other users still cannot access it.
4. The ADADBS utility will issue a UTI request for exclusive control of file 45, which will be granted. ADADBS REFRESH processing will then occur. When it completes, ADADBS will release exclusive control of file 45. However, the LOCKU lock ensures that other users still cannot access it.
5. The ADALOD utility will issue a UTI request for exclusive control of file 45, which will be granted. ADALOD UPDATE processing will then occur. When it completes, ADALOD will release exclusive control of file 45. However, the LOCKU lock ensures that other users still cannot access it.
6. The ADADBS OPERCOM UNLOCKU request explicitly unlocks file 45, making it available for other users.

AOSLOG Command

```
AOSLOG={YES | NO}
```

Use the `AOSLOG` command to activate and deactivate logging of certain Adabas calls that modify the nucleus to DDPRINT. These calls are issued either by ADADBS OPERCOM or Adabas Online System. Read and display calls are not logged.

ASSOSPACEWARN Command

ASSOSPACEWARN=*threshold1*[,*inc1*[,*threshold2*[,*inc2*[,*threshold3*[,*inc3*]]]]]

Use the `ASSOSPACEWARN` command to specify up to three pairs of threshold percentages and percentage change increments, expressed as percentages of the total Associator (ASSO) portion of the database. When ASSO space use reaches a specified threshold percentage, a space usage message is issued. In addition, when the ASSO space use changes by the increment specified for that threshold, additional space usage messages are issued.

Note: For the purposes of database space notifications, the space used in the Adabas database comprises all database blocks that have been assigned either to some common purpose (for example, DSST, PPT, DLOG area) or to a file in the database. Only space listed in the free space table (FST) is not calculated as used. In particular, free space allocated for some file but not yet used is calculated as used.

The following specifications can be made in the ASSOSPACEWARN command:

Parameter	Description
threshold#	A threshold, expressed as a percentage of the total Associator portion of the database. When ASSO space use exceeds or falls below any of the thresholds set, a space usage message is issued. The default is zero (0). If zero is specified for the first threshold setting, no space messages are issued. When more than one threshold is specified, the second threshold specification must be greater than the first. Likewise, the third threshold specification (if made) must be greater than the second.
inc#	An increment percentage, expressed as a percentage of the total ASSO portion of the database. When ASSO space use changes (increases or decreases) from the associated threshold by the amount given in this increment percentage, another space usage message is issued. The default is five percent (5).

Up to three threshold-increment pairs can be specified.

ASYTVS Command

ASYTVS={YES | NO}

Use the `ASYTVS` command to activate or deactivate asynchronous flushing of buffers based on volume serial number.

AUDCONNECT Command

ADADBS OPERCOM AUDCONNECT = { *dbid* | ALL }

```
Force a connection attempt with all related Auditing servers or databases or to the ↩
specified Adabas database ID.
```

Use the `AUDCONNECT` command to dynamically force a connection attempt to either a specific Auditing Server or Adabas database ID or to all related Auditing Server or Adabas database IDs.

One of the parameters of the `AUDCONNECT` command must be specified. There is no default. The parameters are described in the following table:

Parameter	Forces a reconnection attempt with
ALL	All known Auditing Server or Adabas database IDs
dbid	The specified Auditing Server or Adabas database ID.

AUTOINCASSOSIZE Command

AUTOINCASSOSIZE=*nnn*[B]

Use the `AUTOINCASSOSIZE` command to modify the setting of the `AUTOINCASSOSIZE` ADARUN parameter while the database is running. AUTOINCASSOSIZE specifies the size, in cylinders or blocks, by which the highest-numbered (last-defined) Associator (ASSO) data set will be increased by an automatic database increase process.

This parameter only has an effect if the AUTOINCASSOTHRESHOLD setting has a non-zero value. A numeric value without the suffix B indicates the increment is in cylinders. A numeric value followed by the letter B indicates the value specifies a number of blocks, which is then

rounded up to the next highest amount that fills the final track or other platform-dependent minimum space allocation unit.

If an automatic database increase process is started to increase the ASSO component of the database, the last ASSO container data set will be increased by the number of cylinders or blocks specified by AUTOINCASSOSIZE. (The number of cylinders or blocks added may be less, however, if the limit set by the AUTOINCASSOTOTAL parameter is reached.)

If there is insufficient space at the end of the dataset, and the AUTOINCASSOSIZE value is greater than the secondary space allocation for the dataset, a new disk extent equivalent to the AUTOINCASSOSIZE value will be allocated.

If there is insufficient space at the end of the dataset, and the AUTOINCASSOSIZE value is less than or equal to the secondary space allocation for the dataset, a new disk extent equivalent to the secondary space value will be allocated.

If the last ASSO container data set cannot be increased by the specified size, but only by a lesser amount, the automatic increase process will increase the data set by the amount of space that it can get. Note that one data set can span at most 59 volumes with up to 16 extents on each volume. When these limits are reached, the database should be increased manually by adding a new container data set with the ADADBS ONLADD utility function.

When 4-byte RABNs are used, valid values for this parameter range from zero (0) to 2147483647 cylinders or from zero (0) to 2147483647B blocks. When 3-byte RABNs are used, the limits range from zero (0) to 16777215 blocks or the equivalent in cylinders. A value of "0" or "0B" means that no automatic increase will be performed for ASSO space. The default value for this parameter is zero, which functionally disables auto-increase for ASSO space.

If both AUTOINCASSOSIZE and the secondary space amount are set to a small value, the automatic database increase process may be initiated multiple times and cause the container allocation to reach platform-dependent limits on extents and volumes. Once that occurs, there can be no more auto-increases until the container allocations are consolidated, which requires stopping all nuclei using the container. Alternatively, a manual increase can be used to add another container without stopping the Adabas nucleus.

AUTOINCASSOTHRESHOLD Command

AUTOINCASSOTHRESHOLD=*nn*

Use the `AUTOINCASSOTHRESHOLD` command to modify the setting of the `AUTOINCASSOTHRESHOLD` ADARUN parameter while the database is running. AUTOINCASSOTHRESHOLD specifies the Associator space threshold at which Adabas automatically initiates a database increase process.

This threshold is expressed as a percentage of used Associator (ASSO) space to total ASSO space available in the database. ASSO space is considered used if it is not listed as free in the Free Space Table (FST).

If the used ASSO space matches or exceeds the percentage defined by the specified AUTOINCASSOTHRESHOLD setting, Adabas automatically initiates a database increase process for ASSO space.

Note that the database increase process may take some time, the length of which is related to the size by which the database is increased (the value of the AUTOINCASSOSIZE setting). The AUTOINCASSOTHRESHOLD setting should be configured so that the increase process is given sufficient time to complete before the database fills up, assuming a reasonable space consumption rate.

Once a database increase process has been initiated, another automatic increase for ASSO space may be started only after the first process has finished and the ASSO space usage again (or still) matches or exceeds the specified threshold.

Valid values for this command are either zero (0) or integers ranging from 50 through 100. A value of zero means that no automatic increase will be performed for ASSO space. A value of 100 means that an automatic increase process will be initiated only when the ASSO space grows full (i.e., no free space is left at all). The default value is zero, which functionally disables auto-increase processes for ASSO space.

AUTOINCASSOTOTAL Command

AUTOINCASSOTOTAL=*nnn*[B]

Use the `AUTOINCASSOTOTAL` command to modify the setting of the `AUTOINCASSOTOTAL` ADARUN parameter while the database is running. AUTOINCASSOTOTAL specifies the maximum total size, in cylinders or blocks, of the entire Associator space of the database that is not to be exceeded by automatic database increase processes.

A numeric value followed by the letter B indicates the value specifies a number of blocks, which is then rounded up to the next highest amount that fills the final track or other platform-dependent minimum space allocation unit. A numeric value without a suffix B indicates the increment is in cylinders.

An automatic database increase process for ASSO space will not exceed the size specified by the AUTOINCASSOTOTAL setting. This setting can be used to limit the total amount of disk space used for the database, in order to protect the disk space remaining available for other databases and other users of the system. Note that this setting limits only automatic increases by Adabas; it

does not limit increasing the database manually using the ADADBS ADD, INCREASE, ONLADD or ONLINCREASE utility functions.

AUTOINCASSOTOTAL specifies the number of cylinders or blocks beyond which automatic database increase processes will not extend the ASSO space of the database. If the AUTOINCASSOTOTAL setting is less than the current number of cylinders in the ASSO space plus the AUTOINCASSOSIZE value, an automatic increase will increase the ASSO space only by the difference between the AUTOINCASSOTOTAL value and the current size – that is, by less than the increase requested by the AUTOINCASSOSIZE setting. If AUTOINCASSOTOTAL is not greater than the current number of cylinders in ASSO, no further automatic increases will be performed for ASSO space.

When 4-byte RABNs are used, valid values for this parameter range from zero (0) to 2147483647 cylinders or from zero (0) to 2147483647B blocks. When 3-byte RABNs are used, the limits range from zero (0) to 16777215 blocks or the equivalent in cylinders. A value of "0" or "0B" means that no automatic increase will be performed for ASSO space. The default value for this parameter is zero, which functionally disables auto-increase for ASSO space.

The total size of ASSO space may be constrained to less than the AUTOINCASSOTOTAL value for either of the following reasons.

- There may not be enough storage resources to extend to the specified limit.
- When using 4-byte RABNs, a component can have at most 2147483647 blocks, and for 3-byte RABNs the limit is 16777215 blocks.

AUTOINCDATASIZE Command

AUTOINCDATASIZE= *nnn* **[B]**

Use the `AUTOINCDATASIZE` command to modify the setting of the `AUTOINCDATASIZE` ADARUN parameter while the database is running. AUTOINCDATASIZE specifies the size, in cylinders or blocks, by which the highest-numbered (last-defined) Data Storage (DATA) data set will be increased by an automatic database increase process.

This parameter only has an effect if the AUTOINCDATATHRESHOLD has a non-zero value. A numeric value without the suffix B indicates the increment is in cylinders. A numeric value followed by the letter B indicates the value specifies a number of blocks, which is then rounded up to the next highest amount that fills the final track or other platform-dependent minimum space allocation unit.

If an automatic database increase process is started to increase the DATA component of the database, the last DATA container data set will be increased by the number of cylinders or blocks specified

by AUTOINCDATASIZE. (The number of cylinders or blocks added may be less, however, if the limit set by the AUTOINCDATATOTAL parameter is reached.)

If there is insufficient space at the end of the dataset, and the AUTOINCDATASIZE value is greater than the secondary space allocation for the dataset, a new disk extent equivalent to the AUTOINCDATASIZE value will be allocated.

If there is insufficient space at the end of the dataset, and the AUTOINCDATASIZE value is less than or equal to the secondary space allocation for the dataset, a new disk extent equivalent to the secondary space value will be allocated.

If the last DATA container data set cannot be increased by the specified size, but only by a lesser amount, the automatic increase process will increase the data set by the amount of space that it can get. Note that one data set can span at most 59 volumes with up to 16 extents on each volume. When these limits are reached, the database should be increased manually by adding a new container data set with the ADADBS ONLADD utility function.

When 4-byte RABNs are used, valid values for this parameter range from zero (0) to 2147483647 cylinders or from zero (0) to 2147483647B blocks. When 3-byte RABNs are used, the limits range from zero (0) to 16777215 blocks or the equivalent in cylinders. A value of "0" or "0B" means that no automatic increase will be performed for DATA space. The default value for this parameter is zero, which functionally disables auto-increase for DATA space.

If both AUTOINCDATASIZE and the secondary space amount are set to a small value, the automatic database increase process may be initiated multiple times and cause the container allocation to reach platform-dependent limits on extents and volumes. Once that occurs, there can be no more auto-increases until the container allocations are consolidated, which requires stopping all nuclei using the container. Alternatively, a manual increase can be used to add another container without stopping the Adabas nucleus.

AUTOINCDATATHRESHOLD Command

AUTOINCDATATHRESHOLD=*nn*

Use the `AUTOINCDATATHRESHOLD` command to modify the setting of the `AUTOINCDATATHRESHOLD` ADARUN parameter while the database is running. AUTOINCDATATHRESHOLD specifies the Data Storage space threshold at which Adabas automatically initiates a database increase process. This threshold is expressed as a percentage of used Data Storage (DATA) space to total Data Storage space available in the database. DATA space is considered used if it is not listed as free in the Free Space Table (FST).

If the used Data Storage space matches or exceeds the percentage defined by the specified AUTOINCDATATHRESHOLD setting, Adabas automatically initiates a database increase process for DATA space.

Note that the database increase process may take some time, the length of which is related to the size by which the database is increased (the value of the AUTOINCDATASIZE setting). The AUTOINCDATATHRESHOLD setting should be configured so that the increase process is given sufficient time to complete before the database fills up, assuming a reasonable space consumption rate.

Once a database increase process has been initiated, another automatic increase for DATA space may be started only after the first process has finished and the DATA space usage again (or still) matches or exceeds the specified threshold.

Valid values for this command are either zero (0) or integers ranging from 50 through 100. A value of zero means that no automatic increase will be performed for DATA space. A value of 100 means that an automatic increase process will be initiated only when the DATA space grows full (i.e., no free space is left at all). The default value is zero, which functionally disables auto-increase processes for DATA space.

AUTOINCDATATOTAL Command

AUTOINCDATATOTAL= *nnn***[B]**

Use the `AUTOINCDATATOTAL` command to modify the setting of the `AUTOINCDATATOTAL` ADARUN parameter while the database is running. AUTOINCDATATOTAL specifies the maximum total size, in cylinders or blocks, of the entire Data Storage space of the database that is not to be exceeded by automatic database increase processes.

A numeric value followed by the letter B indicates the value specifies a number of blocks, which is then rounded up to the next highest amount that fills the final track or other platform-dependent minimum space allocation unit. A numeric value without a suffix B indicates the increment is in cylinders.

An automatic database increase process for DATA space will not exceed the size specified by the AUTOINCDATATOTAL setting. This setting can be used to limit the total amount of disk space used for the database, in order to protect the disk space remaining available for other databases and other users of the system. Note that this setting limits only automatic increases by Adabas; it does not limit increasing the database manually using the ADADBS ADD, INCREASE, ONLADD or ONLINCREASE utility functions.

AUTOINCDATATOTAL specifies the number of cylinders or blocks beyond which automatic database increase processes will not extend the DATA space of the database. If AUTOINCDATATOTAL setting is less than the current DATA size plus the AUTOINCDATASIZE value, an automatic increase will increase the DATA space only by the difference between the AUTOINCDATATOTAL value and the current size – that is, by less than the increase requested by the AUTOINCDATASIZE setting. If the AUTOINCDATATOTAL setting is not greater than the current DATA size, no further automatic increases will be performed for DATA space.

When 4-byte RABNs are used, valid values for this parameter range from from zero (0) to 2147483647 blocks or the equivalent in cylinders. When 3-byte RABNs are used, the limits range from zero (0) to 16777215 blocks or the equivalent in cylinders. A value of "0" or "0B" means that no automatic increase will be performed for DATA space. The default value for this parameter is zero, which functionally disables auto-increase for DATA space.

The total size of DATA space may be constrained to less than the AUTOINCDATATOTAL value for either of the following reasons.

- There may not be enough storage resources to extend to the specified limit.
- When using 4-byte RABNs, a component can have at most 2147483647 blocks, and for 3-byte RABNs the limit is 16777215 blocks.

CANCEL Command

```
CANCEL [,GLOBAL]
```

Use the `CANCEL` command to cancel the Adabas session immediately. All command processing is immediately suspended. A pending autorestart will be in effect which in turn will cause the autorestart routine to be executed during the initialization of the next Adabas session. In nucleus cluster environments, the GLOBAL option can be used to cancel the Adabas session in all active cluster nuclei.

CLOGMRG Command

```
CLOGMRG={YES | NO}
```

Use the `CLOGMRG` command to dynamically modify the setting of the ADARUN CLOGMRG parameter.

The CLOGMRG command is only valid in cluster environments. It is global by definition and affects all nuclei in the cluster.

CLUFREEUSER Command

CLUFREEUSER=[,TNA= *max-time* **][,{UID | UIDPRFX} =** *userid* **][,FORCE][,GLOBAL]**

Note: 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.

Use the CLUFREEUSER command to delete leftover cluster user table elements (UTEs or PLXUSERs) in common storage that are no longer associated with user queue elements (UQEs) in a nucleus.

TNA
: TNA specifies a decimal number specifying the timeout value in seconds (*max-time*). PLXUSERs that are not used during the time specified may be deleted if other conditions are fulfilled. If TNA= is not specified, PLXUSERs may be deleted without regard to their recent use.

UID
: UID specifies a character string or hexadecimal byte string format user ID (*userid*) as follows:

cccccccc	where the argument is 1-8 letters, digits, or embedded '-' signs without surrounding apostrophes.
'*cccccccc*'	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' '.

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 eight 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. If the last eight bytes of a user's 28-byte communication ID match a specific user ID or user ID prefix (UIDPRFX parameter), that user's PLXUSER may be deleted if other conditions are fulfilled. If UID is not specified, PLXUSERs may be deleted regardless of their user IDs. "UID" may be abbreviated to "U".

UID and UIDPRFX are mutually exclusive; one or the other, but not both, can be specified.

UIDPRFX
: UIDPRFX accepts the same operands as UID. It operates in the same manner as UID, except that the operand is considered a prefix and there is no implicit padding. If the operand matches the initial bytes of the last eight bytes of the 28-byte communication ID, that PLXUSER may be deleted if other conditions are fulfilled. "UIDPRFX" may be abbreviated to "UP".

UID and UIDPRFX are mutually exclusive; one or the other, but not both, can be specified.

FORCE
: FORCE indicates that leftover PLXUSERs are to be deleted even if the users are due a response code 9 (ADARSP009), subcode 20. If FORCE is not specified, such PLXUSERs are not deleted.

Before using the FORCE parameter, ensure that the users owning the PLXUSERs to be deleted will not expect any of their transactions to remain open.

GLOBAL GLOBAL indicates that leftover PLXUSERs throughout the Adabas cluster are to be deleted if they are no longer associated with UQEs and are eligible according to the other specified parameters. Additionally and subject to the other rules, leftover PLXUSERs are deleted if their assigned nuclei have terminated since their last use. If GLOBAL is not specified, only PLXUSERs assigned to the local nucleus and used since the nucleus start are eligible for deletion.

CT Command

CT=*timeout-limit*

Use the `CT` command to dynamically override the ADARUN `CT` parameter value; that is, the maximum number of seconds that can elapse from the time an Adabas command has been completed until the results are returned to the user through interregion communication (which depends on the particular operating system being used). The minimum setting is 1; the maximum is 16777215.

In nucleus cluster environments, the `CT` command is global by definition and affects all nuclei in the cluster.

DATASPACEWARN Command

DATASPACEWARN=*threshold1***[,***inc1***[,***threshold2***[,***inc2***[,***threshold3***[,***inc3***]]]]]]**

Use the `DATASPACEWARN` command to specify up to three pairs of threshold percentages and percentage change increments, expressed as percentages of the total Data Storage (DATA) portion of the database. When Data Storage space use reaches a specified threshold percentage, a space usage message is issued. In addition, when the Data Storage space use changes by the increment specified for that threshold, additional space usage messages are issued.

Note: For the purposes of database space notifications, the space used in the Adabas database comprises all database blocks that have been assigned either to some common purpose (for example, DSST, PPT, DLOG area) or to a file in the database. Only space listed in the free space table (FST) is not calculated as used. In particular, free space allocated for some file but not yet used is calculated as used.

The following specifications can be made in the DATASPACEWARN command:

Parameter	Description
threshold#	A threshold, expressed as a percentage of the total Data Storage (DATA) portion of the database. When DATA space use exceeds or falls below any of the thresholds set, a space usage message is issued. The default is zero (0). If zero is specified for the first threshold setting, no space messages are issued. When more than one threshold is specified, the second threshold specification must be greater than the first. Likewise, the third threshold specification (if made) must be greater than the second.
inc#	An increment percentage, expressed as a percentage of the total DATA portion of the database. When DATA space use changes (increases or decreases) from the associated threshold by the amount given in this increment percentage, another space usage message is issued. The default is five percent (5).

Up to three threshold-increment pairs can be specified.

DAUQ Command

Use the `DAUQ` command to display the user queue elements of those users who have executed at least one Adabas command within the last 15 minutes. Up to 100 user queue elements will be displayed.

DCMDSTAT Command

Use the DCMDSTAT command to display statistics about the number of commands per command type. Message ADAN1C is issued to display the statistics.

DCQ Command

```
DCQ
```

Use the `DCQ` command to display all posted command queue elements (CQEs). The DCQ command displays each CQE's user ID, job name, and buffer length.

DDIB Command

```
DDIB
```

Use the `DDIB` command to display the data integrity block (DIB). This block contains entries indicating which Adabas utilities are active and the resources being used by each utility.

DDSF Command

```
DDSF
```

Use the `DDSF` command to display Adabas Delta Save Facility status. The `DDSF` command is only available if the Adabas nucleus is run with the parameter ADARUN DSF=YES.

DELCQ Command

```
DELCQ= seconds [,E= ALL |CICS] [,J= jobname ] [,GLOBAL]
```

Use the `DELCQ` command to explicitly request that the command queue be reviewed for unprocessed commands that have been waiting in the command queue for a specified period of time (or longer).

The `DELCQ` command can be issued any time during a nucleus session; it is not limited to instances when an online recovery process is ongoing. A one-time pass is made through the command queue once the `DELCQ` command is issued. Only user commands for which processing has not yet been started may be returned.

During these command queue reviews, any commands from the environments identified by the `DELCQ` command, with the job name identified in the command, and with wait times that match or exceed the timeout time specified by the `DELCQ` command will be returned to the user with response code 22 (ADARSP022), subcode 55. When a command in the command queue is returned to the user, no related information is written to the CLOG data set.

The following table describes the substitutions in the syntax of the `DELCQ` command.

Parameter	Required?	Description . . .	Default
seconds	Yes	Specify the timeout time, in seconds, for a command in the command queue. When the wait time for a command in the command queue reaches or exceeds this timeout value, the command is returned to the issuing user with response code 22 (ADARSP022), subcode 55. Valid values range from 1 through 86400 seconds. There is no default.	—
E	No	Specify ALL (to return commands in the command queue from all users) or CICS (to return commands only from CICS sessions).	ALL
J	No	Specify a job name to return only commands in the command queue with that job name. The default value is blanks; all commands exceeding the timeout limit coming from any job are returned.	(blanks)
GLOBAL	No	This parameter applies only to cluster nuclei. When specified, the timeout value specified by the `DELCQ` command applies to all active nuclei in the cluster. The default is not to use the GLOBAL parameter; the timeout value then applies only to the command queue for the current nucleus and is ignored by any other nuclei in the cluster.	—

DELUF Command

DELUF=*file-number*

Use the `DELUF` command to delete all users who are using the specified file. Any open transactions of the deleted users are backed out.

This command does not delete EXF or UTI users.

The `DELUF` command corresponds to the ADADBS OPERCOM STOPF=*file-number*,PURGE function.

Caution: If Adabas is running with ADARUN OPENRQ=NO (specifying that users are not required to issue an OP as the first command of the session), run the DELUF command only if you are certain that the users to be deleted are no longer active. If a user with an

open transaction is deleted, but then returns (by sending a command), no indication is given about the transaction backout. If the user continues the transaction, logical inconsistencies in the database could occur.

DELUI Command

DELUI= *timeout-limit*

Use the `DELUI` command to delete all users who have not executed a command during the specified time interval (in seconds). Any open transactions of the deleted users are backed out.

This command does not delete EXF or UTI users.

The `DELUI` command corresponds to the ADADBS OPERCOM STOPI=*`time`*,PURGE function.

Caution: If Adabas is running with ADARUN OPENRQ=NO (specifying that users are not required to issue an OP as the first command of the session), run the DELUI command only if you are certain that the users to be deleted are no longer active. If a user with an open transaction is deleted, but then returns (by sending a command), no indication is given about the transaction backout. If the user continues the transaction, logical inconsistencies in the database could occur.

DFILES Command

DFILES= { *n* | *n1* ,..., *n5* }

Use the `DFILES` command to display the number of users currently accessing, updating, or controlling either a specific file (*`n`*) or a series of individual files, specified in a list (*`n1`*`,...,`*`n5`*). A maximum of five files can be specified in the list. Users are displayed by job name and Adabas-assigned user ID, and listed by file. Only the first 5 users are displayed. If there are more than five users, the number of remaining users is also shown.

DFILESTAT Command

DFILESTAT

Use the `DFILESTAT` command to display statistics for the number of commands by file. Message ADAN1B is issued to display the statistics.

DFILUSE Command

DFILUSE=*nnn*

Use the `DFILUSE` command to display the number of total commands processed so far for the specified file during the current session. The count is displayed in the nucleus message ADAN33.

DHQ Command

DHQ

Use the `DHQ` command to display up to five hold queue elements (HQEs) and up to five users of each element.

DHQA Command

DHQA

Use the `DHQA` command to display up to 100 hold queue elements and up to 10 users of each element.

DLOCKF Command

Use the `DLOCKF` command to display the locked files. The type of lock is shown in the output. With the exception of advanced file locks, the type of lock appears in parentheses after the file number in console message ADAN30. Possible types are described in the following table:

Type	Description
F	The file was locked by LOCKF.
U	The file was locked by LOCKU.
X	The file was locked by LOCKX.

Files locked by ALOCKF (advanced file locks) are identified with the "A-Files=*nn*" version of the ADAN30 console message.

DMEMTB Command

The DMEMTB command can be issued against Adabas cluster nuclei to display the member state table, which provides information about all of the active nuclei in an Adabas cluster. The information produced by this command may be used by technical support.

The output produced by the DMEMTB command includes a third flag that indicates whether a system- or message-level XCF status monitoring exception has been encountered and whether a message was issued for the exception.

Sample Output

```
ADAX61 00226 2009-06-24 16:53:26 Member Status Table
ADAX61 00226 2009-06-24 16:53:26 Other members:        1
ADAX61 00226 2009-06-24 16:53:26   This system:        1
ADAX61 00226 2009-06-24 16:53:26
ADAX61 00226 2009-06-24 16:53:26         NUCID:    2,261
ADAX61 00226 2009-06-24 16:53:26   Internal ID:        2
ADAX61 00226 2009-06-24 16:53:26        System: DA2F
ADAX61 00226 2009-06-24 16:53:26      Job Name: USACHPD1
ADAX61 00226 2009-06-24 16:53:26   Flags 1/2/3: 11/00/00
ADAX61 00226 2009-06-24 16:53:26
ADAX61 00226 2009-06-24 16:53:26         NUCID:    2,262
ADAX61 00226 2009-06-24 16:53:26   Internal ID:        3
ADAX61 00226 2009-06-24 16:53:26        System: DA2F
ADAX61 00226 2009-06-24 16:53:26      Job Name: USACHPD2
ADAX61 00226 2009-06-24 16:53:26   Flags 1/2/3: D7/00/00
ADAN41 00226 2009-06-24 16:53:26 Function completed
```

Flags 1, 2, and 3 in this report are hexadecimal values that provide the following information about each nucleus:

Flag	Provides
1	The cluster member service summary, as a combination of one or more of the following values: ■ X'80' – All services are active ■ X'40' – Work data set is needed for recovery ■ X'10' – Accepting and processing intracluster commands ■ X'04' – Cache services are available ■ X'02' – Lock services are available ■ X'01' – Messaging services are available

Flag	Provides
	For example, a value of 97 for Flag 1 indicates that all services are active (X'80'), the nucleus is accepting and processing intracluster commands (X'10'), and cache (X'04'), lock (X'02') and messaging (X'01') services are all available (80+10+4+2+1=97).
2	The cluster member failure status, as a combination of one or more of the following values: ■ X'80' – All services are down ■ X'40' – Canceled by this nucleus ■ X'10' – Abnormal termination of intracluster command processing occurred ■ X'04' – Abnormal exit from cache occurred ■ X'02' – Abnormal exit from lock occurred ■ X'01' – Abnormal exit from the XCF group (messaging) occurred
3	The cluster XCF status monitoring summary, as a combination of one or more of the following values: ■ X'80' – A system-level XCF status monitoring exception occurred. ■ X'40' – A member-level XCF status monitoring exception occurred. ■ X'02' – A warning message was issued for the system-level XCF status monitoring exception. ■ X'01' – A warning message was issued for the member-level XCF status monitoring exception. For example, a value of 41 for Flag 3 indicates that a member-level XCF status monitoring exception occurred (X'40') and that a warning message (X'01') was issued for it (40+1=41).

DNC Command

Use the `DNC` command to display the number of posted command queue elements waiting to be selected.

DNFV Command

Use the `DNFV` command to display information about current file use.

This command provides information about the files in use at a particular point in time. It also indicates which other nucleus has exclusive file control if, for example, a user program receives a response 148 (ADARSP148), subcode 15.

Here is some sample output:

```
ADAI29 OPER CMD: DNFV

FNR=00008  A=Y U=Y ID=      CA=00000   CU=00001
```

where;

FNR=*nnnnn*	is the file number
A={Y | N}	(yes or no) indicates whether the file is used for access (read and/or search)
U={Y | N}	(yes or no) indicates whether the file is used for update. Use for update includes use for access.
ID=*nucid*	is the ID of the nucleus that owns the file lock, if the file is locked.
CA=*nnnnn*	is the number of users on this nucleus who are currently accessing this file.
CU=*nnnnn*	is the number of users on this nucleus who are currently updating this file.

DNH Command

DNH

Use the `DNH` command to display the number of ISNs currently in the hold queue.

DNU Command

DNU

Use the `DNU` command to display the number of current users.

DONLSTAT Command

DONLSTAT

Use the `DONLSTAT` command to display the status of each active reorder, invert online, or Event Replicator for Adabas initial-state process together with the process ID.

DPARM Command

DPARM

Use the `DPARM` command to display the Adabas session parameters currently in effect. Here is an example:

```
23:17:08 ADAI29 OPER CMD: DPARM
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AOSLOG=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 ARNWORKBUFFERS=270
23:17:08 ADAN16 61934 2016-02-24 23:17:06 ASSOSPACEWARN=50,10
23:17:08 ADAN16 61934 2016-02-24 23:17:06 ASYTVS=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUDITLOG=CONSOLE
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUTOCQENV=ALL
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUTOCQTIME=60
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUTOINCASSOSIZE=500
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUTOINCASSOTHRESHOLD=80
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUTOINCASSOTOTAL=10,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUTOINCDATASIZE=1000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUTOINCDATATHRESHOLD=80
23:17:08 ADAN16 61934 2016-02-24 23:17:06 AUTOINCDATATOTAL=20,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 CT=900
23:17:08 ADAN16 61934 2016-02-24 23:17:06 DATASPACEWARN=60,5
23:17:08 ADAN16 61934 2016-02-24 23:17:06 FMXIO=1
23:17:08 ADAN16 61934 2016-02-24 23:17:06 INDEXCROSSCHECK=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 INDEXUPDATE=ADVANCED
23:17:08 ADAN16 61934 2016-02-24 23:17:06 INTNAS=3,600
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LARGEPAGE=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LBP= 104,860,416
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LCP=10,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LDEUQP=5,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LFIOP=1
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LFP=10,240,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LI=819,200
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGCB=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGFB=NO
```

```
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGGING=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGIB=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGIO=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGMB=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGRB=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGSB=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGUX=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGVB=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGVOLIO=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LOGWARN=30
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LP=4,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LQ=220,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LRPL=104,857,600
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LS=29,952
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LU=64,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 LWP=1,200,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 MXTNA=7,200
23:17:08 ADAN16 61934 2016-02-24 23:17:06 MXTSX=7,200
23:17:08 ADAN16 61934 2016-02-24 23:17:06 MXTT=900
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NC=2,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NH=5,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NISNHQ=1,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NONDES=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NPLOGBUFFERS=9
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NQCID=10
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NSISN=51
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NT=30
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NU=10,000
23:17:08 ADAN16 61934 2016-02-24 23:17:06 NWORK1BUFFERS=9
23:17:08 ADAN16 61934 2016-02-24 23:17:06 READONLY=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 REFSTPRT=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 REVIEW=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 RIAFTERUPDATE=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 SECUID=WARN
23:17:08 ADAN16 61934 2016-02-24 23:17:06 SMF=YES
23:17:08 ADAN16 61934 2016-02-24 23:17:06 SMFDETAIL=NONE
23:17:08 ADAN16 61934 2016-02-24 23:17:06 SMFINTERVAL=5
23:17:08 ADAN16 61934 2016-02-24 23:17:06 SMFRECNO=255
23:17:08 ADAN16 61934 2016-02-24 23:17:06 SMFSUBSYS=CURR
23:17:08 ADAN16 61934 2016-02-24 23:17:06 SMF89=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 TLSCMD=900
23:17:08 ADAN16 61934 2016-02-24 23:17:06 TNAA=3,600
23:17:08 ADAN16 61934 2016-02-24 23:17:06 TNAE=3,600
23:17:08 ADAN16 61934 2016-02-24 23:17:06 TNAX=300
23:17:08 ADAN16 61934 2016-02-24 23:17:06 TT=300
23:17:08 ADAN16 61934 2016-02-24 23:17:06 UPDATECONTROL=NODELAY
23:17:08 ADAN16 61934 2016-02-24 23:17:06 UTIONLY=NO
23:17:08 ADAN16 61934 2016-02-24 23:17:06 V64BIT=YES
23:17:08 ADAN41 61934 2016-02-24 23:17:06 FUNCTION COMPLETED
```

Additional Adabas add-on products and nucleus components may display more parameters than are shown in this sample. For example, the DPARM command includes settings for the ADARUN

parameters related to Adabas Cluster Services and Adabas Parallel Services alert and timeout statistics.

DPPT Command

DPPT

Use the `DPPT` command to display the parallel participant table (PPT) block for a nucleus.

This command produces internal information for use by Software GmbH technical support.

Sample Output

```
ADAI29 Oper cmd: DPPT
ADAN24 00199 2009-01-27 22:31:35 Display PPT RABNs 00000083 to 000000A2
ADAN24 00199 2009-01-27 22:31:35
ADAN24 00199 2009-01-27 22:31:35          PPT RABN: 00000083
ADAN24 00199 2009-01-27 22:31:35 Number of entries: 05
ADAN24 00199 2009-01-27 22:31:35 Nucleus indicator: E2
ADAN24 00199 2009-01-27 22:31:35             NUCID: 0000
ADAN24 00199 2009-01-27 22:31:35    Session number: 0005
ADAN24 00199 2009-01-27 22:31:35   Last PLOG block: 00000000
ADAN24 00199 2009-01-27 22:31:35    PLOG block ind: 00
ADAN24 00199 2009-01-27 22:31:35  PPT Entry length: 0023
ADAN24 00199 2009-01-27 22:31:35          Entry ID: E6
ADAN24 00199 2009-01-27 22:31:35 Dataset=/SAGUID/PLX2/V7/WORKR1/
ADAN24 00199 2009-01-27 22:31:35  PPT Entry length: 0023
ADAN24 00199 2009-01-27 22:31:35          Entry ID: 61
ADAN24 00199 2009-01-27 22:31:35 Dataset=/ SAGUID /PLX2/V7/PLOGR1/
ADAN24 00199 2009-01-27 22:31:35  PPT Entry length: 0023
ADAN24 00199 2009-01-27 22:31:35          Entry ID: 62
ADAN24 00199 2009-01-27 22:31:35 Dataset=/ SAGUID /PLX2/V7/PLOGR2/
ADAN24 00199 2009-01-27 22:31:35  PPT Entry length: 0023
ADAN24 00199 2009-01-27 22:31:35          Entry ID: 41
ADAN24 00199 2009-01-27 22:31:35 Dataset=/ SAGUID /PLX2/V7/CLOGR1/
ADAN24 00199 2009-01-27 22:31:35  PPT Entry length: 0023
ADAN24 00199 2009-01-27 22:31:35          Entry ID: 42
ADAN24 00199 2009-01-27 22:31:35 Dataset=/ SAGUID /PLX2/V7/CLOGR2/
ADAN41 00199 2009-01-27 22:31:35 Function completed    ↩
```

DRES Command

```
DRES[,SESSION][,TS]
```

Use the DRES command to display the allocated pool space and the highest use level (high water mark) reached so far during the current interval by record count for the following resources:

- attached buffers (AB)
- command queue (CQ)
- format pool (FP)
- hold queue (HQ)
- pool for the table of ISNs (TBI)
- pool for the table of sequential commands (TBQ or TBLES)
- user queue (UQ)
- unique descriptor pool (DUQPOOL)
- security pool
- user queue file list pool
- work pool (WP)
- pool for global transaction IDs (XIDs; nonzero only with Adabas Transaction Manager)
- cluster block update redo pool (nonzero only for a cluster nucleus with ADARUN LRDP greater than zero)
- Work part 1 area (WKP1)

  Note: The maximum pool value of Work part 1 is derived from the LP parameter. It corresponds to the maximum number of blocks a transaction can spend on Work Part 1 before Adabas decides to back it out.

- Work part 2 area (WKP2)
- Work part 3 area (WKP3)

The actual values for the current interval are displayed in the nucleus message ADAN28, described in the *Adabas Messages and Codes Manual*.

SESSION: Display High Watermarks for the session

By default, the DRES operator command returns High Watermark statistics for the current interval, which are High Watermark statistics collected after the last refresh. If ADADBS REFRESHSTATS ALL or ADADBS REFRESHSTATS POOLUSAGE has not been invoked, High Watermark statics

for the session are returned, which are collected from the start of the nucleus. See ADADBS REFRESHSTATS for more information about invoking statistic refreshes.

When the SESSION parameter is specified, high watermark statistics collected from the start of the nucleus will be displayed in the nucleus with the message ADAN2H. However, if Session statistics have not been generated yet using ADADBS REFRESHSTATS, the message ADAN2J will be printed to inform the user and Interval statistics will be displayed using the ADAN28 message.

TS: Display timestamp information

When the TS parameter is specified, the time and date of the recorded high watermark will be included in the output. The date time will correspond to the interval, or the session, depending on which is specified. If only the TS parameter is specified, interval statistics with the corresponding timestamp will be displayed in the nucleus with the message ADAN2G. If the SESSION parameter is also specified, the session statistics with the corresponding timestamp will be displayed in the nucleus with the message ADAN2I. If the SESSION parameter is specified with the TS parameter, but the statistics are not available, the message ADAN2J will be displayed in the nucleus, and only interval data and timestamps will be displayed in the nucleus using the ADAN2G message.

DRPLSTAT Command

Use this command to display the replication-related statistics for an Adabas database (with replication turned on) or for an Event Replicator Server.

When issued against an Adabas database (with replication turned on), the statistics listed include:

- The total number of replication transactions completely processed.
- The current number of pending replicated transactions (transactions that have been committed, but not yet processed)
- The current number of incomplete transactions that will be replicated (but are not yet committed).

When issued against an Event Replicator Server, the statistics related to destinations, global values, and subscriptions in the database are listed. Replay Utility (ADARPL) statistics are also included.

The syntax for DRPLSTAT is:

DRPLSTAT [, {D = *destname* **| GLOBALS | S =** *sname* **| TOKENS}]**

The DRPLSTAT parameters are *always* optional and should be used only when the command is issued against an Event Replicator Server; the parameters are not valid when DRPLSTAT is issued for an Adabas database.

Note: Errors will occur if you attempt to run DRPLSTAT for an Adabas database using any of the parameters.

DRPLSTAT parameters are described in the following table. If you do not specify any of these parameters, replication-related statistics about all destinations, global values, and subscriptions in the Event Replicator Server are displayed.

Parameter	Displays replication-related statistics for:
D=*destname*	The specified destination (*destname*).
GLOBALS	All global values only.
S=*sname*	The specified subscription (*sname*).
TOKENS	ADARPL or ADALOD tokens. When a synchronized or replay-only request is submitted, a token is created in the Event Replicator Server nucleus. The TOKENS option of the DRPLSTAT facility allows you to view the details of the token, including the DBID, subscription, destination, start date, start time, end date, and end time in the token.

DSPACE Command

Use the DSPACE command to display space defined and used in ASSO and DATA. Message ADANS5 is issued with usage statistics.

DSTAT Command

Use the DSTAT command to display statistics about the current Adabas nucleus status.

When this command is issued against a running Event Replicator ADARPL job, the Replay Utility (ADARPL) statistics are displayed.

Note: After issuing a REFRESHSTATS, DSTAT displays the refreshed statistics.

DTARG Command

Use the `DTARG` command to display the system image currently servicing commands routed using the Entire Net-Work DBID target. This command can be issued against Adabas cluster nuclei.

The Entire Net-Work DBID target is used to route commands from users on remote systems with no active PLXCB. Such systems may be part of the z/OS sysplex or other Entire Net-Work nodes outside the sysplex.

DTH Command

Use the `DTH` command to display thread status.

DUQ Command

Use the `DUQ` command to display up to five active and inactive user queue elements.

Note: The ETID field has been enhanced to display the ETID in hex if it contains non-displayable characters, otherwise it will be displayed in character format. Trailing spaces are removed.

DUQA Command

DUQA

Use the `DUQA` command to display up to 100 user queue elements.

Note: The ETID field has been enhanced to display the ETID in hex if it contains non-displayable characters, otherwise it will be displayed in character format. Trailing spaces are removed.

DUQE Command

DUQE= X'*userid*'

Use the `DUQE` command to display the user queue element for the specified Adabas-assigned user ID. The user ID must be entered in hexadecimal format as follows:

```
DUQE=X'A3C1F2'
```

Do not enter a job name in place of the user ID.

Note: The ETID field has been enhanced to display the ETID in hex if it contains non-displayable characters, otherwise it will be displayed in character format. Trailing spaces are removed.

DUUQE Command

DUUQE

Use the `DUUQE` command to display utility user queue elements. Up to 100 user queue elements will be displayed.

Note: The ETID field has been enhanced to display the ETID in hex if it contains non-displayable characters, otherwise it will be displayed in character format. Trailing spaces are removed.

DVOLIO Command

DVOLIO

Use the `DVOLIO` command to display ASSO and DATA I/Os by volume serial (VOLSER) number. The number of reads and writes to each volume is displayed. Message ADAN1A is issued with these statistics.

In nucleus cluster environments, the NUCID=0 option can be used to display information for all active cluster nuclei. Information is displayed for each nucleus, one after the other.

DXCACHE Command

Issue the `DXCACHE` command against cluster nuclei to display the primary cache-related statistics. Statistics are displayed in messages on the console. The kinds of statistics displayed are described in *External Cache Statistics (Cluster Nucleus Only)*, in the *Adabas Cluster Services Operations Guide* or in *Global Cache Statistics (Cluster Nucleus Only)* in the *Adabas Parallel Services Operations Guide*.

The full set of statistics shown in this output is displayed only for users who have the selectable unit Adabas Online System (AOS) installed.

If you have installed only the demo version of AOS delivered with Adabas, only the statistics in the sections *Totals*, *Data Storage*, and *Normal Index* are displayed using this command.

DXFILE Command

DXFILE= *fnr* **[,** *fnr1* **] ...**

Issue the `DXFILE` command against cluster nuclei to display cache-related statistics for one to five files.

Statistics are displayed in messages on the console. The kinds of statistics displayed are described in *External Cache Statistics (Cluster Nucleus Only)*, in the *Adabas Cluster Services Operations Guide* or in *Global Cache Statistics (Cluster Nucleus Only)*, in the *Adabas Parallel Services Operations Guide*.

DXLOCK Command

```
DXLOCK
```

Issue the DXLOCK command against cluster nuclei to display lock-related statistics. Statistics are displayed in messages on the console. The kinds of statistics displayed are described in *External Lock Statistics (Cluster Nucleus Only)*, in the *Adabas Cluster Services Operations Guide* or in *Global Lock Statistics (Cluster Nucleus Only)*, in the *Adabas Parallel Services Operations Guide*.

DXMSG Command

```
DXMSG
```

The DXMSG command can be used to display the messaging performance statistics available for Adabas Cluster Services or Adabas Parallel Services. Statistics are displayed in messages on the console. The kinds of statistics displayed are described in *Messaging Statistics*, in the *Adabas Cluster Services Operations Guide* or *Adabas Parallel Services Operations Guide*.

The statistics produced by the DXMSG command can be used to determine the impact of messaging on system performance and to determine how to set the ADARUN MXMSG parameter.

DXSTAT Command

```
DXSTAT
```

Use the `DXSTAT` command against cluster nuclei to display all cache- and lock-related statistics as well as I/O statistics for WORK datasets in the cluster. Statistics are displayed in messages on the console. The kinds of statistics displayed are described in *External Cache Statistics (Cluster Nucleus Only)*, *External Lock Statistics (Cluster Nucleus Only)* and *WORK Data Set I/O Statistics (Cluster Nucleus Only)*, in the *Adabas Cluster Services Operations Guide* and in *Global Cache Statistics (Cluster Nucleus Only)*, *Global Lock Statistics (Cluster Nucleus Only)* and *WORK Data Set I/O Statistics (Cluster Nucleus Only)*, in the *Adabas Parallel Services Operations Guide*.

The full set of external cache statistics shown in this output is displayed only for users who have the selectable unit Adabas Online System (AOS) installed.

If you have installed only the demo version of AOS delivered with Adabas, only the external cache statistics in the sections *Totals*, *Data Storage*, and *Normal Index* are displayed using this command. All file cache statistics for files and all external lock statistics are displayed.

DZSTAT Command

```
DZSTAT [,ALL]
```

Use the DZSTAT command to an Adabas nucleus that was started with ADARUN parameter `ZIIP=YES` to display statistics about the execution of Adabas in TCB mode and SRB mode and about the CPU time consumed on System z Integrated Information Processors (zIIP) and general processors (GP).

The statistics displayed by DZSTAT correspond and are equivalent to the zIIP-related statistics at the end of the nucleus session statistics. See *Understanding the zIIP-Related Statistics* in the *Adabas for zIIP* documentation for detailed information about the various statistical figures.

If only `DZSTAT` is specified, statistics about the following will be displayed:

- current execution mode: "SRB" (execution on zIIP enabled) or "TCB" (execution on zIIP disabled)
- CPU time consumed on GPs and zIIPs by the entire Adabas address space
- CPU time consumed on GPs and zIIPs by the Workload Manager enclave created for the Adabas main task
- TCB/SRB mode switches and parallel requests to the TCB

If `DZSTAT,ALL` is specified, the following additional statistics will be displayed:

- "Extended statistics" about internal pause and release operations
- "Scheduling by type of work" statistics about reasons for switching modes or issuing parallel requests
- general processors (GP) and System z Integrated Information Processors (zIIP) in the system

Example

The following example output illustrates the zIIP-related statistics displayed by the DZSTAT command:

```
ADAN1Z dbid date time zIIP-related statistics:
ADAN1Z dbid date time  Adabas is executing in SRB mode
ADAN1Z dbid date time
ADAN1Z dbid date time CPU times for Adabas address space
ADAN1Z dbid date time  Total CPU time       =0:30:29.902
ADAN1Z dbid date time   Non-enclave GP times=0:01:01.257
ADAN1Z dbid date time   All enclave GP times=0:01:33.399
ADAN1Z dbid date time   Enclave zIIP times  =0:27:55.245
ADAN1Z dbid date time   Enclave zIIP time(%)=91.54
ADAN1Z dbid date time
ADAN1Z dbid date time CPU times for Adabas enclave
ADAN1Z dbid date time  Total enclave CPU    =0:29:28.644
ADAN1Z dbid date time   Enclave GP time     =0:01:33.398
ADAN1Z dbid date time   Enclave zIIP time   =0:27:55.245
ADAN1Z dbid date time   Enclave zIIP time(%)=94.71
ADAN1Z dbid date time
ADAN1Z dbid date time  Eligible zIIP CPU    =0:28:08.148
ADAN1Z dbid date time   Enclave zIIP time   =0:27:55.245
ADAN1Z dbid date time   zIIP on GP          =0:00:12.903
ADAN1Z dbid date time   zIIP on GP (%)      =0.76
ADAN1Z dbid date time
ADAN1Z dbid date time  Mode switches        =1,739
ADAN1Z dbid date time  Parallel requests    =2,224,887
ADAN1Z dbid date time   No free element     =0
ADAN1Z dbid date time   Per TCB pause       =2.38
```

The following additional statistics are displayed if the `ALL` parameter has been specified:

```
ADAN1Z dbid date time  Extended statistics
ADAN1Z dbid date time
ADAN1Z dbid date time   Pause   SRB         =261
ADAN1Z dbid date time   Release SRB         =0
ADAN1Z dbid date time   Pause   TCB         =933,077
ADAN1Z dbid date time   Release TCB         =933,338
ADAN1Z dbid date time   Pause   for wait    =4,725,414
ADAN1Z dbid date time   Release from wait   =5,020,326
ADAN1Z dbid date time
ADAN1Z dbid date time  SRB/TCB scheduling by type of work
ADAN1Z dbid date time
```

```
ADAN1Z dbid date time   EXCPs               =2,225,552
ADAN1Z dbid date time   Miscellaneous       =115
ADAN1Z dbid date time   Operator commands   =4
ADAN1Z dbid date time   Sequential writes   =209
ADAN1Z dbid date time   Timer services      =328
ADAN1Z dbid date time   User exit 8         =418
ADAN1Z dbid date time  Number of GPs        =2
ADAN1Z dbid date time  Number of zIIPs      =1
ADAN1Z dbid date time   zIIP SMT threads    =2
ADAN1Z dbid date time  Normalization factor =10.97
```

See *Understanding the zIIP-Related Statistics* in the *Adabas for zIIP* documentation for detailed information about the various statistical figures.

FEOFAL Command

```
FEOFAL [,GLOBAL]
```

Use the `FEOFAL` command to close the current dual or multiple command log and switch to another audit log. This command is valid only if dual or multiple command logging is in effect and only for Adabas Audit Servers.

In nucleus cluster environments, the `GLOBAL` option can be used to close and switch dual or multiple audit logs in all active cluster nuclei.

FEOFCL Command

```
FEOFCL [,GLOBAL]
```

Use the `FEOFCL` command to close the current dual or multiple command log and switch to another command log. This command is valid only if dual or multiple command logging is in effect.

In nucleus cluster environments, the `GLOBAL` option can be used to close and switch dual or multiple command logs in all active cluster nuclei.

FEOFPL Command

```
FEOFPL [,GLOBAL]
```

Use the `FEOFPL` command to close the current dual or multiple data protection log and switch to the another protection log. This command is valid only if dual or multiple data protection logging is in effect.

In nucleus cluster environments, the `GLOBAL` option can be used to close and switch dual or multiple protection logs in all active cluster nuclei.

FMXIO Command

```
FMXIO=nn
```

Use the `FMXIO` command to dynamically modify the setting of the ADARUN `FMXIO` parameter.

HALT Command

```
HALT [,GLOBAL]
```

Use the `HALT` command to perform a BT (back out transaction) for each active ET logic user's session, then terminate the Adabas session. No dumps are produced by `HALT`.

In nucleus cluster environments, the `GLOBAL` option can be used to perform a BT for all active ET logic user sessions and terminate the Adabas session in all active cluster nuclei.

INDEXCROSSCHECK Command

```
INDEXCROSSCHECK = { YES | NO }
```

Use the `INDEXCROSSCHECK` command to turn index cross-checking on and off. Index cross-checking is a process Adabas uses to ensure that a data storage record it is reading actually matches the descriptor value in the index by which the record was found. When index cross-checking is turned on, Adabas extracts the descriptor value from the record being read and compares it to the value used for positioning in the index. If there is a mismatch, Adabas returns response code 175 (ADARSP175). Valid values are "YES" and "NO".

If "YES" is specified (the default value), Adabas extracts the descriptor value from the record being read and compares it to the value used for positioning in the index. If the two values are not equal, Adabas checks for a parallel update command on the same record and retries after that update has finished. If there is no parallel update command, Adabas returns response code 175 (ADARSP175) to indicate an inconsistency between the data storage and the index.

If "NO" is specified, Adabas does not extract the descriptor value from the record, but immediately checks for a parallel update command on the same record. If there is one, even if it does not update the descriptor in question, Adabas waits until the update has finished and then retries. With this parameter setting, Adabas does not validate the index against the data storage record and does not return response code 175 (ADARSP175).

Note: Attempts to change the `INDEXCROSSCHECK` setting (via this command) in a cluster environment will be rejected. In a cluster nucleus (CLUSTER=SYSPLEX or CLUSTER=LOCAL), Adabas sets INDEXCROSSCHECK to "YES", overriding any user-specified INDEXCROSSCHECK value, and validates the descriptor value from the index against the corresponding value from the data storage.

Setting INDEXCROSSCHECK to "NO" may, under certain circumstances, reduce the CPU time required for ensuring that the Data Storage record being read matches the descriptor value by which the record was found. This applies in particular to cases where neither the descriptor itself nor (if it is a special descriptor) its parent field(s) occur in the format buffer specifying the fields to be returned and where the descriptor or one of its parents were defined near the end of the FDT.

LICREFRESH Command

LICREFRESH

Use the `LICREFRESH` command to:

- reload the license module or reread the license file from the library identified by the DDLIC JCL statement in the startup job for the nucleus
- display and check the license identified by the DDLIC JCL statement in the startup job for the nucleus.

In cluster environments, the LICREFRESH command must be run on each nucleus in the cluster.

LOCKF Command

LOCKF=*n*

Use the `LOCKF` command to lock the specified file. No use of the specified file is possible at any security level.

LOCKU Command

LOCKU=*n*

Use the `LOCKU` command to lock the specified file for all non-utility users. Adabas utilities can use the specified file normally.

LOCKX Command

LOCKX=*n*

Use the `LOCKX` command to lock the specified file for all users except EXU or EXF users. EXU and EXF users can use the file normally. The lock is released automatically when an EXU user issues an OP command.

LOGGING Command

LOGGING

Use the `LOGGING` command to start command logging.

LOGCB Command

LOGCB

Use the `LOGCB` command to start logging of the Adabas control block for each command logged.

LOGFB Command

LOGFB

Use the `LOGFB` command to start logging of the Adabas format buffer for each command logged.

LOGIB Command

```
LOGIB
```

Use the LOGIB command to start logging of the Adabas ISN buffer for each command logged.

LOGIO Command

```
LOGIO
```

Use the `LOGIO` command to start logging of Adabas I/O activity for each command logged.

LOGRB Command

```
LOGRB
```

Use the `LOGRB` command to start logging of the Adabas record buffer for each command logged.

LOGSB Command

```
LOGSB
```

Use the `LOGSB` command to start logging of the Adabas search buffer for each command logged.

LOGUX Command

```
LOGUX
```

Use the `LOGUX` command to start logging of user exit B data for inclusion in the CLOG record.

LOGVB Command

```
LOGVB
```

Use the `LOGVB` command to start logging of the Adabas value buffer for each command logged.

LOGVOLIO Command

```
LOGVOLIO
```

Use the `LOGVOLIO` command to initiate writing of the extended I/O list to the command log.

LOGWARN Command

```
LOGWARN = { seconds | 0 }
```

Use the `LOGWARN` command to specify how often the PLOG and CLOG status is checked and resulting alert messages are produced. Valid values range from zero (0) through 2147483647 seconds. The default is 0, indicating that no PLOG or CLOG status checking occurs and no corresponding alert messages are produced. If a non-zero value is specified for LOGWARN, a valid user exit 2 or user exit 12 must also be specified.

MXCANCEL Command

The MXCANCEL command can be used against cluster nuclei to dynamically change (while the Adabas nucleus is running) the setting of the `MXCANCEL` ADARUN parameter for a nucleus. This command can be used to specify the maximum number of seconds for which one cluster nucleus waits for the termination of another nucleus in the cluster that it has canceled for not responding to an internal, intracluster command. Valid values range from 15 through 2147483647.

If the new MXCANCEL value is less than or equal to the current MXCANCELWARN setting, the MXCANCELWARN setting is automatically set to zero (0). (You can then use the MXCANCEL-WARN command to dynamically set the MXCANCELWARN setting appropriately.)

For more information about the MXCANCEL setting for a nucleus, read about the ADARUN `MXCANCEL` parameter in the *Adabas Operations Manual*.

MXCANCELWARN Command

MXCANCELWARN

The MXCANCELWARN command can be used to dynamically change (while the Adabas nucleus is running) the setting of the `MXCANCELWARN` ADARUN parameter for a nucleus. This command can be used to specify the number of seconds after an intracluster cancellation request is made when the cluster nucleus requesting the cancellation should issue warnings about the inability of the canceled peer nucleus to terminate. Valid values can be zero (0) or an integer ranging from 4 through the value of MXCANCEL minus 1 (MXCANCEL-1).

The value of the MXCANCELWARN setting must be less than the value of the MXCANCEL setting. If MXCANCELWARN is set to a value greater than or equal to the value specified for MXCANCEL, the value is rejected. If MXCANCELWARN is set to zero, no warnings are issued about canceled cluster nuclei that are slow to terminate.

For more information about the MXCANCELWARN setting for a nucleus, read about the ADARUN `MXCANCELWARN` parameter in the *Adabas Operations Manual*.

MXMSG Command

The MXMSG command can be used to dynamically change (while the Adabas nucleus is running) the setting of the `MXMSG` ADARUN parameter for a cluster nucleus. This command can be used to specify the maximum number of seconds in which one cluster nucleus waits for the response of another nucleus in the cluster to an internal, intracluster command. Valid values range from 15 through 32767.

If the new MXMSG value is less than or equal to the current MXMSGWARN setting, the MXMSGWARN setting is automatically set to zero (0). (You can then use the MXMSGWARN command to dynamically set the MXMSGWARN setting appropriately.)

For more information about the MXMSG setting for a nucleus, read about the ADARUN `MXMSG` parameter in the *Adabas Operations Manual*.

MXMSGWARN Command

MXMSGWARN

The MXMSGWARN command can be used to dynamically change (while the Adabas nucleus is running) the setting of the `MXMSGWARN` ADARUN parameter for a cluster nucleus. This command can be used to specify the number of seconds after an intracluster request is made when the cluster nucleus issuing the intracluster request should issue warnings about the failure of the target cluster nucleus to respond. Valid values can be zero (0) or an integer ranging from 4 through the value of MXMSG minus 1 (MXMSG-1).

The value of MXMSGWARN must be less than the value of MXMSG. If MXMSGWARN is set to a value greater than or equal to the value specified for MXMSG, the value is rejected. If MXMSGWARN is set to zero, no warnings are issued about cluster nuclei that are slow to respond.

For more information about the MXMSGWARN setting for a nucleus, read about the ADARUN `MXMSGWARN` parameter in the *Adabas Operations Manual*.

MXSTATUS Command

The MXSTATUS command can be used to dynamically change (while the Adabas nucleus is running) the setting of the `MXSTATUS` ADARUN parameter for a cluster nucleus. This command can be used to specify the number of seconds between member-level XCF status monitoring heartbeats. Valid values range from 15 through 21474836.

Note: The MXSTATUS command is only used for Adabas Cluster Services and not for Adabas Parallel Services. Adabas Parallel Services does not use XCF and ignores the setting.

The MXSTATUS command can only be used to change the value of the `MXSTATUS` parameter if the ADARUN MXSTATUS parameter is not zero (0). This command cannot be used to turn status monitoring on (by specifying a nonzero value if the ADARUN `MXSTATUS` setting is zero) or to turn it off (by specifying a value of zero when the ADARUN `MXSTATUS` setting is nonzero). However, setting MXSTATUS to a very high value is effectively the same as turning status monitoring off (monitoring will occur very infrequently at least).

For more information about the MXSTATUS setting for a nucleus, read about the ADARUN `MXSTATUS` parameter in the *Adabas Operations Manual*.

MXWTOR Command

```
MXWTOR
```

The optional MXWTOR command can be used to dynamically change (while the Adabas nucleus is running) the setting of the `MXWTOR` ADARUN parameter for a cluster nucleus. This command can be used to specify the maximum number of seconds that a cluster nucleus should wait for an operator response to a self-terminating operator query. Valid values range from 15 through 64800 or zero (0).

If MXWTOR is set to zero, the operator query does not occur (no ADAX9J messages are written) and no operator intervention is expected.

For more information about the MXWTOR setting for a nucleus, read about the ADARUN `MXWTOR` parameter in the *Adabas Operations Manual*.

NOLOGGING Command

```
NOLOGGING
```

Use the `NOLOGGING` command to stop or prevent command logging.

NOLOGCB Command

```
NOLOGCB
```

Use the `NOLOGCB` command to stop or prevent logging of the Adabas control block.

NOLOGFB Command

```
NOLOGFB
```

Use the `NOLOGFB` command to stop or prevent logging of the Adabas format buffer.

NOLOGIB Command

```
NOLOGIB
```

Use the `NOLOGIB` command to stop or prevent logging of the Adabas ISN buffer.

NOLOGIO Command

```
NOLOGIO
```

Use the `NOLOGIO` command to stop or prevent logging of Adabas I/O activity.

NOLOGRB Command

```
NOLOGRB
```

Use the `NOLOGRB` command to stop or prevent logging of the Adabas record buffer.

NOLOGSB Command

```
NOLOGSB
```

Use the `NOLOGSB` command to stop or prevent logging of the Adabas search buffer.

NOLOGUX Command

```
NOLOGUX
```

Use the `NOLOGUX` command to stop logging of user exit B data for inclusion in the CLOG record.

NOLOGVB Command

```
NOLOGVB
```

Use the `NOLOGVB` command to stop or prevent logging of the Adabas value buffer.

NOLOGVOLIO Command

```
NOLOGVOLIO
```

Use the `NOLOGVOLIO` command to terminate (stop) writing the extended I/O list to the command log.

NWCONNECT Command

```
NWCONNECT
```

Use the NWCONNECT command to retry establishing the Entire Net-Work target associated with the nucleus's IDT entry. For classic Adabas nuclei, this is the DBID; for Adabas Cluster Services or Adabas Parallel Services, this is the nucleus ID (NUCID).

During nucleus initialization, certain Entire Net-Work errors may have prevented the target from being established, resulting in message ADAM76. Other errors may prevent Entire Net-Work from defining the target when it is started after the nucleus is initialized. The Entire Net-Work operator command DISPLAY TARGETS may be used to see whether the DBID target is known to Entire Net-Work.

Note: Adabas Cluster Services and Adabas Parallel Services maintain the Entire Net-Work DBID target using a different protocol. The nucleus will attempt to reestablish the DBID target automatically at timed intervals.

ONLRESUME Command

```
ONLRESUME=X'identifier'
```

Use the `ONLRESUME` command to resume a previously suspended online reorder, invert, or Event Replicator for Adabas initial-state process.

ONLSTOP Command

```
ONLSTOP=X'identifier'
```

Use the `ONLSTOP` command to stop an online reorder, invert, or Event Replicator for Adabas initial-state process cleanly. The process continues up to its next interrupt point in order to produce a consistent state, and then terminates after performing all necessary cleanup.

ONLSUSPEND Command>

```
ONLSUSPEND=X'identifier'
```

Use the `ONLSUSPEND` command to suspend an online reorder, invert, or Event Replicator for Adabas initial-state process. The process continues up to its next interrupt point in order to produce a consistent state, performs a command throwback, and enters a state where it cannot be selected for processing. This command is useful if the online process is consuming too much of the nucleus resources.

RALOCKF Command

```
RALOCKF=n
```

Use the `RALOCKF` command to release the advance-lock on the specified file (see ALOCKF command) without running the utility.

This command is available in cluster and non-cluster environments.

RALOCKFA Command

```
RALOCKFA
```

Use the `RALOCKFA` command to release the advance-lock on all files for which it has been set (see ALOCKF command) without running the utility.

This command is available in cluster and non-cluster environments.

RDUMPST Command

```
RDUMPST
```

Use the `RDUMPST` command to terminate online dump status. This command is normally used if online execution of the ADASAV utility has terminated abnormally.

READONLY Command

```
READONLY={YES | NO}
```

Use the `READONLY` command to switch READONLY status on or off. A value of "YES" switches it on; a value of NO switches it off.

REFSTPRT Command

```
REFSTPRT={YES | NO}
```

Use the `REFSTPRT` command to alter the setting of the ADARUN REFSTPRT parameter while the nucleus is active. This operator command allows you to dynamically specify whether printing of statistics is done before an ADADBS REFRESHSTATS utility function run or before interval statistics have been refreshed using Adabas Online System. For more information, read *REFSTPRT Parameter: Print Statistics after Refresh Statistics*.

REVIEW Command

```
REVIEW={ NO | LOCAL | hub-id }
```

Use the `REVIEW` command to:

- deactivate Adabas Review,
- change from hub mode to local mode, or

- to specify or change the Adabas Review hub with which a nucleus communicates.

RPLCHECK Command

RPLCHECK

Use this command to perform the replication cross-check function for all active databases known (defined in one or more subscriptions) to the Event Replicator Server. When this command is run using the ADADBS OPERCOM function, the information about the cross-check function is printed to the ADADBS DDDRUCK data set. The information printed by ADADBS is the same as the information printed by the Event Replicator Server during the cross-check process initiated by the RPLCHECK operator command.

Note: This command can only be issued against an Event Replicator Server; it is not valid for the Adabas nucleus. If this command is issued against a database that is not an Event Replicator Server, error messages result.

RPLCONNECT Command

RPLCONNECT = { *dbid* | ALL }

Use this command to dynamically force a connection attempt to either a specific Event Replicator Server or Adabas database ID or to all related Event Replicator Server or Adabas database IDs.

One of the parameters of the RPLCONNECT command must be specified. There is no default. The parameters are described in the following table:

Parameter	Forces a reconnection attempt with:
ALL	All known Event Replicator Server or Adabas database IDs
dbid	The specified Event Replicator Server or Adabas database ID.

RPLCONNECTCOUNT Command

RPLCONNECTCOUNT = *nnn*

Use this command to dynamically specify the number of connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails (response 148 is issued).

For *nnn*, specify a valid integer ranging from zero (0) through 2147483647. A value of zero indicates that no connection attempts should occur; a value of zero makes the most sense in situations where the Adabas database and the Event Replicator Server execute together on the same logical partition (LPAR). If the Adabas database and the Event Replicator Server execute on different LPARs, however, setting a real value using this command helps avoid errors that might arise if network problems occur because the network is not started or a network connection between the Adabas database and the Event Replicator Server is lost.

RPLCONNECTINTERVAL Command

RPLCONNECTINTERVAL = *nnn*

Use this command to dynamically specify the interval between connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails (response 148 is issued).

For *nnn*, specify the number of seconds for the interval, ranging from zero (0) through 2147483647 seconds. A value of zero indicates that no connection attempts should occur; a value of zero makes the most sense in situations where the Adabas database and the Event Replicator Server execute together on the same logical partition (LPAR). If the Adabas database and the Event Replicator Server execute on different LPARs, however, setting a real value using this command helps avoid errors that might arise if network problems occur because the network is not started or a network connection between the Adabas database and the Event Replicator Server is lost.

SECUID Command

SECUID = { ALLOW | REQUIRE | WARN }

Use the SECUID command to alter the ADARUN SECUID parameter while the nucleus is active. For complete information about the SECUID ADARUN parameter, read *SECUID Parameter: Security System User ID Requirement Level*, in the *Adabas Operations Manual*.

SMFDETAIL Command

SMFDETAIL = { ALL | NONE | *section-name-list* **}**

Adabas SMF records can contain a variable set of detail sections in Interval and Termination records (subtypes 2 and 3). This command allows you to override the setting of the SMFDETAIL ADARUN parameter defined for this Adabas session. Using this command you can select the detail section types in Interval and Termination records that should be included in the Adabas SMF records. Unlike the SMFDETAIL ADARUN parameter, the value or values you specify for the SMFDETAIL command do *not* need to be enclosed in parentheses. If you specify NONE or ALL, they should be specified alone. However, you can specify one or more of the other detail section names (CMD, CSHB, CSHF, CSHG, CSHP, FILE, IODD, LOCK, MSGB, MSGC, MSGH, PARM, STG, THRD, ZIIP or USER) in one SMFDETAIL parameter, separating each value with a comma.

The following table describes the meaning of the possible detail section names that can be used in the SMFDETAIL command:

Detail Section Name	Description
ALL	Generate all possible detail sections in the Adabas SMF records. If this value is specified, no others can be specified on the SMFDETAIL parameter.
CMD	Generate Adabas command activity detail sections in the Adabas SMF records.
CSHB[1]	Generate global cache activity by block detail sections in the Adabas SMF records.
CSHF[1]	Generate global cache activity by Adabas file number detail sections in the Adabas SMF records.
CSHG[1]	Generate global cache activity detail sections in the Adabas SMF records.
CSHP[2]	Generate Adabas Parallel Services cache activity detail sections in the Adabas SMF records.
FILE	Generate Adabas file activity detail sections in the Adabas SMF records.
IODD	Generate I/O activity by DD statement detail sections in the Adabas SMF records.
LOCK[1]	Generate global lock activity detail sections in the Adabas SMF records.
MSGB[1]	Generate internucleus messaging control block activity detail sections in the Adabas SMF records.
MSGC[1]	Generate internucleus messaging count detail sections in the Adabas SMF records.
MSGH[1]	Generate internucleus messaging service time histogram detail sections in the Adabas SMF records.
NONE	Generate no detail sections in the Adabas SMF records. If this value is specified, no others can be specified on the SMFDETAIL parameter.
PARM	Generate ADARUN parameter value detail sections in the Adabas SMF records.

Detail Section Name	Description
STG	Generate Adabas storage pool detail sections in the Adabas SMF records.
THRD	Generate thread activity detail sections in the Adabas SMF records.
ZIIP	Generate zIIP statistics detail sections in the Adabas SMF records.
USER	Generate user-defined details sections in the Adabas SMF records. If USER is specified, a value for the UEXSMF parameter should also be specified to identify the user exit to be invoked to generate the user-defined detail section.

[1] *Available only in cluster environments (Adabas Cluster Services or Adabas Parallel Services must be installed).*

[2] *Available only in cluster environments with Adabas Parallel Services installed.*

SMFDETAILADD Command

SMFDETAILADD = *section-name-list*

Adabas SMF records can contain a variable set of detail sections in Interval and Termination records (subtypes 2 and 3). This command allows you to add specific detail sections to your Adabas SMF records for the running Adabas session. The sections you specify are added to those already specified for the Adabas sesssion. If more than one section is listed on an SMFDETAILADD command, separate the section names with commas.

The following table describes the meaning of the possible detail section names that can be used in the SMFDETAILADD command:

Detail Section Name	Description
CMD	Generate Adabas command activity detail sections in the Adabas SMF records.
CSHB[1]	Generate global cache activity by block detail sections in the Adabas SMF records.
CSHF[1]	Generate global cache activity by Adabas file number detail sections in the Adabas SMF records.
CSHG[1]	Generate global cache activity detail sections in the Adabas SMF records.
CSHP[2]	Generate Adabas Parallel Services cache activity detail sections in the Adabas SMF records.
FILE	Generate Adabas file activity detail sections in the Adabas SMF records.
IODD	Generate I/O activity by DD statement detail sections in the Adabas SMF recods.
LOCK[1]	Generate global lock activity detail sections in the Adabas SMF records.
MSGB[1]	Generate internucleus messaging control block activity detail sections in the Adabas SMF records.

Detail Section Name	Description
MSGC[1]	Generate internucleus messaging count detail sections in the Adabas SMF records.
MSGH[1]	Generate internucleus messaging service time histogram detail sections in the Adabas SMF records.
PARM	Generate ADARUN parameter value detail sections in the Adabas SMF records.
STG	Generate Adabas storage pool detail sections in the Adabas SMF records.
THRD	Generate thread activity detail sections in the Adabas SMF records.
ZIIP	Generate zIIP statistics detail sections in the Adabas SMF records.
USER	Generate user-defined details section sin the Adabas SMF records. If USER is specified, a value for the UEXSMF ADARUN parameter can also be specified to identify the user exit that should be used to generate the user-defined detail section.

[1] *Available only in cluster environments (Adabas Cluster Services or Adabas Parallel Services must be installed).*

[2] *Available only in cluster environments with Adabas Parallel Services installed.*

SMFDETAILDEL Command

SMFDETAILDEL = *section-name-list*

Adabas SMF records can contain a variable set of detail sections in Interval and Termination records (subtypes 2 and 3). This command allows you to remove specific detail sections from your Adabas SMF records for the running Adabas session. The sections you specify are removed from those already specified for the Adabas sesssion. If more than one section is listed on an SMFDETAILDEL command, separate the section names with commas.

The following table describes the meaning of the possible detail section names that can be used in the SMFDETAILDEL command:

Detail Section Name	Deletes
CMD	Adabas command activity detail sections in the Adabas SMF records.
CSHB[1]	Global cache activity by block detail sections in the Adabas SMF records.
CSHF[1]	Global cache activity by Adabas file number detail sections in the Adabas SMF records.
CSHG[1]	Global cache activity detail sections in the Adabas SMF records.
CSHP[2]	Adabas Parallel Services cache activity detail sections in the Adabas SMF records.
FILE	Adabas file activity detail sections in the Adabas SMF records.

Detail Section Name	Deletes
IODD	I/O activity by DD statement detail sections in the Adabas SMF records.
PARM	ADARUN parameter value detail sections in the Adabas SMF records.
LOCK[1]	Global lock activity detail sections in the Adabas SMF records.
MSGB[1]	Internucleus messaging control block activity detail sections in the Adabas SMF records.
MSGC[1]	Internucleus messaging count detail sections in the Adabas SMF records.
MSGH[1]	Internucleus messaging service time histogram detail sections in the Adabas SMF records.
STG	Adabas storage pool detail sections in the Adabas SMF records.
THRD	Thread activity detail sections in the Adabas SMF records.
ZIIP	zIIP statistics detail sections in the Adabas SMF records.
USER	User-defined details section sin the Adabas SMF records.

[1] *Available only in cluster environments (Adabas Cluster Services or Adabas Parallel Services must be installed).*

[2] *Available only in cluster environments with Adabas Parallel Services installed.*

SMFINTERVAL Command

SMFINTERVAL = { GLOBAL | NONE | SUBSYS | *minutes* **}**

Adabas SMF records can be generated at specific intervals. The SMFINTERVAL command enables and disables interval recording and specifies the interval or the source from which the interval can be derived.

The interval can be one of several values from z/OS specifications or an explicit interval in minutes. Interval records contain a product detail section and whatever detail sections are selected in the SMFDETAIL parameter, except for the ADARUN parameter detail section which are not included in SMF interval records.

The following table describes the meaning of the specifications that can be used in the SMFINTERVAL command:

Valid Value	Description
GLOBAL	Generate all interval SMF records at the rate established by the INTVL and SYNCVAL SMF parameters in PARMLIB member SMFPRM*xx*.
NONE	Generate no interval SMF records.
SUBSYS	Generate interval SMF records at the rate specified in PARMLIB member SMFPRM*xx* for the subsystem identified in the ADARUN SMFSUBSYS parameter.

Valid Value	Description
minutes	Generate interval SMF records at the specified interval, in minutes. Valid minute values can range from 1 through 9999..

SMFRECNO Command

SMFRECNO = *record-number*

Use this command to override the SMFRECNO setting currently specified for this Adabas session (either through the SMFRECNO ADARUN parameter or by another SMFRECNO command). This command sets the Adabas SMF record number used for user-defined SMF records. IBM designates the range of SMF numbers from 128 through 255 for user-defined records, so valid values range from 128 through 255.

SMFSUBSYS Command

SMFSUBSYS = { CURRENT | *SMF-subsystem-name* **}**

This command overrides any other SMFSUBSYS setting active in the Adabas session (either via the ADARUN SMFSUBSYS parameter or another SMFSUBSYS command). It allows you to associate the Adabas SMF records with an IBM or user-defined subsystem. The z/OS systems staff can provide different sets of SMF system parameters for IBM-defined subsystems. Up to eight user-specified subsystems can be defined, each with their own set of system parameters. This parameter allows you to optionally identify one of these subsystem names to be used with Adabas SMF recording, if the one you want is different from the current one.

The possible values for this command are:

- CURRENT: This is the default value and associates the Adabas SMF record with the IBM-defined subsystem under which the Adabas nucleus is active. The subsystem may be TSO for a nucleus running in a TSO session, STC for a starting task or the name of the job entry subsystem under which a batch job is running, JES2 or JES3.
- An explicit subsystem name can be specified. It may be either an IBM-defined name or a user-defined name from PARMLIB member SMFPRM*xx*. The subsystem name is one to four characters long. The first character must be alphabetic or national (#, @, or $) and the remaining characters can be either alphanumeric or national characters. Contact your z/OS administrator for more information about the subsystem names available in your z/OS environment.

STOPF Command

STOPF=*file-number*

Use the `STOPF` command to stop all users who are using the specified file. Any open transactions of the stopped users will be backed out. A stopped user who returns (by sending a command) will receive response code 9 (ADARSP009).

This command does not stop EXF or UTI users.

STOPI Command

STOPI= *time*

Use the `STOPI` command to stop all users who have not executed a command during the specified time interval (in seconds). Any open transactions of the stopped users will be backed out. A stopped user who returns (by sending a command) will receive response code 9 (ADARSP009).

This command does not stop EXF or UTI users.

STOPSU Command

STOPSU= *secuid*

Use the STOPSU command to stop and delete all users with the specified security user ID (*`secuid`*). Any open transactions of the stopped users will be backed out. No response code is issued; the next time the stopped user issues a command, a new user queue element (UQE) is created.

Caution: If Adabas is running with ADARUN OPENRQ=NO (specifying that users are not required to issue an OP as the first command of the session), run the STOPSU command only if you are certain that the users to be deleted are no longer active. If a user with an open transaction is deleted, but then returns (by sending a command), no indication is given about the transaction backout. If the user continues the transaction, logical inconsistencies in the database could occur. If this is a concern, use the STOPSUR command instead.

Note: The STOPSU=*secuid* command is not allowed for online reorder or invert processes. Use the ONLSTOP=X'*identifier*' command instead.

STOPSUR Command

STOPSUR= *secuid*

Use the STOPSUR command to stop a user with a specified security user ID (*secuid*). Any open transactions of the stopped user are backed out. The stopped user is only deleted after he or she has issued a subsequent command and response code 22 (ADARSP22), subcode 54 has been issued in response to that command. This response code-subcode combination is used to notify users that their Adabas activity has been halted and their user session resources have been freed. Only after the response code-subcode combination has been issued is the user queue element (UQE) of the stopped user deleted.

Note: The STOPSUR=*secuid* command is not allowed for online reorder or invert processes. Use the ONLSTOP=X'*identifier*' command instead.

STOPU Command

STOPU= { X'*userid*' | *job-name* }

Use the STOPU command to stop and delete the user with the Adabas-assigned user ID (in the form shown in the display commands), or stop and delete all users with the specified job name (*job-name*). Any open transactions of the stopped users will be backed out. No response code is issued; the next time a stopped user issues a command, a new user queue element (UQE) is created.

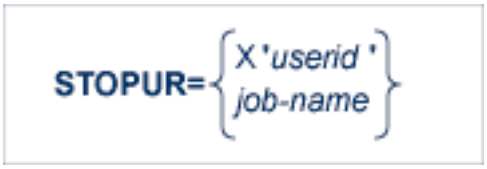
Caution: If Adabas is running with ADARUN OPENRQ=NO (specifying that users are not required to issue an OP as the first command of the session), run the STOPU command only if you are certain that the users to be deleted are no longer active. If a user with an open transaction is deleted, but then returns (by sending a command), no indication is given about the transaction backout. If the user continues the transaction, logical inconsistencies in the database could occur. If this is a concern, use the STOPUR command instead.

Note: The STOPU=X'*userid*' command is not allowed for online reorder or invert processes. Use the ONLSTOP=X'*identifier*' command instead.

The user ID must be specified in hexadecimal format; for example:

```
STOPU=X'1CF2'
```

STOPUR Command



Use the STOPUR command to stop a user with the Adabas-assigned user ID (in the form shown in the display command) or to stop a user with the specified job name. Any open transactions of the stopped user are backed out. The stopped user is only deleted after he or she has issued a subsequent command and response code 22 (ADARSP22), subcode 54 has been issued in response to that command. This response code-subcode combination is used to notify users that their Adabas activity has been halted and their user session resources have been freed. Only after the response code-subcode combination has been issued is the user queue element (UQE) of the stopped user deleted.

Note: The STOPUR=X'*userid*' command is not allowed for online reorder or invert processes. Use the ONLSTOP=X'*identifier*' command instead.

The user ID must be specified in hexadecimal format; for example:

```
STOPUR=X'1CF2'
```

SYNCC Command

Use the `SYNCC` command to force synchronization for all ET users. The nucleus waits for all ET users to reach ET status.

TARGET Command

TARGET = { * | *nucid* | *sysname* }

Use this command against an Adabas Cluster Services nucleus to specify the system on which the Entire Net-Work DBID target for the cluster database is to be established. Specify a system name or the NUCID of a cluster nucleus on that system. The DBID target can be established on any system where a nucleus of the cluster is active. It is used to route commands from users on remote systems with no active PLXCB. Such systems may be part of the z/OS sysplex or other Entire Net-Work nodes outside the sysplex.

The command instructs Entire Net-Work to route requests to an Entire Net-Work node on an implicitly or explicitly specified system image. All future commands will be routed to the new destination. If there are existing users who have been routed using the DBID target and the target is relocated to another Entire Net-Work node, they will be reassigned to any cluster nuclei on the new node. These users lose their existing sessions with their previously assigned nuclei. As a consequence, they may receive response code 9 (ADARSP009) on the first command after the target is moved. Furthermore, they may receive response code 21 (ADARSP021) if they attempt to reuse previously defined command IDs (logical read sequences or search results).

Several possible operands can be used to specify the desired DBID target assignment, as described in the following table.

Operand	Description
*	Specify an asterisk (*) to assign the DBID target to the Entire Net-Work node associated with the nucleus receiving the command.
nucid	Specify the nucleus ID (as defined in the ADARUN NUCID parameter) of an active cluster nucleus to assign the DBID target to the Entire Net-Work node associated with that nucleus.
sysname	Specify a system name (LPAR name) to assign the DBID target to the Entire Net-Work node associated with an active cluster nucleus on that system.

Note: The DBID target identifies an Entire Net-Work node, not a specific nucleus on a system. The DBID target is jointly serviced by all nuclei on that Entire Net-Work node.

TCPIP Command

TCPIP = { OPEN=*url* | CLOSE | + *net-work_cmd*}

Use the `TCPIP` command to open or close a direct TCP/IP link to the Adabas nucleus or to close all TCP/IP links when no URL is specified. You can also use the TCP/IP command to perform any valid Entire Net-Work operator command (*`net-work_cmd`*) or direct it to a TCP/IP link to the Adabas nucleus. The Entire Net-Work operator commands you can use are described in the section entitled *Entire Net-Work Operator Commands* under the section *Statement and Command Reference* in the *Entire Net-Work Reference Guide*. These Entire Net-Work commands must be preceded by a plus character ("+").

Note: The OPEN command can be replaced by the START command and the CLOSE command can be replaced by ADAEND, END, HALT, STOP or TERMINATE, as used by Entire Net-Work.

This command is only possible when the ADARUN parameter `TCPIP` is set to "YES" and all conditions for that setting have been met. This command can be used to close the URL set in the ADARUN `TCPURL` parameter, or to open or close additional TCP/IP links.

You must identify the universal resource locator (URL) for the TCP/IP link you want to open or close. The URL is a 20-byte address that conforms to the RFC specification for URLs. For information on how these URLs should be specified, read about the TCPURL ADARUN parameter, elsewhere in your Adabas documentation.

Examples

```
TCPIP=OPEN=OES::12216 ↩
```

```
TCPIP=CLOSE
```

To turn logging on:

```
TCPIP=+LOG=YES
```

TNAA Command

TNAA= *n*

Use the `TNAA` command to set the non-activity time limit for access-only users. This value must be greater than zero and replaces the value set by the `TNAA` ADARUN parameter.

In nucleus cluster environments, the `TNAA` command is global by definition and affects all nuclei in the cluster.

TNAE Command

TNAE= *n*

Use the `TNAE` command to set the non-activity time limit for ET logic users. This value must be greater than zero and replaces the value set by the `TNAE` ADARUN parameter.

In nucleus cluster environments, the `TNAE` command is global by definition and affects all nuclei in the cluster.

TNAX Command

TNAX= *n*

Use the `TNAX` command to set the non-activity time limit for exclusive control users. This value must be greater than zero and replaces the value set by the `TNAX` ADARUN parameter.

In nucleus cluster environments, the `TNAX` command is global by definition and affects all nuclei in the cluster.

TT Command

TT=*n*

Use the `TT` command to set the transaction time limit for ET logic users. This value must be greater than zero and replaces the value set by the `TT` ADARUN parameter.

In nucleus cluster environments, the `TT` command is global by definition and affects all nuclei in the cluster.

UNLOCKF Command

UNLOCKF=*n*

Use the `UNLOCKF` command to unlock the specified file. File usage is restored to its pre-locked status.

UNLOCKU Command

UNLOCKU=*n*

Use the `UNLOCKU` command to unlock the specified file that was previously locked for all non-utility users. File usage is restored to its pre-locked status.

UNLOCKX Command

UNLOCKX=*n*

Use the `UNLOCKX` command to unlock the specified file that was previously locked for non-exclusive control users. File usage is restored to its pre-locked status.

UTIONLY Command

UTIONLY={YES | NO}

Use the `UTIONLY` command to switch the ADARUN `UTIONLY` status parameter on or off. The default is NO.

ZIIP Command

ZIIP ={YES | NO}

Use the `ZIIP` operator command to turn on or off the use of System z Integrated Information Processors (zIIP) in the Adabas nucleus. Issuing `ZIIP=YES` will tell the Adabas nucleus to run in SRB mode when possible and enable the use of zIIPs.

Issuing `ZIIP=NO` will tell the Adabas nucleus to stay in TCB mode and disable the use of zIIPs.

The ZIIP operator command may only be used in an Adabas nucleus started with `ADARUN ZIIP=YES`.

Adabas Utility Operator Command DSTAT

DSTAT

Use the `DSTAT` command to display the current ADACHK, ADALOD, ADAORD, ADARES, ADASAV, or ADAULD operating status. The following are examples of the output that results when `DSTAT` is entered during specific Adabas utility operations:

Note: ADAORD, ADARES, and ADAULD do not accept operator commands if they run in MODE=SINGLE.

Example 1: ADACHK

Messages ADAU93 appears in addition to the following statistics:

```
19:53:04 ADAU08 44824 Operator Type-In: DSTAT
 19:53:04 ADAU93 ADACHK File=1, Processing in progress
 19:53:04
 19:53:04 *****ADACHK Nucleus Statistics - Error Verifications*****
 19:53:04 Number of transient errors = 0
 19:53:04 Number of nucleus verification calls = 0
 19:53:05 Number of nucleus retries = 0
 19:53:05 Number of AC blocks returned from nucleus = 0
 19:53:05 Number of Index blocks returned from nucleus = 0
 19:53:05 Number of Data blocks returned from nucleus = 0
 19:53:05 Number of nucleus status changes = 0
 19:53:05 Number of Buffer flushes = 0
```

For a description of these statistics, read *ADACHK Statistics*, in the *Adabas Utilities Manual*.

Example 2: ADALOD

```
ADAU00 dbid OPERATOR TYPE-IN : DSTAT ADAU22 dbid LOADING DATA STORAGE. REC-NO=3599
ADAU00 dbid OPERATOR TYPE-IN : DSTAT ADAU21 dbid SORTING/LOADING DESCRIPTOR AA
```

Example 3: ADAORD

```
ADAU08 dbid OPERATOR TYPE-IN : DSTAT ADAU26 dbid UNLOADING INDEX. FILE=1
ADAU08 dbid OPERATOR TYPE-IN : DSTAT ADAU25 dbid UNLOADING DATASTORAGE. FILE=2
ADAU08 dbid OPERATOR TYPE-IN : DSTAT ADAU29 dbid LOADING DATASTORAGE. FILE=2
```

Example 4: ADARES

```
ADAU08 dbid OPERATOR TYPE-IN : DSTAT REGENERATE INPUT VOLUME = BMC002, PLOG-NUM = 12
FROMBLK =        1, FROMTIME = 1996-04-10 11:27:56
TOBLK =          1, TOTIME   = 1996-04-10 11:27:56
```

Example 5: ADASAV

```
ADAU08 dbid OPERATOR TYPE-IN : DSTAT ADAU92 dbid       STILL INITIALIZING
ADAU08 dbid OPERATOR TYPE-IN : DSTAT ADAU10 dbid 435 BLOCKS OUT OF 465 SAVED
ADAU08 dbid OPERATOR TYPE-IN : DSTAT ADAU11 dbid 342 BLOCKS OUT OF 451 RESTORED
```

Example 6: ADAULD

```
ADAU08 dbid OPERATOR TYPE-IN : DSTAT ADAU67 dbid UNLOADING FILE=17, RECNO=2875
```

SMGT Operator Command

The Adabas error handling and message buffering facility uses a single operator command, `SMGT`, followed by a comma and one or more operands:

SMGT, *operand* **[,** *operand* **],...**

Valid *operands* are: `ABNORMAL`, `ACTPIN`, `ADDPIN`, `DEACTPIN`, `DELPIN`, `DISPLAY`, `DUMP`, `MSGBUF`, `OFF`, `ON`, `SNAP`, `XACTIVATE`, `XCRITICAL`, `XDEACTIVATE`, `XLOAD`, and `XNOTCRITICAL`. Some of these operands are mutually exclusive. All operands are described in ***Operands for SMGT***, elsewhere in this section.

Operands may require that you enter one or more of the following variables:

Variable Type	Description
exit-code	The code that identifies an exit routine to an SMGT operator command is one of the following: ■ UEX*n*, where *n* is a single-digit user exit number ■ UX*nn*, where *nn* is a double-digit user exit number ■ HX*nn*, where *nn* is a hyperdescriptor exit number (single-digit numbers are preceded by a zero; e.g., HX02) ■ CX00 (the Adabas Caching Facility user exit) ■ SX00 (the PIN routine user exit)
start, *end*	The range of addresses for the SNAP operand where *start* is the hexadecimal address where the SNAP dump begins and *end* is the hexadecimal address where it ends.
module-name	The name of the module. For the XLOAD operand, it is the name of the exit routine module to be loaded; for ADDPIN and DELPIN operands, it is the name of the PIN module to be added or deleted, respectively.
pin-number	When a PIN is added, it is assigned a number that can be found using the DISPLAY=PINS operand. This pin number is used for the ACTPIN and DEACTPIN operands to identify the PIN routine to be activated or deactivated, respectively.

Acceptable abbreviations for operands are provided in mixed case notation (capital letters required, lowercase letters optional). Default values for operands are underlined.

Operands for SMGT

SMGT,{ ABNormalterm | MSGbuf } = { ON | OFF }

The ABNORMALTERM (ABN) operand determines whether the error handling and message buffering facility will handle abnormal termination errors.

The MSGBUF (MSG) operand temporarily deactivates (OFF) or reactivates (ON) message buffering.

The ABN and MSG operands are mutually exclusive; if one is specified on an individual SMGT command, the other cannot be.

SMGT,{ ACTpin | DEACTpin } = *pin-number*

The `ACTPIN` (`ACT`) operand is used to activate an individual PIN routine or to reactivate it after it has been temporarily deactivated.

The `DEACTPIN` (`DEACT`) operand is used to deactivate an individual PIN routine.

The `ACT` and `DEACT` operands are mutually exclusive; if one is specified on an individual `SMGT` command, the other cannot be.

SMGT,{ ADDpin | DELpin } = *module-name*

The `ADDPIN` (`ADD`) operand adds PIN routines to the error handling facility. The PIN routine module indicated is loaded and the PINs that are found in it are added to the facility. When a PIN is added, it is assigned a number which can be found using the DISPLAY=PINS operand.

The `DELPIN` (`DEL`) operand deletes a PIN module and all the PINs it contains from the facility.

The `ADD` and `DEL` operands are mutually exclusive; if one is specified on an individual `SMGT` command, the other cannot be.

SMGT,Display = { ALL | EXITS | MSGBUF | PINS | SUMMARY | LAST }

The `DISPLAY` (`D`) operand writes status and history information about the error handling and message buffering facility to the job log and to DDPRINT:

ALL	(the default) displays all reports
EXITS	displays current user exit, hyperdescriptor exit, and other exit status
MSGBUF	displays message buffering status
PINS	displays PIN routine status
SUMMARY	displays only the summary information from ALL
LAST	displays details of the most recent recovery action

Sample output for the various `DISPLAY` values is provided in *SMGT Display Output Samples*, in *Adabas Operations*.

SMGT,DUMP = { ON | OFF }

The `DUMP` operand determines whether a full system dump is taken for the Adabas nucleus in the event of an error. The default (OFF) means that only a snap dump is taken.

Note: To use this command successfully under z/OS, the data set ADASNAP must be provided in the Adabas startup JCL. For more information, read *Adabas Session Execution*, in *Adabas Operations*.

SMGT,{ ON | OFF }

The `OFF` operand deactivates the error handling and message buffering facility temporarily; the `ON` operand reactivates it.

When SMGT=OFF:

- The only valid `SMGT` operands that can be specified after SMGT=OFF is specified are `ON` (in a new `SMGT` command) and `DISPLAY`. All other SMGT commands are rejected until SMGT=ON is issued.
- All other functions of the error handling facility are disabled. PIN modules cannot be invoked. Any user exits marked NOTCRITICAL are treated as CRITICAL; that is, the nucleus terminates abnormally if an error occurs in the exit.

The `ON` and `OFF` operands are mutually exclusive; if one is specified on an individual `SMGT` command, the other cannot be.

SMGT,SNAP [= (*start* , *end*)]

The `SNAP` operand displays a formatted dump of the nucleus without error diagnostics. If `SNAP` is specified without any additional parameters, the whole nucleus is displayed; if hexadecimal addresses are specified for start and end, the `SNAP` dump is displayed only for that range of addresses.

The `SNAP` operand can degrade system performance as long as it is active.

Note: To use this command successfully under z/OS, the data set ADASNAP must be provided in the Adabas startup JCL. For more information, read *Adabas Session Execution*, in *Adabas Operations*.

SMGT,{ XActivate | XDeactivate } = *exit-code*

The `XACTIVATE` (`XA`) operand activates a loaded exit module; the `XDEACTIVATE` (`XD`) operand deactivates an active exit module.

The `XA` and `XD` operands are mutually exclusive; if one is specified on an individual `SMGT` command, the other cannot be.

SMGT,{ XNotcritical | XCritical } = *exit-code*

The `XNOTCRITICAL` (`XN`) operand changes the status of an exit from the default critical to noncritical for the functioning of the Adabas nucleus. Any abnormal termination or program check in a noncritical exit results in the exit being automatically deactivated; however, the Adabas nucleus continues to run. The disabled user exit is not recalled until it is reset. Once the exit error has been corrected, the exit can be reactivated using the `XACTIVATE` operand.

The `XCRITICAL` (`XC`) operand changes the status of an exit from noncritical back to the default critical for the functioning of the Adabas nucleus. Any abnormal termination or program check in a critical exit causes the Adabas nucleus to terminate.

The `XN` and `XC` operands are mutually exclusive; if one is specified on an individual `SMGT` command, the other cannot be.

SMGT,XLoad = { *exit-code* | (*exit-code*, *module-name*) }

The `XLOAD` (`XL`) operand loads a new version of an exit module.

The module-name parameter is required only for new exits. Because the default is the previous module name, the parameter is optional for exits that have previously been used. The parameter is also optional for any exit defined with a number of zero; for example, the ADACSH exit which is always named ADACSHUX.

Caution: ADACSHUX is a sample user program and is not supported under any maintenance contract agreement.

SMGT Display Output Samples

This section provides sample output for the `DISPLAY` operand of the `SMGT` command.

```
SMGT,Display = { ALL | EXITS | MSGBUF | PINS | SUMMARY | LAST }
```

The `DISPLAY` (`D`) operand writes status and history information about the Adabas error handling and message buffering facility to the job log and to DDPRINT:

ALL	(the default) displays all reports
EXITS	displays current user exit, hyperdescriptor exit, and other exit status
MSGBUF	displays message buffering status
PINS	displays PIN routine status
SUMMARY	displays only the summary information from ALL
LAST	displays details of the most recent recovery action

This section provides sample output for the following options of the `DISPLAY` operand:

- DISPLAY=ALL
- DISPLAY=EXITS
- DISPLAY=MSGBUF
- DISPLAY=PINS
- DISPLAY=SUMMARY
- DISPLAY=LAST

DISPLAY=ALL

The ALL parameter displays all of the information shown for the other parameters in the following paragraphs.

DISPLAY=EXITS

```
ADANA1 00127 SMGT DISPLAY ACTIVE
ADANA2 00127 SMGT ACTIVE
ADANAX 00127 EXIT: UX04 MODNAME: EXIT1    STATUS: ACTIVE
ADANAX 00127 EXIT: HX22 MODNAME: NULLEXIT STATUS: ACTIVE
ADANO2 00127 SMGT COMMAND PROCESSED
```

DISPLAY=MSGBUF

```
ADANA1 00127 SMGT DISPLAY ACTIVE
ADANA2 00127 SMGT ACTIVE
ADANAE 00127 MESSAGE BUFFERING IS ACTIVE
ADANO2 00127 SMGT COMMAND PROCESSED
```

DISPLAY=PINS

```
ADANA1 00127 SMGT DISPLAY ACTIVE
ADANA2 00127 SMGT ACTIVE
ADANAG 00127 PIN 0001 USES: 0000 CONDITION: 000C4000
             THIS PIN VALID FOR ALL LOCATIONS
ADANAG 00127 PIN 0002 USES: 0000 CONDITION: 000C1000
             LOCATION: 00081C6C 0008259B (EBL2    )
ADANAA 00127 002 CONDITION PIN ROUTINES RECOVERED 000 ERRORS
ADANAB 00127 000 LOCATION PIN ROUTINES RECOVERED 000 ERRORS
ADANAG 00127 PIN 0003 USES: 0000 CONDITION: RSP: 017
             THIS PIN VALID FOR ALL LOCATIONS
ADANAC 00127 001 RESPONSE PIN ROUTINES RECOVERED 000 ERRORS
ADANAD 00127 003 TOTAL PIN ROUTINES RECOVERED 000 ERRORS
ADANA8 00127 000 EXECUTIONS OF ABNORMAL TERMINATION HANDLER
ADANA8 00127 000 EXECUTIONS OF PROGRAM CHECK HANDLER
ADANA8 00127 000 EXECUTIONS OF RESPONSE CODE HANDLER
ADANO2 00127 SMGT COMMAND PROCESSED
```

DISPLAY=SUMMARY

```
ADANA1 00127 SMGT DISPLAY ACTIVE
ADANA2 00127 SMGT ACTIVE
ADANA3 00127 ABNORMAL TERMINATION HANDLER ACTIVE
ADANA4 00127 PROGRAM CHECK HANDLER ACTIVE
ADANAA 00127 002 CONDITION PIN ROUTINES RECOVERED 000 ERRORS
ADANAB 00127 000 LOCATION PIN ROUTINES RECOVERED 000 ERRORS
ADANAC 00127 001 RESPONSE PIN ROUTINES RECOVERED 000 ERRORS
ADANAD 00127 003 TOTAL PIN ROUTINES RECOVERED 000 ERRORS
ADANA8 00127 000 EXECUTIONS OF ABNORMAL TERMINATION HANDLER
```

```
ADANA8 00127 000 EXECUTIONS OF PROGRAM CHECK HANDLER
ADANA8 00127 000 EXECUTIONS OF RESPONSE CODE HANDLER
ADANO2 00127 SMGT COMMAND PROCESSED
```

DISPLAY=LAST

No error:

```
ADANA1 00127 SMGT DISPLAY ACTIVE
ADANA2 00127 SMGT ACTIVE
ADANA5 00127 NO ERROR CONDITIONS HANDLED
ADANO2 00127 SMGT COMMAND PROCESSED
```

Error Encountered:

```
ADANA1 00127 SMGT DISPLAY ACTIVE
ADANA2 00127 SMGT ACTIVE
ADANA6 00127 LAST ERROR OCCURRED AT: 1998-07-07 14:36:18
ADANA7 00127 CONDITION: RSP: 017 LOCATION: *N/A*
ADANO2 00127 SMGT COMMAND PROCESSED
```

9 Adabas Session Execution

- Program Synchronization Using z/OS Session System Services 328
- Single-User Mode 335
- User Profile 335

An Adabas session involves the execution of the Adabas nucleus which controls access/update to a single database. This chapter describes the job control statements needed when executing an Adabas session under each supported operating system. For examples of the Adabas utility jobs, see the Adabas Utilities documentation.

Adabas version uses operating system services to synchronize the start and end of nucleus and utility executions. Only one program can modify the data integrity block (DIB) at a time.

Program Synchronization Using z/OS Session System Services

The services consist of systems-wide ENQ/DEQ macros (SCOPE=SYSTEMS) with major name (QNAME) 'ADABAS'

This feature reliably and efficiently guarantees proper synchronization of DIB updates within a single operating-system image.

If your database resides on disks that are shared among multiple images of the operating system and you run nucleus or utility jobs against the same database on more than one of the system images, you need to ensure that

- the system images are installed in such a way that synchronization is effective on all systems where nucleus and utility jobs execute; or
- nucleus and utility jobs do not execute concurrently on different system images.

Consult your system programmer for the needed information.

Caution: If different nucleus or utility jobs updating the same file are allowed to start or terminate on different system images at the same time without proper synchronization, a DIB update may be lost. If this happens, a lock in the DIB may be violated, thereby opening the file to the possibility of destruction due to concurrent unsynchronized updates by utilities.

The following data sets are required when executing an Adabas session under z/OS.

Data Set	DD Name	Storage Medium	Additional Information
ADARUN parameters	DDDDCARD	card image	**note 1**
ADARUN / Adabas messages	DDPRINT	printer	**note 2**
Associator	DDASSOR*n* or DDASSO*nn*	disk	**note 3**
ADATCP parameters	TCPIN	card image	This optional dataset is only used with ADATCP, to provide input parameters. See *JCL Required for*

Data Set	DD Name	Storage Medium	Additional Information
			UES and TCP/IP Support (z/OS) for further details.
Data Storage	DDDATAR*n* or DDDATA*nn*	disk	**note 3**
Work	DDWORKR1 DDWORKR4	disk	**note 4**
Recovery Aid log	DDRLOGR1	disk	**note 5**
Protection log multiple log 1 multiple log 2	DDSIBA DDPLOGR1 DDPLOGR2	tape/disk disk disk	**note 6** **note 7** **note 7**
Command log multiple log 1 multiple log 2	DDLOG DDCLOGR1 DDCLOGR2	tape/disk disk disk	**note 8** **note 9** **note 9**
Abnormal termination	MPMDUMP	printer	**note 10**
Abnormal termination (if large buffer pools are used)	SVCDUMP	printer	**note 11**
SMGT dump and snap dump	ADASNAP	printer	**note 12**
SMGT print file	DDTRACE1	printer	**note 13**
Time zone files	TZINFO	disk	**note 15**

If you did not convert the license file to a license load module and added it to the Adabas load library (preferable), you need to add a DD statement for each Adabas and add-on product relating the sequential license file.

For a list of license file names, load modules and DD names, refer to *Adabas and Add-on Licenses*.

Example of Adabas Session Job Control (z/OS)

This job includes multiple protection logging, multiple command logging, and Recovery Aid logging:

```
//NUC099    EXEC PGM=ADARUN
//STEPLIB   DD DISP=SHR,DSN=ADABAS.ADAvrs.LOAD
//DDASSOR1  DD   DISP=SHR,DSN=EXAMPLE.ADAyyyyy.ASSOR1
//DDDATAR1  DD   DISP=SHR,DSN=EXAMPLE.ADAyyyyy.DATAR1
//DDWORKR1  DD   DISP=OLD,DSN=EXAMPLE.ADAyyyyy.WORKR1
//DDPLOGR1  DD   DISP=SHR,DSN=EXAMPLE.ADAyyyyy.PLOGR1
//DDPLOGR2  DD   DISP=SHR,DSN=EXAMPLE.ADAyyyyy.PLOGR2
//DDCLOGR1  DD   DISP=SHR,DSN=EXAMPLE.ADAyyyyy.CLOGR1
//DDCLOGR2  DD   DISP=SHR,DSN=EXAMPLE.ADAyyyyy.CLOGR2
//DDRLOGR1  DD   DISP=SHR,DSN=EXAMPLE.ADAyyyyy.RLOGR1
//TZINFO    DD   DISP=SHR,DSN=ADABAS.Vvrs.TZOO
//DDPRINT   DD  SYSOUT=X
```

```
//DDTRACE1   DD  SYSOUT=X
//SYSUDUMP  DD  SYSOUT=X
//MPMDUMP   DD  SYSOUT=X
//ADASNAP   DD  SYSOUT=X
//DDDDCARD    DD  *
ADARUN PROG=ADANUC,DB=yyyyy
ADARUN LBP=600000
ADARUN LWP=320000
ADARUN LS=80000
ADARUN LP=400
ADARUN NAB=24
ADARUN NC=1000
ADARUN NH=2000
ADARUN NU=100
ADARUN TNAE=180,TNAA=180,TNAX=600,TT=90
ADARUN NPLOG=2,PLOGSIZE=1800,PLOGDEV=dddd
ADARUN NCLOG=2,CLOGSIZE=1800,CLOGDEV=dddd
//
```

where:

dddd	is a valid device type.
vrs	is the *version* of the product.
yyyyy	is the physical database ID.

JCL Required for UES Support (z/OS)

If you have UES-enabled your database, you must also include the ICS load library in the STEPLIB:

```
//STEPLIB DD ....
//        DD DISP=SHR,DSN=SAG.ICSvrs.LOAD
```

Note: The library contains conversion objects that are loaded on demand for text conversions.

JCL Required for UES and TCP/IP Support (z/OS)

If you are connecting your UES-enabled database directly through a TCP/IP link, you must also:

- Include the ADATCP library in the STEPLIB:

```
//STEPLIB DD ....
//        DD DISP=SHR,DSN=WCPvrs.LOAD
//        DD DISP=SHR,DSN=WTCvrs.LOAD
```

Note: These libraries are distributed with Entire Net-Work.

- If necessary, include the TCPIN DD statement, for example:

```
//TCPIN DD *
ADI=Y
ADIHOST=AHOST
ADIPORT=4952
/* ↩
```

Using the z/OS SVC Dump Facility

The dump produced by MPMDUMP may be too slow for users with very large buffer pools. You may instead elect to use the z/OS SVC dump facility to speed up nucleus dump processing. An SVC dump is triggered by the presence of an //SVCDUMP DD statement in the nucleus startup JCL.

If `//SVCDUMP DD DUMMY` is specified, and the job is running with APF authorization, a z/OS SVC dump is produced on the system dump data set, normally SYS1.DUMP*xx*. If `//SVCDUMP DD DUMMY` is specified and the job is *not* running with APF authorization, message ADAM77 is issued and dump processing continues as if the SVCDUMP DD statement had *not* been specified. For information on this message, refer to the *Adabas Messages and Codes Manual*.

If `//SVCDUMP DD DSN=dsn` is specified (with an appropriate data set name), a z/OS SVC dump is produced on the specified data set. Note that the SVCDUMP data set needs to be allocated with DCB attributes `RECFM=FB,LRECL=4160,BLKSIZE=4160`. Note also that, for APF-authorized jobs, secondary extents are ignored.

For APF-authorized jobs, the SVC dump title consists of "ADABAS System Dump" plus additional job name, DBID, and timestamp information in the form: `Job jjjjjjjj DBID nnnnn yyyy/mm/dd hh:mm:ss.hhttmm`. For non-authorized jobs, the title consists of "ADABAS Sys Tx Dump" plus the additional job name, DBID, and timestamp information. Note that the timestamp reflects the time at which the SVC dump request was passed to z/OS -- not the time at which any preceding abend occurred.

The shortest dump processing time occurs when you specify `//SVCDUMP DD DUMMY`. This is because the nucleus only needs to wait for the dump to be captured, not written out to a dump data set. A specification of `//SVCDUMP DD DSN=dsn` will give you a shorter processing time for an APF-authorized job than for a non-APF-authorized job. In both cases, the time taken for dump processing may actually be longer than with MPMDUMP.

When an SVCDUMP DD statement is included in your JCL, any MPMDUMP DD statement is ignored, unless Adabas detects that it is unable to proceed with SVC dump processing.

If an error is encountered while writing the SVC dump, message ADAM78 appears. If dump writing completes successfully, message ADAM79 appears. For more information on these messages, refer to your Adabas messages and codes documentation.

No error message is produced when a dump can only be partially written. You should therefore ensure that sufficient space is available on the dump data set to accommodate the dump.

When an SVCDUMP DD statement is included in the JCL, but the SVC dump is unable to complete successfully, dump processing reverts to the standard dump options as specified in the JCL via the MPMDUMP, SYSUDUMP, SYSABEND or SYSMDUMP DD statements.

Note: SVC dump processing might be suppressed due to installation SLIP or DAE options. If dump processing is still required in this case, the relevant MPMDUMP, SYSUDUMP, SYSABEND or SYSMDUMP DD statement should be specified in the JCL in addition to the SVCDUMP DD statement.

When the SVCDUMP DD statement is omitted from the JCL, existing dump options, specified via the MPMDUMP, SYSUDUMP, SYSABEND or SYSMDUMP DD statements, continue to operate as normal.

z/OS JCL Notes

The following notes apply to Adabas startup jobs in various platforms (as described by the notes).

1. This data set is used to provide the Adabas session parameters.
2. This data set is used to print messages produced by the control module ADARUN or the Adabas nucleus.
3. The Adabas Associator and Data Storage data sets. These data sets are mandatory.

   n and *nn* represent the number of the Associator and Data Storage data set, respectively.

   If more than one data set exists for Associator or Data Storage, a separate statement is required for each. For example, if the Associator consists of two data sets, DD statements for DDASSOR1 and DDASSOR2 are required.

   If *less* than 10 data sets exist for each, you must use the DDASSOR* and DDDATAR* DD names in the JCL. For example, if the Associator consists of two data sets and Data Storage consists of three data sets, the following names would be used in the JCL: DDASSOR1, DDASSOR2, DDDATAR1, DDDATAR2, and DDDATAR3. If 10 or more data sets exist, the first nine must use the ASSOR* and DATAR* DD names in the JCL and the remainder must use the ASSO* and DATA* DD names in the JCL (dropping the "R" in the DD names). For example, the tenth Associator data set is identified in the JCL using the name DDASSO10, while the third Associator data set in the same JCL would be identified using the name DDASSOR3.

A maximum of 99 physical extents is now set for Associator and Data Storage data sets. However, your actual real maximum could be less because the extent descriptions of all Associator, Data Storage, and Data Storage Space Table (DSST) extents must fit into the general control blocks (GCBs). For example, on a standard 3390 device type, there could be more than 75 Associator, Data Storage, and DSST extents each (or there could be more of one extent type if there are less for another).

So, the range of Associator DD names that can be used in JCL is DDASSOR1 to DDASSOR9 and DDASSO10 to DDASSO99. And the range of Data Storage DD names that can be used in JCL is DDDATAR1 to DDDATAR9 and DDDATA10 to DDDATA99.

4. The Adabas Work data sets. The WORKR1 data set is mandatory. If you have Adabas Transaction Manager installed, an additional work data set, WORKR4 is also mandatory.

   we recommend running the nucleus with DISP=OLD for the WORKR1 data set as a way of preventing two nuclei from writing to the same WORK data set and corrupting the database. This could otherwise happen if the ADARUN parameters FORCE and IGNDIB are improperly used.

   Work part 4 of WORKR1 is no longer supported if you have Adabas Transaction Manager installed. Instead, you should use the WORKR4 data set. WORKR4 is used for the same purpose as Work part 4, but it can be used in parallel by all members in a cluster. It is used to store the PET (preliminary end-of transaction) overflow transactions (those that cause a work overflow) of a database or of all members in a multiplex/SMP cluster.

   The WORKR4 data set is a container data set that should be allocated and formatted in the normal way (use ADAFRM WORKFRM), using a block size greater than or equal to WORKR1. WORKR4 can have the same or a different device type than WORKR1. It should be at least as large as the cluster's LP parameter of the database or cluster. The smaller WORKR1 Work part 1 is, the larger WORKR4 should be. This is because the nucleus must prevent a work overflow due to incomplete DTP transactions, but the nucleus must keep all PET transactions; they cannot be backed out.

5. If the Adabas Recovery Aid is being used, this logging data set is required.

6. The data protection log data set. This data set is required if the database will be updated during the session and logging of protection information is desired. This data set is not applicable if multiple protection logging is used.

   The data protection log may be assigned to tape or disk. A new data set must be used for each Adabas session (DISP=MOD may not be used). See ***Adabas Restart and Recovery*** for additional information.

7. Multiple (two to eight) data protection log data sets. These data sets are required only if multiple data protection logging is to be in effect for the session.

   Multiple data protection logging is activated by the ADARUN NPLOG and PLOGSIZE parameters. The device type of the multiple protection logs is specified with the ADARUN PLOGDEV parameter.

Whenever one of multiple protection log data sets is full, Adabas switches automatically to another data set and notifies the operator through a console message that the log which is full should be copied using the PLCOPY function of the ADARES utility. This copy procedure may also be implemented using the user exit 12 facility as described in the User Exits documentation.

If no command logging is to be performed, this data set may be omitted.

8. The command log data set. This data set is required if command logging is to be performed during the session. Command logging is activated by the ADARUN LOGGING parameter.
9. Multiple (two to eight) command log data sets. These data sets are required only if multiple command logging is to be in effect for the session.

   Multiple command logging is activated by the ADARUN NCLOG and CLOGSIZE parameters. The device type of the multiple command log data sets is specified with the ADARUN CLOGDEV parameter.

   Whenever one of multiple command log data sets is full, Adabas switches automatically to another data set and notifies the operator through a console message that the log which is full should be copied using the CLCOPY function of the ADARES utility. This copy procedure may also be implemented using the user exit 12 facility as described in the Adabas User Exits documentation.
10. This data set is used to take an Adabas dump including SVC, ID-TABLE and allocated CSA in the event that an abnormal termination occurs.

    The line count in the JCL must be set appropriately; otherwise, the dump cannot be printed in its entirety.
11. The z/OS SVC dump facility can be used to speed up nucleus dump processing if an MPMDUMP is too slow for users with very large buffer pools. We recommend that you specify both an MPMDUMP and an SVCDUMP DD statement in your JCL to ensure that one of the dumps is produced when needed. If an SVCDUMP DD statement is included, an SVC dump is created if possible and the MPMDUMP DD statement is ignored. Should problems arise during processing of the SVC dump, an MPMDUMP will be taken, but only if the MPMDUMP DD statement is also specified in the JCL. For more information, read ***Using the z/OS SVC Dump Facility***, elsewhere in this guide.
12. This data set is used under z/OS to take an Adabas dump (SMGT,DUMP) or snap dump (SMGT,SNAP) when using the error handling and message buffering facility.
13. This data set is used when the PIN output is to be directed to DDTRACE1 rather than DDPRINT as specified by the user in ADASMXIT when using the error handling facility. If DDTRACE1 has not been specified in the JCL and PRINTDD is set to "NO" in ADASMXIT, output will be lost. The PIN output will not be written to DDPRINT unless "YES" is specified.
14. The TZINFO data set is required when defining fields with the TZ option and using the user session open parameter TZ. This data set is a library or partioned data set containing information on local time offsets and daylight savings time offsets and its transition times.

Single-User Mode

Although the normal mode of operation is multiuser mode, it is also possible to execute Adabas together with a user program or Adabas utility in the same region.

For single-user mode, you must include the Adabas nucleus job control that you use along with the job control for the utility or user program.

The Adabas prefetch option cannot be used in single-user mode; however, single-user mode *must* be used when running a read-only nucleus and an update nucleus simultaneously.

User Profile

Some information within an Adabas database is user-related and must be retained from session to session. One such kind of information is ET data records; another is the priority value assigned to a user.

A set of user-related information can be stored in a profile table. The values stored in this table are read at OPEN time and assigned to the user. The direct call user must OPEN the Adabas session with the proper call; that is, as an ID user with an ETID in the Additions 1 field of the Adabas control block. For Natural users, the profile table is identified by the Natural ETID.

The associated fields are user-related timeout and threshold values, and the OWNERID for multiclient fields. One record per user is stored. The profile table is maintained using Adabas Online System.

Profile Table Values

The user-related values shown below are currently stored in the profile table.

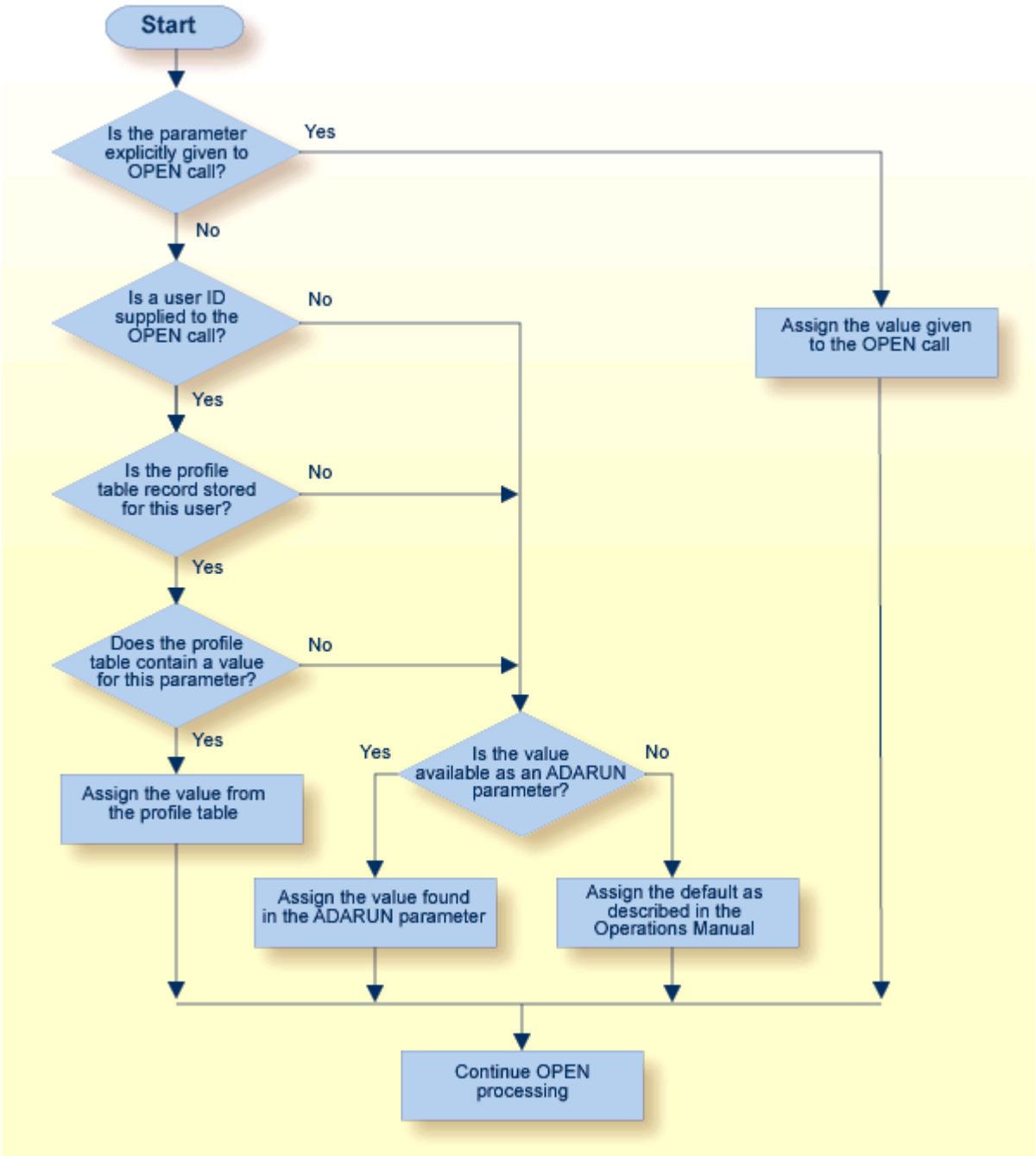
Value	Description
PRIORITY	User's priority (0-255)
TNAA*	Access user non-activity time
TNAE*	ET user non-activity time
TNAX*	EXU/EXF user non-activity time
TT*	Transaction time threshold
TLSCMD*	S*x* command threshold
NSISN*	Maximum number of ISNs per TBI element
NSISNHQ*	Maximum number of records held by user
NQCID*	Maximum number of active command IDs per user

Value	Description
OWNERID	Owner ID for multiclient file access

* The decision sequence for determining the values for a user at the time of an open call is shown in *Managing the User Profile*.

Managing the User Profile

Adabas Online System (AOS) must be used to maintain the profile table. See the Adabas Online System documentation for detailed information about managing the profile table.


Start
Is the parameter explicitly given to OPEN call?
Yes
No
Assign the value given to the OPEN call
Is a user ID supplied to the OPEN call?
No
Yes
Is the profile table record stored for this user?
No
Yes
Does the profile table contain a value for this parameter?
No
Yes
Assign the value from the profile table
Yes
Is the value available as an ADARUN parameter?
No
Assign the value found in the ADARUN parameter
Assign the default as described in the Operations Manual
Continue OPEN processing

10 Adabas Utility Execution

- Condition and Termination Codes 340
- Utility Usage of Files and Databases 341
- Disk Accesses by Utility and Function 344
- Possible Resource Conflicts 346

The Adabas Utilities documentation contains detailed information about the function and execution of each Adabas utility.

You can use Adabas Online System to execute various utility functions online.

Condition and Termination Codes

Refer to the Adabas messages and codes documentation for detailed descriptions of the condition and termination code meanings for each utility.

The following condition codes can be set by an Adabas utility:

Code	The utility operation ...
0	was successful.
4	was successful but encountered a warning condition. With the ADACHK utility, a return code of 4 indicates that either the number of unreachable index blocks is greater than zero or any CHK*nnn*W (warning) message was received.
8	encountered an error condition but then continued. With the ADACHK utility, a return code of 8 indicates that a database inconsistency was encountered.
16	was successful in its main function but then encountered an error condition.
20	could not be completed. The operation encountered an error condition but did not terminate abnormally because NOUSERABEND was specified.

For errors that occur during the execution of a utility, the following termination codes are returned:

Code	The utility terminated . . .
34	with a dump.
35	without a dump.
36	with a dump. Termination was caused by an I/O error while writing to DDDRUCK or DDPRINT. Check the JCL.

Utility Usage of Files and Databases

The table in this section shows the type of file/database usage required for each utility, the database status required, whether the Adabas nucleus must be active while an Adabas utility is running, and whether the function creates an entry in the data integrity block (DIB).

Any utility that requires a nucleus can be used in single-user mode, provided that the nucleus JCL statements are specified.

Utility	Function	File/DB Usage	Database Activity Permitted?	Adabas Nucleus Status?	DIB Entry Created
ADACDC	All	n/a	Permitted	Active or Inactive	NO
ADACHK	ACCHECK	ACC	Permitted	Active or Inactive	YES
	CHECK		Permitted	Active or Inactive	YES
	DSCHECK		Permitted	Active or Inactive	YES
	ICHECK		Permitted	Active or Inactive	YES
	VALIDATE		Permitted	Active or Inactive	YES
	All other functions	n/a	Permitted	Active or Inactive	NO
ADACMP	Compress FDT=...	ACC		Active	NO
	Compress data	n/a	Permitted	Active or Inactive	NO
	Decompress data	n/a	Permitted	Active or Inactive	NO
	Decompress				
	INFILE=...	EXU (EXF)		Active	NO
ADACNV	All	(1)	Not Permitted	Inactive	YES
	With TEST parameter	ACC	Permitted	Active or Inactive	NO
ADADBS	ONLADD	n/a	Permitted	Active	NO
	ONLINCREASE	n/a	Permitted	Active	NO
	OPERCOM DDIB	n/a	Permitted	Active or Inactive	NO
	RECOVER	UTI	(5)	Active	NO
	RESETDIB	n/a	Permitted	Active or Inactive	NO
	RESETPPT	n/a	Not Permitted (7)	Inactive (7)	NO
	others	ACC, UTI or EXF	Permitted (4)	Active	NO
ADADEF	All	(1)	Not Permitted	Inactive	NO
ADADRU	All	n/a	Permitted	Active or Inactive	NO
ADAFRM	Formatting other than TEMP/SORT data set	n/a	Not Permitted	Inactive	NO

Utility	Function	File/DB Usage	Database Activity Permitted?	Adabas Nucleus Status?	DIB Entry Created
	Formatting TEMP/SORT data set	n/a	Permitted	Active or Inactive	NO
	Formatting single blocks	n/a	Permitted	Active or Inactive	NO
ADAINV	All	EXU or UTI	Permitted	Active	YES
ADALOD	All	UTI	Permitted	Active or Inactive	YES
ADAMER	ADAM estimation	n/a	Permitted	Active or Inactive	NO
ADAORD	REORDER A/D/DB	(1)	Not Permitted	Active	YES
	REORDER FILE (A)	EXU or UTI(2)	Permitted	Active	YES
	REORDER FILE (D)	EXU or UTI(2)	Permitted	Active	YES
	REORDER FILE	EXU or UTI(2)	Permitted	Active	YES
	RESTRUCTURE DB	(1)	Not Permitted	Active	YES
	RESTRUCTURE FILE	(EXU (2)	Permitted	Active	YES
	STORE	UTI	Permitted	Active	YES
ADAPLP	All	n/a	Permitted	Active or Inactive	NO
ADARAI	DISABLE				
	PREPARE				
	REMOVE CHKDB	n/a	Permitted	Inactive	NO
	LIST				
	RECOVER	n/a	Permitted	Active or Inactive	NO
ADAREP	Database reporting	n/a	Permitted	Active or Inactive	NO
	CPLIST/CPEXLIST	ACC	Permitted	Active	NO
ADARES	Regenerate file				
	Backout file	UTI (3)	Permitted	Active	NO
	Regenerate				
	Backout	(1)	Not Permitted	Active	NO
	Copy	n/a	Permitted	Active or Inactive	NO
	Copy dual/multi log	n/a	Permitted	Active or Inactive	NO
	Repair	UTI	Permitted	Active	NO
ADASAV	Save file	UTS	Permitted	Active or Inactive	YES
	Save database	UTS	Permitted (6)	Active or Inactive	YES
	Restore file	UTI	Permitted	Active or Inactive	YES
	Restore database	(1)	Not Permitted	Inactive	YES
	Restore protection log	(1)	Not Permitted	Inactive	YES

Utility	Function	File/DB Usage	Database Activity Permitted?	Adabas Nucleus Status?	DIB Entry Created
ADASEL	Select protection log data	n/a	Permitted	Active or Inactive	NO
ADAULD	Unload file	EXU or EXF	Permitted	Active	NO
	Unload file from save tape	n/a	Permitted	Active or Inactive	NO
ADAWRK	Work area recovery reports	n/a	Not Permitted	Inactive	NO
ADAZAP	All	(1)	Not Permitted	Inactive	NO
ADAZIN	Lists maintenance and status information about Adabas load modules and status information about the Adabas SVC	n/a	Permitted	Active or Inactive	NO

The codes used in the File/Database Usage column of this table and the footnotes used in the table are now described.

File/Database Usage Codes and Notes

Code	Meaning
n/a	Does not apply.
ACC	Access-only usage: other users may access and update the file/database.
EXF	Exclusive file control: other users may not access or update the file.
EXU	Exclusive file update: other users may not update the file.
UTI	Utility update control: other users may not access or update the file.
UTS	Utility SAVE control: other users may access and update the file.
UPD	Update/ET user status does not apply to utilities. Utilities cannot be timed out.

Footnotes

Code	Meaning
(1)	Exclusive database control: no other database activity is permitted.
(2)	Exclusive database control if the checkpoint or security file is involved; otherwise, read-only access is allowed during the read phase of utility operation.
(3)	Exclusive database control if the CONTINUE parameter is specified.
(4)	When completed, ADADBS ADD/INCREASE end the nucleus session to permit reallocation and formatting.
(5)	ADADBS RECOVER cannot run when other UTI utilities are running.
(6)	When running ADASAV SAVE with an active nucleus (online), no other utilities can be run which update the database.

Code	Meaning
(7)	ADADBS RESETPPT must be run in MODE=MULTI, with the nucleus down. The reason for this is that utilities run in MODE=SINGLE activate the Adabas nucleus, which causes a new PPT to be written. This is not allowed when trying to reset the PPT.

Disk Accesses by Utility and Function

The following table lists the data sets that each utility function accesses on disk:

Utility	Function	Data Set Accessed Directly
ADACDC	all	ASSO
ADACHK	ACCHECK	ASSO, DATA, SORT (if SORTTYPE=INTERNAL)
	ASSOPRINT	ASSO, DATA
	CHECK	ASSO, DATA, SORT (if SORTTYPE=INTERNAL)
	CLOGPRINT	ASSO, DATA, CLOG
	DATAPRINT	ASSO, DATA
	DSCHECK	ASSO, DATA, SORT (if SORTTYPE=INTERNAL)
	DSIMPRINT	ASSO, DATA, DSIM
	DSSTPRINT	ASSO, DATA
	FCBPRINT	ASSO, DATA
	FDTPRINT	ASSO, DATA
	FSTPRINT	ASSO, DATA
	GCBPRINT	ASSO, DATA
	ICHECK	ASSO, DATA
	NIPRINT	ASSO, DATA
	PLOGPRINT	ASSO, DATA, PLOGs
	PPTPRINT	ASSO, DATA
	RLOGPRINT	ASSO, DATA, RLOG
	SORTPRINT	ASSO, DATA, SORT
	TEMPPRINT	ASSO, DATA. TEMP
	UIPRINT	ASSO, DATA
	VALIDATE	ASSO, DATA, SORT (if SORTTYPE=INTERNAL)
	WORKPRINT	ASSO, DATA, WORK
ADACMP	all	n/a
ADACNV	all	ASSO, DATA, WORK, dual PLOG if specified
ADADBS	ONLADD	ASSO, DATA (as specified)
	ONLINCREASE	ASSO, DATA (as specified)

Utility	Function	Data Set Accessed Directly
	OPERCOM DDIB, RESETDIB	ASSO, if nucleus is down
	others	n/a
ADADEF	DEFINE NEWWORK	ASSO, DATA, WORK ASSO, WORK
ADADRU	all	ASSO, DATA
ADAFRM	ASSOFRM, ASSORESET	ASSO
	CLOGFRM	CLOG
	DATAFRM, DATARESET	DATA
	PLOGFRM	PLOG
	RLOGFRM	RLOG
	SORTFRM	SORT
	TEMPFRM	TEMP
	WORKFRM, WORKRESET	WORK
	others	ASSO
ADAINV	all	ASSO
ADALOD	all	ASSO, DATA, WORK
ADAMER	all	n/a
ADAORD	REORDB, REORFILE, REORDATA, REORFDATA, STORE	ASSO, DATA
	REORASSO, REORFASSO, RESTRUCTUREDB, RESTRUCTUREF	ASSO
ADAPLP	WORKPRI	WORK
	PLOGPRI	Dual PLOG
	others	n/a
ADARAI	PREPARE, REMOVE	ASSO
	all others	n/a
ADAREP	all	ASSO; DATA is not required but is recommended because ADAREP tries to print the volume numbers.
	report from SAVETAPE*	n/a
ADARES	CLCOPY, PLCOPY	Dual CLOG, PLOG; ASSO is recommended but not required
	all others	ASSO
ADASAV	SAVE DB, SAVE FILE, RESTORE FILE	ASSO, DATA; also WORK if nucleus is down
ADASEL	all	ASSO
ADAULD	all	n/a
ADAWRK	all	ASSO, WORK
ADAZAP	all	ASSO

Utility	Function	Data Set Accessed Directly
	ADAZAP is to DATA	DATA
	ADAZAP is to WORK	WORK
ADAZIN	all	ASSO, DATA, WORK

* Adabas makes no attempt to open DATA and ASSO for an ADAREP from a SAVETAPE. For a regular ADAREP, an attempt is made to open DATA.

Possible Resource Conflicts

Note: See *Program Synchronization Using Operating System Services* for information about how Adabas uses operating system services to ensure that the start and end of nucleus and utility jobs are synchronized.

An Adabas utility which requires exclusive control of one or more Adabas files will be terminated with an error message under any of the following conditions:

1. The file has been locked using the operator command LOCKF (or Adabas Online System).

   In this case, you must determine why the file was locked and whether it can now be unlocked. The UNLOCKF operator command (or Adabas Online System) may be used to unlock a locked file.

2. The file is marked as *in use* by another Adabas utility.

   Adabas maintains a list of the files used by each Adabas utility in the DIB block (block 3 of the Associator). The DDIB operator command (or Adabas Online System) may be used to display this block to determine which job is using the file in question.

   It is important to note that Adabas will remove entries in the DIB only when a utility terminates normally or with an error message. If a utility terminates abnormally, the files used by that utility remain in use. The DBA may explicitly release any such files with the RESETDIB function of the ADADBS utility. For example, if the job name of the utility using the file is JOBUT1, the following ADADBS parameters would be entered:

   ```
   ADADBS RESETDIB JOBNAME=JOBUT1
   ```

   If the jobname is not unique, the IDENT parameter must be used to specify the job name qualifier for the abnormally ended utility. The DDIB command (or Adabas Online System) can be used to display the identifier:

   ```
   ADADBS RESETDIB IDENT=identifier
   ```

   The DIB may also be reset using Adabas Online System.

Caution: The DIB should be reset only if the utility in question actually terminated abnormally. Resetting the DIB for an active utility will cause unpredictable results.

3. The file is marked as in use by another application (user).

   Adabas creates a user queue element (UQE) when the first Adabas command is received from a user program or Adabas utility. The UQE contains user information (user ID, user type, etc.) and a file list. The file list identifies each file the user is using and the type of file usage. The file-usage type is one of the following:

Type	The file is opened for . . .
ACC	access only
EXU	exclusive update
EXF	exclusive use
UPD	update (ET logic user)

   See the table in the section ***File and Database Usage*** for information about file usage by Adabas utilities.

   The file list is deleted as a result of the STOPU operator command (or using Adabas Online System), when an automatic timeout occurs, or when the user issues an Adabas CL (close) command. ET/BT users file lists are released when the user is at ET status.

   Utility functions that modify the database cannot run if there is a pending autorestart, *and* the nucleus is not active.

   A pending autorestart condition can only be recovered by starting the Adabas nucleus. If the nucleus cannot execute the autorestart successfully, the only valid restart method is to run

   - ADASAV RESTORE (DATABASE); and then
   - ADARES REGENERATE.

11 Linking Applications to Adabas

- How the Adabas API Works 350
- Available Link Routines 352
- Required Work Area 353
- Required Application Reentrancy Properties 354
- Adabas Control Block (ACB) Options 356
- Extended Adabas Control Block (ACBX) Options 358
- Programming Conventions for Issuing Direct Calls 359
- Using the CICS COMMAREA or TWA with COBOL Programs 365
- Using the Adabas API in Batch Mode 368

Since most systems do not allow a standard call to Adabas, Software AG provides an application programming interface (API) to translate calls issued by an application program into a form that can be handled by Adabas.

Batch applications are supported in both single-user and multiuser mode; online operations are controlled by teleprocessing (TP) monitors. The Adabas API is available across all supported mainframe platforms; versions of the API that are specific to particular TP monitors are provided.

Adalink is a generic term that refers to the portion of the API that is specific to a particular TP monitor.

How the Adabas API Works

Online Operation

As an online operation, a request to Adabas is processed as follows:

1. The TP monitor invokes the application program. The application program must be loaded into the TP monitor region.
2. The application program invokes the Adabas API. *The Adabas API module must be installed in the TP monitor as an application module.*
3. The Adabas API takes the Adabas command passed to it from the application program and
   - builds the required control blocks and structures;
   - translates the Adabas parameter list provided by the application program call into a request that can be handled by the Adabas router or SVC;
   - includes information that identifies the user (terminal ID, TJID etc.) to Adabas.

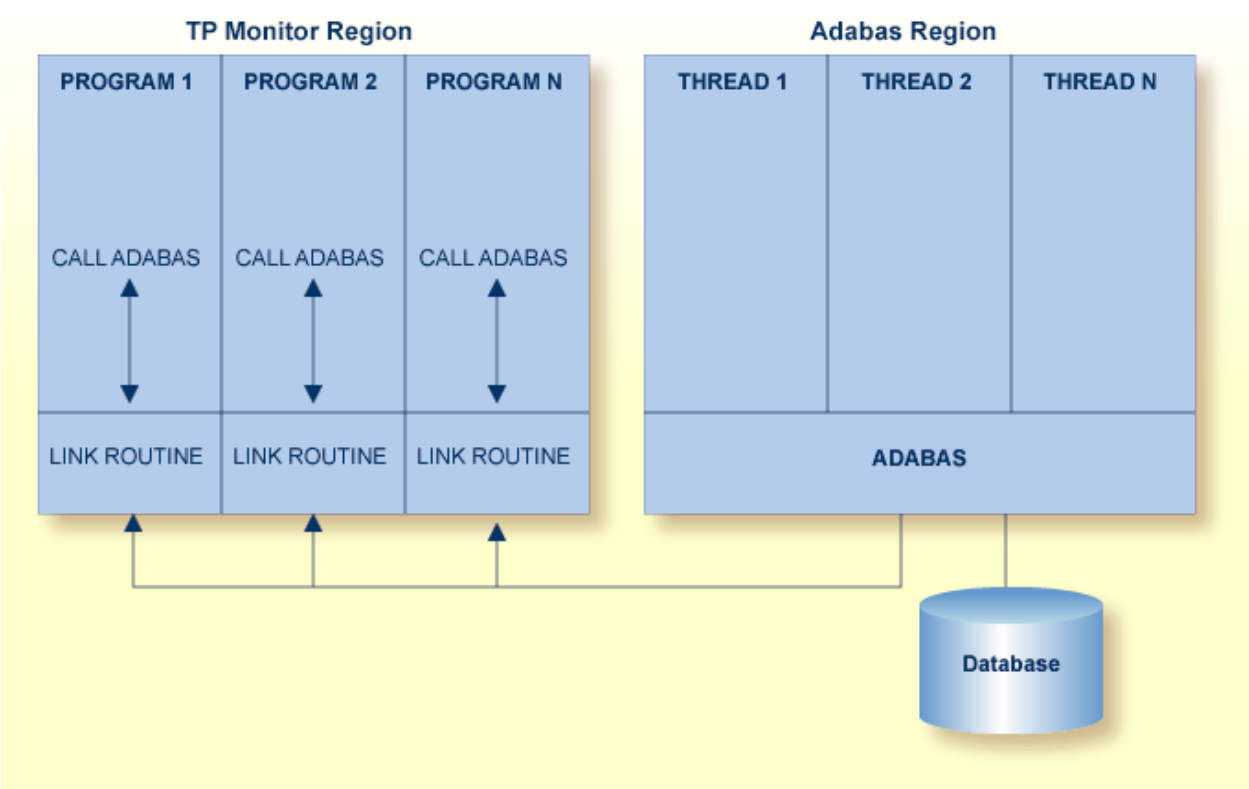
   The TP monitor's equivalent of the LINK function is used to pass the user's Adabas control block and buffers to the API.
4. The Adabas API usually uses the Adabas router or SVC (supervisor call) installed on the operating system to send the formatted request to Adabas.
5. The Adabas router or SVC moves the user's control block and buffers from the TP monitor region to the Adabas region (into the Adabas nucleus).
6. The Adabas API waits for a response from the Adabas nucleus.

   The TP monitor's equivalent of a WAIT is posted when the Adabas function is complete.
7. Adabas performs the function requested, then invokes the Adabas router or SVC, which returns the resulting data and response codes to the user application buffer.
8. The Adabas router or SVC then posts the Adabas API in the TP monitor region.

9. The Adabas API sends the response from the Adabas nucleus to the application program and returns control to the application program.
10. The application program returns control to the TP monitor.

The specific way each of the above functions is handled depends on the TP monitor used. In addition to these functions, each link routine can call one or more user exits at various processing points to provide additional capability and flexibility when making Adabas requests.

The following figure illustrates the basic configuration used by the majority of systems:



Batch Operation

As a batch operation, a request to Adabas is processed as follows:

1. The operating system loads the batch application.
2. The batch application makes an Adabas request (CALL ADABAS ...).
3. The Adabas stub program ADAUSER loads and invokes ADARUN, which then loads and invokes the batch Adabas link routine ADALNK.
4. The batch link routine transforms the Adabas request into a format suitable for use by the Adabas nucleus.

5. The link routine invokes Adabas, usually through a call to the Adabas router or SVC installed in the operating system. It also determines a unique identification for the user.
6. The link routine then waits for Adabas to handle the request.
7. Adabas, which is usually running in a different address space, processes the request and posts the link routine, returning all necessary buffers and response codes.
8. The link routine receives control and returns the Adabas buffers with response codes and data to the application.

Available Link Routines

The Adabas API is available for both batch and online applications. For online applications, the Adabas API is under the control of the TP monitor. When Adabas is installed, the Adabas API that is specific to the TP monitor in use is also installed.

The following table lists the API versions and the corresponding supported TP monitors available:

Link Routine Member	TP Monitor
ADACICS	CICS command-level.
ADACICT	CICS task-related user exit (TRUE).
ADALCO	Executable module for Com-plete.
ADALCO8	Base module for Com-plete. This member must be linked with a link globals module you prepare and with any link routine exits you require to create the final ADALCO load module that is loaded by Com-plete to service Adabas calls when Com-plete is initialized.
ADALNI	Executable module for IMS TM.
ADALNI8	Base module for IMS TM.
ADALNK	Executable module for batch/TSO.
ADALNK8	Base module for batch/TSO.
ADALNKR	Executable reentrant batch/TSO module.
ADALNKR8	Base reentrant batch/TSO module.
LNCSTUB	High-performance CICS stub.

Required Work Area

Parameters must be passed to the API. Many of the monitors do not allow standard parameter passing conventions, and the Adabas control block and buffer addresses must be moved into a special work area.

Note: If your application program is written in Natural, the necessary API work area is handled without change to the program code.

Adabas link routines do not require that the application program obtain a work area, except if your applications make ACB interface calls to the Adabas batch/TSO reentrant link routine (ADALNKR). This section describes the calling requirements peculiar to the ADALNKR routine and how applications mixing ACB and ACBX calls to the ADALNKR routine should be programmed:

- Adabas Batch/TSO Reentrant Link Routine (ADALNKR) Calling Requirements
- Mixing ACB and ACBX Interface Direct Calls to ADALNKR

Adabas Batch/TSO Reentrant Link Routine (ADALNKR) Calling Requirements

Your existing Adabas 7 applications can call the Adabas 8 ADALNKR link routine without modification.

If your application makes ACB interface direct calls using the Adabas 8 batch/TSO reentrant link routine (ADALNKR), the seventh parameter must provide a four-byte fullword address that for a work area that must be initially set to binary zeros before the first call to ADALNKR. The contents of this work area should not be modified by calling application programs thereafter. The Adabas 8 ADALNKR routine uses this work area to store the reentrancy token. If this area is modified improperly, results are unpredictable and may range from poor performance to abnormal termination of the link routine.

Likewise, if your application makes ACBX interface direct calls using the Adabas 8 batch/TSO reentrant link routine (ADALNKR), the ACBX call's reentrancy token (APLXRTOK) field must initially be set to binary zeros and remain unchanged by calling application programs thereafter.

When calling ADALNKR, it is critical to mark the last parameter address in the calling parameter list with an X'80':

- High-level languages do this automatically when their CALL statements are employed.
- Assembler applications can do this by using the CALL macro to invoke ADALNKR.

Mixing ACB and ACBX Interface Direct Calls to ADALNKR

The Adabas 8 ADALNKR routine accepts calls with the Adabas 7 parameter list with one change. To relieve the application code of the burden of obtaining the work area required by these reentrant modules, the seventh parameter should address a four-byte area initialized to binary zero, into which the ADALNKR routine will place a reentrancy token (as described earlier). This reentrancy token must remain intact between calls. If Adabas 7 and Adabas 8 calls are intermixed from the same application, this token must also be placed in the reentrancy token (APLXRTOK field) in the ACBX call.

If you provide the seventh parameter on an ACB direct call, be sure to specify the address used as the seventh parameter also in the ACBX call's reentrancy token (APLXRTOK field) to preserve user context between the logic performed by the two types of calls. Likewise, if you provide a reentrancy token on an ACBX direct call, be sure to specify the same address as the seventh parameter of the ACB direct call. If the work area address is not used in this way across ACB and ACBX direct calls, the user context for the two calling types will be different, and this could lead to incorrect results for the application.

For more information about the structure of an ACBX direct call, read *Specifying an ACBX Interface Direct Call*, in the *Adabas Command Reference Guide*.

Note that the work area used by all Adabas 8 link routines is substantially larger than the equivalent work area used in Adabas 7. This is necessary to support the additional data structures introduced in Adabas 8 and the additional context information Adabas 8 will support.

Note: If the Adabas 8 non-reentrant ADALNK is invoked using both ACB and ACBX interface direct calls, the context will be preserved because the work area is part of the ADALNK module itself.

Required Application Reentrancy Properties

Applications running under most TP monitors must use nonstandard calls to perform functions that are transparently handled by the operating system in a batch environment. In these cases, it is the reentrant properties of application code that determine how multiple users execute Adabas API calls online.

Each Adabas API version complies with the reentrancy requirements for its associated TP monitor. Application programs that use the Adabas API must also comply with the requirements for the TP monitor used.

Note: The reentrancy requirement set by the TP monitor is a minimum. For example, if the TP monitor requires a quasi-reentrant application program, a fully reentrant program will also be accepted (see CICS special requirement below). However, if a reentrant application program is required, a quasi- or non-reentrant program is not acceptable.

Ideally, code for application programs that are shared by a large number of users (commonly used TP transactions) is reentrant. The code itself never changes. All work areas are either in general registers or in user-specific work areas that are addressed by general registers. A transfer of control from one user to another requires only a change in the program counter (PSW) and the general registers. Many system routines are coded in this manner.

The PL/ I compiler produces reentrant code, but by using operating system functions that are not allowed by most TP monitors. These limitations have led to the concept of quasi-reentrancy.

A *quasi-reentrant program* may alter its code between calls to TP monitor functions. When a monitor function is invoked, all user data must be saved in a special work area obtained from the TP monitor system. The TP monitor will then schedule another user task as the active task in the system, and this task may reuse the same code. When the original user's task becomes active again, his work area is reestablished and control is passed back to the point at which the user requested a TP monitor function.

The following subsections give more detailed information about the reentrancy requirements of several TP monitors.

Com-plete: Code Reentrancy Requirements

Com-plete does not require nonstandard calling sequences: users may use standard non-reentrant code. Adabas linkage is provided by a Com-plete service routine, which is automatically included in the user's load module if Adabas calls are contained in the user program. The service routine simply passes the user parameters to Com-plete and returns control when the Adabas command has been executed.

CICS: Code Reentrancy Requirements

The Adabas 8 CICS link routine components are fully reentrant. This is possible because the Adabas task-related user exit (TRUE) is no longer optional. Because these modules are fully reentrant, they may be loaded from LPA and the task-related user exit (TRUE), ADACICT, may be defined as thread safe to CICS. To support fully reentrant and thread safe operation and to support the larger work areas required for Adabas 8, the storage working set of these link routine components has increased. Where possible, all storage areas used by the CICS Adabas 8 components are obtained above the 16 megabyte line.

Adabas Control Block (ACB) Options

The first parameter passed to the Adabas API by the application program in an ACB interface direct call is a pointer to the Adabas control block (ACB). The ACB contains information needed to process an Adabas request.

The first byte of the ACB is used by the Adabas API to determine the processing to be performed. The values for logical requests are:

Hex	Indicates ...
X'00'	a 1-byte file number (file numbers between 1 and 255)
X'30'	a 2-byte file number (file numbers between 1 and 65535)
X'40'	values greater than or equal to a blank. These are accepted as logical application calls to maintain compatibility with earlier releases of Adabas. The following calls, however, are reserved for use in special Software AG functions or products and are therefore not accepted: X'44', X'48', and X'4C'.

All other values in the first byte of the ACB are reserved for use by Software GmbH.

This section covers the following topics:

- Using One-Byte File Numbers
- Using Two-Byte File Numbers
- Using Both One- and Two-Byte File Numbers in a Single Application
- Using COBOL to Set the Control Byte

Using One-Byte File Numbers

For an application program issuing Adabas commands for file numbers between 1 and 255 (single byte), build the control block as follows:

Position	Action
1	Place X'00' in the first byte of the ACB.
9	Place the file number in the second (rightmost) byte of the ACBFNR field of the ACB. The first (leftmost) byte of the ACBFNR field is used to store the logical (database) ID or number.

If the first byte in ACBFNR is zero, the API will use either the database ID value provided in the DDDDCARD input data (ADARUN cards) or the default database ID value assembled into the link routine at offset X'80'. Applications written in Software GmbH's Natural language need not include the first byte of the ADACB because Natural supplies appropriate values.

Using Two-Byte File Numbers

Adabas permits the use of file numbers greater than 255 on logical requests. For an application program issuing Adabas commands for file numbers between 256 and 5000 (two-byte), build the control block as follows:

Position	Action
1	Place X'30' in the first byte of the ACB.
9	Use both bytes in ACBFNR for the file number, and use the two bytes in ACBRESP for the database (logical) ID.

If the ACBRESP field is zero, the API will use either the database ID from the ADARUN cards provided in DDDDCARD input data, or the default database ID value assembled into the link routine at offset X'80'.

Using Both One- and Two-Byte File Numbers in a Single Application

Because the application can reset the value in the first byte of the ACB on each call, it is possible to mix both one- and two-byte file number requests in a single application.

If this method is used, you must ensure the proper construction of the ACBFNR and ACBRESP fields in the ACB for each call type.

we recommend that an application written to use two-byte file numbers always place X'30' in the first byte of the ADACB, the logical ID in the ACBRESP field, and the file number in the ACBFNR field. The application can then treat both the database ID and file number as 2-byte binary integers, regardless of the value for the file number in use.

Using COBOL to Set the Control Byte

A programming language such as COBOL is not designed to easily manipulate single-byte values as required to establish two-byte file number support for the Adabas API. The following COBOL example illustrates one way to set these values:

```
WORKING-STORAGE SECTION
01  ACB-CONTROL
    05  ACB-TYPE           PIC 9(4) COMP.
    05  ACB-DATA REDEFINES ACB-TYPE.
          07  FILLER       PIC X.
          07  ACB-TYPE-X   PIC X.
01  ADABAS-CB.
    05  ACBTYPE            PIC X.
 .
PROCEDURE DIVISION
 .
 *  FOR SINGLE-BYTE FILE NUMBERS . . .
```

```
       MOVE 0 TO ACB-TYPE.
.
*  FOR TWO-BYTE FILE NUMBERS . . .
       MOVE 48 TO ACB-TYPE.
.
   MOVE ACB-TYPE-X TO ACBTYPE.
.
   CALL 'ADABAS' USING ADABAS-CB, . . .
.
.
```

The key to this code segment is the use of the REDEFINES clause to remap the PIC 9(4) COMP field to its constituent two bytes. Then the second byte containing the hexadecimal value for the Adabas control byte can be moved as character data to the Adabas control block.

Extended Adabas Control Block (ACBX) Options

The first parameter passed to the Adabas API by the application program in an ACBX interface direct call is a pointer to the extended Adabas control block (ACBX). The ACBX contains information needed to process an Adabas request.

The first byte of the ACBX is used by the Adabas API to determine the processing to be performed. When issuing an Adabas command, set this field to binary zeros. This indicates that a logical user call is being made (ACBXTUSR equate)

The following values in ACBXTYPE are reserved for use by Software GmbH and are therefore not accepted by application programs: X'04', X'08', X'0c', X'10', X'14', X'18', X'1c', X'20', X'24', X'28', X'2c', X'34', X'38', X'3c', X'44', X'48', and X'4c'.

Applications written in Software GmbH's Natural language need not include this first byte of the Adabas ACBX because Natural supplies appropriate values.

This section covers the following topics:

- Specifying File Numbers

- Specifying Database IDs

Specifying File Numbers

For an application program issuing Adabas commands using the ACBX interface, specify the file number in bytes 21-24 (ACBXFNR field) of the ACBX.

- For a one-byte file number, enter the file number in the rightmost byte (24); the leftmost bytes (21-23), should be set to binary zeros (B'0000 0000').
- For a two-byte file number, use the rightmost bytes (23-24) of the field and set the leftmost bytes (21-22) to binary zeros.

To specify the database ID of a file, use bytes 17-20 (ACBXDBID field) of the ACBX. For complete information about the layout of the ACBX, read *Extended Adabas Control Block (ACBX)*, in the *Adabas Command Reference Guide*.

Specifying Database IDs

For an application program issuing Adabas commands using the ACBX interface, specify the database ID in bytes 17-20 (ACBXDBID field) of the ACBX.

Specify the database ID for a call in the rightmost two bytes of this field, setting the leftmost bytes to binary zeros. At this time, only two-byte database IDs are support by Entire Net-Work.

If this field is set to binary zeros, the Adabas API uses either the database ID from the ADARUN cards provided in DDDDCARD input data, or the default database ID value provided in the LNKGBLS module linked with or loaded by the link routine.

If a file number is also needed, specify it in bytes 21-24 (ACBXFNR field) of the ACBX. For complete information about the layout of the ACBX, read *Extended Adabas Control Block (ACBX)*, in the *Adabas Command Reference Guide*.

Programming Conventions for Issuing Direct Calls

This section describes the procedures used to issue Adabas calls in direct mode from a program that is to be run under the control of one of the following teleprocessing (TP) monitors:

- Com-plete
- CICS

- IMS TM

Com-plete

Application programs that are to be run under control of Com-plete may be coded in exactly the same manner as batch programs. Since each application program is assigned a processing thread by Com-plete, the program need not be written using reentrant or quasi-reentrant code.

The following example shows an Adabas ACB interface direct call from a COBOL program that is to be run under Com-plete:

```
WORKING-STORAGE SECTION
 .
 .
01  CONTROL-BLOCK COPY ADACBCOB.
01  FORMAT-BUFFER COPY FORDEF.
01  RECORD-BUFFER COPY RECDEF.
01  SEARCH-BUFFER COPY SEADEF.
01  VALUE-BUFFER  COPY VALDEF.
01  ISN-BUFFER    COPY ISNBUF.

PROCEDURE DIVISION
 .
 .
 .
CALL 'ADABAS' USING
             CONTROL-BLOCK, FORMAT-BUFFER, RECORD-BUFFER,
             SEARCH-BUFFER, VALUE-BUFFER, ISN-BUFFER.
 .
```

CICS

Applications running under CICS use the command-level API ADACICS and the CICS Transaction Work Area (TWA) or the CICS COMMAREA to communicate parameters.

The Adabas 8 CICS link routine components accept application calls using either the TWA or the CICS COMMAREA. The COMMAREA is preferred. We strongly recommend that any new CICS applications that will make ACBX interface direct calls use the COMMAREA instead of the TWA. The new ACBX direct call interface, with its variable number of Adabas buffer descriptions (ABDs) means that the Adabas parameter list length is of indeterminate size. Therefore, sizing the TWA for each transaction is more difficult using ACBX direct calls, whereas the size of the COMMAREA used is dynamically controlled by the application and is therefore better suited for ACBX calls. For information about the format of the COMMAREA, read ***COMMAREA Formats***, later in this section.

The high-level language interface guarantees the quasi-reentrancy of COBOL, PL/I, and Assembler.

Language	Control blocks and buffers may be defined ...
COBOL	in working storage. All of working storage is copied to a user work area when a transaction is initiated.
PL/ I	as automatic storage (default storage class) variables.

Using the TWA With an ACB Call

The addresses of the Adabas control block and buffers are passed in the same way for all releases of CICS. In an ACB call, these addresses must be placed in the first six words of the TWA.

To place the ACB parameter addresses in the TWA, Software GmbH provides an Assembler subroutine that can be called from a COBOL or Assembler application program. The subroutine uses entry point ADASTWA and accepts the TWA as its first parameter.

The EXEC CICS ADDRESS TWA function is used to make the TWA addressable. The second to seventh parameters of an ACB call are the usual Adabas calling parameters. The Assembler subroutine places the parameter addresses into the TWA, and the CICS/Adabas link routine retrieves them from the TWA.

This section covers the following topics:

- Addressing the CICS TWA : Assembler
- Addressing the CICS TWA : PL/I
- Addressing the CICS TWA: VS COBOL
- Addressing the CICS TWA: COBOL II or COBOL/LE

Addressing the CICS TWA : Assembler

A CICS Assembler programmer can address the TWA directly by using an installation macro to place the addresses in the TWA and call Adabas.

Addressing the CICS TWA : PL/I

PL/ I offers a facility for addressing the TWA and obtaining the addresses of data areas. The programmer himself can place parameter addresses in the TWA. Your site may wish to establish a PL/ I preprocessor procedure to generate the calling code.

```
DCL 1        TWA  BASED  (TWAPTR),
    2    CBPTR    POINTER,
    2    FBPTR    POINTER,
    2    RBPTR    POINTER,
     .
    EXEC  CICS  ADDRESS  TWA  (TWAPTR)  END-EXEC;
     .
     .
    CBPTR=ADDR(ADA-CONTROL-BLOCK);
```

```
    FBPTR=ADDR(FORMAT-BUFFER);
    RBPTR=ADDR(RECORD-BUFFER);
```

Addressing the CICS TWA: VS COBOL

Under VS COBOL, Adabas is called using the statement:

```
EXEC CICS LINK PROGRAM ('ADABAS')
  END-EXEC.
```

```
CBL XOPTS (APOST)
IDENTIFICATION DIVISION.
 .
 .
WORKING-STORAGE SECTION.
 .
 .
01  ADABAS-CB   COPY  ADACBCOB.
01  ADABAS-FB   COPY  ADAFBCOB.
01  ADABAS-RB   COPY  ADARBCOB.
01  ADABAS-SB   COPY  ADASBCOB.
01  ADABAS-VB   COPY  ADAVBCOB.
01  ADABAS-IB   COPY  ADAIBCOB.
 .
 .

LINKAGE SECTION.
 .
 .
01  PARMLIST.
    05  FILLER    PIC S9(08) COMP.
    05  TWAPTR    PIC S9(08) COMP.

01  TWA.
    05  PARM-ADDRESSES OCCURS 7 TIMES   PIC S9(08) COMP.
 .
 .
PROCEDURE DIVISION.
 .
 .
    SERVICE RELOAD PARMLIST.
 .
    EXEC CICS ADDRESS TWA (TWAPTR) END-EXEC
    SERVICE RELOAD TWA.
 .
    CALL 'ADASTWA' USING TWA, ADABAS-CB, ADABAS-FB,
                         ADABAS-RB, ADABAS-SB, ADABAS-VB
                         ADABAS-IB.
    EXEC CICS LINK PROGRAM ('ADABAS') END-EXEC.
 .
```

```
 .
 .
```

Addressing the CICS TWA: COBOL II or COBOL/LE

Under COBOL II or COBOL/LE, Adabas is called using the statement:

```
EXEC CICS LINK PROGRAM ('ADABAS')  END-EXEC.
```

```
CBL XOPTS (APOST,ANSI85)
IDENTIFICATION DIVISION.
 .
 .
WORKING-STORAGE SECTION.
 .
 .
01  ADABAS-CB   COPY  ADACBCOB.
01  ADABAS-FB   COPY  ADAFBCOB.
01  ADABAS-RB   COPY  ADARBCOB.
01  ADABAS-SB   COPY  ADASBCOB.
01  ADABAS-VB   COPY  ADAVBCOB.
01  ADABAS-IB   COPY  ADAIBCOB.
 .
 .
LINKAGE SECTION.
 .
01  TWA.
    05  PARM-ADDRESSES OCCURS 7 TIMES   PIC S9(08) COMP.
 .
 .
PROCEDURE DIVISION.
 .
 .
 .  EXEC CICS ADDRESS TWA (ADDRESS OF TWA) END-EXEC.
 .
    CALL 'ADASTWA' USING TWA, ADABAS-CB, ADABAS-FB,
                         ADABAS-RB, ADABAS-SB, ADABAS-VB,
                         ADABAS-IB.
    EXEC CICS LINK PROGRAM ('ADABAS') END-EXEC.
 .
 .
 .
```

COMMAREA Formats

The layout of the Adabas 8 CICS COMMAREA is:

```
V8COMID  DC   CL8'ADABAS8X'       Adabas V8 COMMAREA ID
V8APLX   DS   0A                  Beginning of Adabas V8 APLX
V8ACBX   DC   A(ACBX)             Adabas ADACBX
V8RSV1   DC   A(0)                ACBX direct call reserved field
V8APLXR  DC   A(0)                ACBX direct call reentrancy token
V8ABD1   DC   A(ABD1)             First ABD
         ...
V8ABD#n  DC   A(ABD#N+X'800000')  Last ABD address
```

IMS TM

IMS message processing programs that use the Adabas API require no special link. No reentrant option is supported.

The IMS link routine for Adabas 8 is ADALNI.

This section covers the following topics:

- Adabas ACB Call Using IMS TM (Nonreentrant)

Adabas ACB Call Using IMS TM (Nonreentrant)

A nonreentrant Adabas API call under IMS TM is made like a conventional Adabas API call under batch as follows:

```
WORKING-STORAGE-SECTION.
 .
 .
01  ADA-CONTROL BLOCK   COPY  ADACBCOB.
01  FORMAT-BUFFER       COPY  FORDEF.
01  RECORD-BUFFER       COPY  RECDEF.
01  SEARCH-BUFFER       COPY  SEADEF.
01  VALUE-BUFFER        COPY  VALDEF.
01  ISN-BUFFER          COPY  ISNDEF.
 .
PROCEDURE DIVISION.
 .
 .
    CALL 'ADABAS' USING ADA-CONTROL-BLOCK, FORMAT-BUFFER,
                        RECORD-BUFFER, SEARCH-BUFFER,
                        VALUE-BUFFER, ISN-BUFFER.
 .
 .
```

Using the CICS COMMAREA or TWA with COBOL Programs

Modern COBOL programs may use either the CICS Transaction Work Area (TWA) or the CICS Communications Area (COMMAREA) to pass data to the Adabas CICS link routines. For calls using the Adabas Version 8 API, the CICS COMMAREA is preferred because it is dynamically sized on a per call basis. The Adabas CICS high-performance stub routine (LNCSTUB) uses the CICS COMMAREA if the ADAGSET keyword PARMTYP is set to ALL or COMM.

- Using the TWA
- Using the CICS COMMAREA

Using the TWA

In previous Adabas releases and with older COBOL compilers it was necessary to assemble and link-edit the provided subprogram ADASTWA with COBOL programs to properly set the Adabas parameter addresses in the CICS Transaction Work Area.

A skeleton of such a program might have the following kinds of statements:

```
WORKING-STORAGE SECTION.
         ...
         ...
      01  ADABAS-CB       PIC X(80).
      01  ADABAS-FB       PIC X ...
      01  ADABAS-RB       PIC X ...
      01  ADABAS-SB       PIC X ...
      01  ADABAS-VB       PIC X ...
      01  ADABAS-IB       PIC X ...
         ...
         ...
     LINKAGE SECTION.
      01  PARMLIST.
         05 FILLER     PIC S9(08) COMP.
         05 TWAPTR     PIC S9(08) COMP.
      01  TWA.
         05 PARM-ADDRESSES OCCURS 7 TIMES PIC S9(08) C

    PROCEDURE DIVISION
         ...
         ...
      CALL 'ADASTWA' USING TWA,
                     ADABAS-CB,
                     ADABAS-FB,
                     ADABAS-RB,
                     ADABAS-SB,
                     ADABAS-VB,
                     ADABAS-IB.
```

```
    EXEC CICS LINK PROGRAM (LINK-NAME) END-EXEC.
       ...
       ...
```

The use of the ADASTWA subprogram is perfectly acceptable with modern compilers but because these compilers offer a limited means for address manipulation, the above skeleton may be changed as follows to avoid the need to link the ADASTWA program with CICS COBOL applications.

```
   WORKING-STORAGE SECTION.
       ...
     01 TWA-LEN              PIC S9(04) COMP VALUE +0.
     01 ABEND-CODE           PIC X(4) VALUE SPACES.
       ...
     01  ADABAS-CB       PIC X(80).
     01  ADABAS-FB       PIC X ...
     01  ADABAS-RB       PIC X ...
     01  ADABAS-SB       PIC X ...
     01  ADABAS-VB       PIC X ...
     01  ADABAS-IB       PIC X ...
       ...
       ...
   LINKAGE SECTION.
       ...
     01 TWA.
       05 CB-PTR           USAGE POINTER.
       05 FB-PTR           USAGE POINTER.
       05 RB-PTR           USAGE POINTER.
       05 SB-PTR           USAGE POINTER.
       05 VB-PTR           USAGE POINTER.
       05 IB-PTR           USAGE POINTER.
       ...

   PROCEDURE DIVISION USING TWA.
       ...
   INIT-TWA.
      EXEC CICS ASSIGN TWALENG (TWA-LEN) END-EXEC.
      IF TWA-LEN = 0
         MOVE 'U649' TO ABEND-CODE
         GO TO ...
      IF TWA-LEN < 28
         MOVE 'U650' to ABEND-CODE
         GO TO ...
      EXEC CICS ADDRESS TWA (ADDRESS OF TWA) END-EXEC.
       ...
      SET  CB-PTR TO ADDRESS OF ADABAS-CB.
      SET  FB-PTR TO ADDRESS OF ADABAS-FB.
      SET  RB-PTR TO ADDRESS OF ADABAS-RB.
      SET  SB-PTR TO ADDRESS OF ADABAS-SB.
      SET  VB-PTR TO ADDRESS OF ADABAS-VB.
      SET  IB-PTR TO ADDRESS OF ADABAS-IB.
```

```
      ...
    EXEC CICS LINK PROGRAM (LINK-NAME) END-EXEC.
      ...
```

Using the CICS COMMAREA

The CICS TWA size is part of the CICS transaction definition for a given set of application programs and its size may not be dynamically altered during execution of application programs running under that CICS transaction. For this reason the CICS Communications Area (COMMAREA) is recommended for Adabas Version 8 programs. The COMMAREA size may be altered as needed by the requirements of the data to be passed between programs using it. The Adabas Version 8 parameter list and data structures are dynamic in size, so the COMMAREA is the best method for passing data between CICS applications and the Adabas Version 8 CICS link routines.

Using the address manipulation features of a modern COBOL compiler, a sample skeleton of the statements needed to use the CICS COMMAREA might be:

```
WORKING-STORAGE SECTION.
     ...
     ...

  01 ADA-COMM-AREA.
    05 COMMID           PIC X(8) VALUE 'ADABAS52'.
    05 CB-PTR           USAGE POINTER.
    05 FB-PTR           USAGE POINTER.
    05 RB-PTR           USAGE POINTER.
    05 SB-PTR           USAGE POINTER.
    05 VB-PTR           USAGE POINTER.
    05 IB-PTR           USAGE POINTER.
     ...
  01  ADABAS-CB       PIC X(80).
  01  ADABAS-FB       PIC X ...
  01  ADABAS-RB       PIC X ...
  01  ADABAS-SB       PIC X ...
  01  ADABAS-VB       PIC X ...
  01  ADABAS-IB       PIC X ...
     ...
  01 ADA-COMM-AREA-LENGTH PIC S9(4) COMP.
     ...

PROCEDURE DIVISION.
     ...
     ...
 INIT-ADA-COMM-AREA.
    SET  CB-PTR TO ADDRESS OF ADABAS-CB.
    SET  FB-PTR TO ADDRESS OF ADABAS-FB.
    SET  RB-PTR TO ADDRESS OF ADABAS-RB.
    SET  SB-PTR TO ADDRESS OF ADABAS-SB.
    SET  VB-PTR TO ADDRESS OF ADABAS-VB.
    SET  IB-PTR TO ADDRESS OF ADABAS-IB.
```

```
        MOVE 32 TO ADA-COMM-AREA-LENGTH.
         ...
        EXEC CICS LINK PROGRAM ('ADABAS')
             COMMAREA(ADA-COMM-AREA)
             LENGTH(ADA-COMM-AREA-LENGTH)
        END-EXEC.
```

Using the Adabas API in Batch Mode

The Adabas API in batch mode uses a standard call with a parameter list in register 1 and register 13 pointing to a register save area. This convention is supported by all major programming languages through their CALL mechanisms.

Under most mainframe operating systems, the batch API (ADALNK or ADALNKR) can either be linked directly with the batch application module or it can be loaded by ADAUSER. Software AG *strongly* recommends that batch applications be linked with ADAUSER and not the ADALNK or ADALNKR.

ADAUSER and ADARUN with the Adabas API

The ADAUSER module can be linked with the Adabas API. ADAUSER provides upward compatibility with Adabas releases and a degree of isolation from changes that might be made in the API or the Adabas SVC in the future.

Each user program to be executed should be linked with the Adabas version-independent module ADAUSER, which dynamically loads the Adabas control module ADARUN. For batch mode execution, the user program should be linked with ADAUSER to achieve maximum environment independence, as shown below:

User Program linked with ...	Advantage
ADAUSER, ADARUN, and ADALNK	independent of mode and Adabas version
ADARUN and ADALNK	independent of mode only
ADALNK	none; no version- or mode-independence

The following section illustrates the JCL required to link the batch application module with ADAUSER.

Link Example

```
// EXEC LKED,PARM='NCAL'
//LKED.SYSLMOD   DD           ... User Library
//LKED.ADALIB    DD           ... Adabas Library
//LKED.SYSIN     DD *
    INCLUDE SYSLMOD(USERPGM)
    INCLUDE ADALIB(ADAUSER)
    ENTRY USEREP                (see note)
    NAME USERPROG(R)
/* ↩
```

Note: The entry point, if specified, must be the entry point of the user program.

Batch Execution Modes

When executing under batch, the program can be run in either single-user or multiuser mode:

- Single-user mode runs the application program, the batch API, ADARUN, and the Adabas nucleus in the same address space.
- Multiuser mode executes the application program and the Adabas API in an address space separate from the Adabas nucleus.

The recommended mode of operation is multiuser mode. The user must provide only those job control statements required by ADARUN and the user program.

Multiuser Mode Example

The following example assumes that the user program USERPROG has been linked with the module ADAUSER and is to be executed in multiuser mode.

```
// EXEC PGM=USERPROG
//STEPLIB   DD        ... User Library
//          DD        ... Adabas Library
//DDDDCARD    DD *
ADARUN MODE=MULTI
//DDPRINT   DD SYSOUT=*
//... user  DD statements ... ↩
```

Execution in Single-User Mode

In single-user mode, the appropriate Adabas nucleus JCL must be included with the JCL of the user program. This includes job control statements to define the Adabas data sets for the Associator, Data Storage, the Work data set, and any data sets for protection or command logging. For more information about Adabas runtime job control requirements, see ***Adabas Session Execution***.

12 Adabas Restart and Recovery

- Work Data Set 372
- Protection Log 373
- Restart Operations 375
- Database Recovery 377

A user session is a sequence of Adabas calls optionally starting with an OP command and ending with a CL command. A user is either a batch mode program or a person using a terminal. The user can be identified by an 8-byte unique ID provided with the OP command. This ID enables Adabas to retain restart information (ET data) beyond the end of a session.

For communication purposes, a terminal user is identified by machine, address space, and terminal ID, thereby ensuring that each user ID is unique.

During an Adabas session (from activation to termination), the Adabas nucleus creates a sequence of protection entries in exact historical sequence reflecting all modifications made in the database. The sequence of protection entries is written to the Work data set (part 1) and to a protection log in the form of blocks. Each block contains the nucleus session number, a unique block number, and a time stamp.

Work Data Set

Part 1 of the Work data set (ADARUN LP parameter) stores the most recent protection entries using a wrap-around method.

The protection entries on the Work data set are used to execute a BT command and to execute autorestart/autobackout when reactivating the Adabas nucleus after a system failure. The entries may contain all or some of the following:

- before/after images of a data record;
- before/after images of elements of the inverted lists (DVT);
- special before-images of inverted list blocks for automatically repairing the database after a system failure;
- checkpoint entries;
- ET entries including ET data;
- special entries for handling the internal restart procedures.

Adabas identifies a batch user by checking a store clock (STCK) value during the program's first Adabas call.

Protection Log

The protection log contains the same entries as part 1 of the Work data set (except the special before-images mentioned in the **discussion of the Work data set**. Additional entries on the protection log which are not stored on the Work data set include an entry for the infrequently used C5 data and an after-image Associator or Data Storage block written during buffer flush; the latter occurs while an online SAVE function of the ADASAV utility is running.

All protection log entries created by the nucleus describe the modifications made to the database in exact historical order. Each block is associated with a block sequence number.

The protection log can be written either:

- directly to a sequential protection log (DDSIBA) data set; or
- to multiple protection log (DDPLOGR1, DDPLOGR2, ... DDPLOGR8) data sets.

Note: Adabas still supports dual protection logging using DUALPLD/S and user exit 2.

Multiple protection log data sets are each physical data sets of the same size and block length, randomly accessed, and used in succession. This means that one of the data sets can be used and written by the nucleus while others are being copied in order for archive purposes.

Sequential Protection Log

The sequential protection log data set is opened by the nucleus when a session is started and closed when a session is ended. In general, this data set is assigned to tape to avoid disk space problems that may cause an unexpected abnormal termination. Because the amount of data to be written to such a data set depends upon the amount of activity of all users, estimating disk space is difficult.

At the end of the Adabas session, an end-of-file mark (EOF) is written to the tape to indicate end of session. Adabas supports multivolume protection log data sets. If one reel is not enough to store all protection log entries, subsequent tape reels can be used.

The nucleus writes a checkpoint for each volume written. This checkpoint contains information about session number, volume serial number, and block sequence numbers.

The ADARES COPY function must be used to copy such a sequential protection log data set for archive purposes.

If a nucleus session ended abnormally, the final EOF mark on the tape may not be written. This may cause problems if the protection log is used directly as input to the ADARES BACKOUT or REGENERATE function. The COPY function of ADARES is able to detect the logical end of the in/out protection log and writes a valid EOF mark to the output. ADARES writes checkpoints in the same way the nucleus does for each output volume.

Note: When using a sequential protection log on tape, one tape unit is allocated for the Adabas nucleus for the duration of the entire nucleus session. This session can last a very long time, during which the log tape unit must be available; a single sequential logging tape is therefore not adequate in every case. Software GmbH therefore recommends using multiple protection logging.

Multiple Data Set Protection Log

The Adabas multiple data set protection log consists of two to eight data sets (DDPLOGR*n* where *n* is the sequential number of the data set) with the following attributes:

- fixed block size;
- reside on DASD;
- preformatted by ADAFRM;
- all data sets have the same number of blocks and identical block sizes;
- all data sets can be shared by the nucleus and other utilities (ADARES).

Assuming newly formatted dual or multiple protection log data sets, Adabas selects DDPLOGR1 at startup and starts writing protection log entries to it. Writing starts at block 2. Block 1 contains status information about the data set. Other PLOG data sets are still unused. Protection log entries are written to multiple protection log data sets in the same order they are written to a sequential log.

Each protection log data set need not be large enough to accommodate all protection log entries for a session. When one data set becomes full, protection log switching occurs as follows:

1. status information is written to block 1 to terminate the current protection log data set;
2. there is a switch to another data set;
3. a message is written to the operator and to the log output; and
4. user exit 12 is called (see below).

While the nucleus continues writing protection log entries to the other data set, the first one is copied to a sequential data set by ADARES PLCOPY. ADARES can be started manually or initiated by user exit 12, which is called whenever a switch from one protection log data set to another occurs. ADARES writes a checkpoint for each output volume written. This checkpoint contains the session number, volume serial number, and block sequence number.

A protection log switch may occur more than once in a single session. The content of each protection log data set must be copied to a single sequential data set. All subsequent copies produced within one session are logically equivalent to the information the nucleus would have written to a sequential protection log (DDSIBA).

All sequential copies can be concatenated to form a single sequential data set containing all protection log entries for a session. In fact, a sequential copy is required as input by the BACKOUT/REGENERATE functions of ADARES.

Note: A tape unit to store the sequential protection log entries is required only during the ADARES PLCOPY run.

If multiple data set protection logging is used, but user exit 12 is not available to call ADARES PLCOPY, protection log switching occurs as follows:

1. the current protection log data set is closed; and
2. if no other data set is empty, the following message is issued and the old data is overwritten:

   ```
   Now it's too late to copy DDPLOGRn
   ```

   In this case, protection log information is lost.

Restart Operations

Protection entries are needed if any of the following fail:

- a user application program
- Adabas
- the operating system
- the hardware

This section covers the following topics:

- Restart after a User Application Program Failure
- Restart after an Adabas, Operating System, or Hardware Failure
- Restart after a Power Failure
- Using Automatic Restart Management (ARM)

Restart after a User Application Program Failure

An application program that is in the middle of a transaction can detect that the transaction cannot be completed successfully. Removing the first portion of the transaction, called back out or roll back, is performed by the BT command.

The BT command is executed by reading the Work data set backwards and executing the entries for the specific transaction in reverse (after-image is used to scratch an element in the database, before-image is used to insert an element in the database). The start-transaction bit in an element serves as the stop indicator for the BT process.

Restart after an Adabas, Operating System, or Hardware Failure

When Adabas is reactivated after any failure that caused the Adabas nucleus to terminate abnormally (that is, failure of Adabas, the operating system, or hardware), an automatic procedure is executed to bring the database to a physically and logically valid status. All partially executed update commands are reset. All incomplete transactions are backed out.

This automatic procedure comprises three steps:

1. repair the database
2. autorestart
3. autobackout

The repair is needed to modify the database to the status it would have if a buffer flush had just been completed at the time of the failure. In other words, all blocks in the database are at a status that enables the nucleus to perform normally by addressing Data Storage records through the address converter and normal index entries through the upper index.

Autorestart backs out updates of single update commands that were partially executed when the system failed; Autobackout backs out updates of user transactions that were partially executed when the system failed.

The major protection entries used for autorestart and autobackout are the before-images and after-images of Data Storage and the inverted lists (DVT).

Restart after a Power Failure

Depending on the hardware, a power failure during an I/O operation may damage the Adabas blocks that were being processed. This damage cannot be detected during autorestart and therefore can result in problems later, such as unexpected response codes of lost database updates.

Note: If the cause of the abend was a power failure, we strongly recommend recovering the affected files using the ADASAV and ADARES utilities as described in the section *Database Recovery* .

Whenever an Adabas session is reactivated with the IGNDIB=YES parameter, which forces the new session to ignore an existing session communication block (DIB) in the Associator, Adabas checks whether a buffer flush was active when the abend occurred. If a buffer flush was in process, the autorestart shuts down and issues an ADAN58 message:

```
BUFFER-FLUSH START RECORD DETECTED DURING AUTORESTART. THE NUCLEUS WILL TERMINATE
AFTER AUTORESTART. IN CASE OF POWER FAILURE, THE DATABASE MIGHT BE INCONSISTENT...
```

The message also includes a list of the files that were being updated when the buffer flush was in process. In this case, the DBA must check whether the cause of the abend was a power failure.

If the abend was *definitely not* a power failure and the integrity of the information on the output hardware can be guaranteed, the database can be reactivated immediately. Database recovery is not necessary.

Using Automatic Restart Management (ARM)

Automatic restart management (ARM) is 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 non-cluster environments.

The ADARUN parameter **ARMNAME** 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.

Database Recovery

If an Adabas, operating system, or hardware failure occurs and the physical database is still readable (which is the normal case), the Adabas nucleus automatically takes all necessary steps to ensure that database processing can be continued in a normal manner.

Database recovery is described in the following topics:

- Recreating a Database
- Database or File Recovery Considerations
- Database Recovery Guidelines
- Using the Adabas Recovery Aid (ADARAI)

Recreating a Database

In case of a head crash (physical damage to the database), application program error (logical damage to the database), or a power failure during a buffer flush (described in the section ***Restart after a Power Failure***, the utilities ADASAV and ADARES must be used to recreate the database.

To restore and regenerate the entire database

1. Restore the database with ADASAV using the sequential data set containing the most recent copy of the database.
2. Restore the database blocks that were updated during the SAVE of the database (online SAVE).
3. Regenerate with ADARES from the checkpoint taken at the end of the SAVE function to the latest point at which the database was still intact (this is done automatically by Adabas).

To restore and regenerate a single file of the database

- Use the steps described for restoring and regenerating an entire database, but make the appropriate changes in the parameter statements of the utilities.

To restore and regenerate single blocks of Data Storage

- Use the REPAIR function of ADARES.

To restore the status of a database at the start of a single batch update run

1. If a long-running batch program performs a large number of erroneous updates (e.g., logic error in program), and the program was the only user performing updates, it may be desirable to back out all updates performed by the program.
2. Use the ADARES BACKOUT function with the sequential, dual, or multiple protection log.

Database or File Recovery Considerations

The information written on the protection log (SIBA/PLOG) contains record-based information, including the record identifiers (ISNs). The ADARES utility passes the ISN-based information to Adabas; however, Adabas can neither verify the validity of the ISN nor check that the logical content of the record is correct.

Therefore, you must ensure that the combinations of ISN and logical record used in the original session are also used during the procedure for recreating the file or database. Use the ISNs and parameters that were specified when the file was originally loaded.

Database Recovery Guidelines

To ensure that the database can be recovered in the event of a software or hardware failure, we recommend that the DBA

- create a sequential data set copy of the database using the ADASAV utility (SAVE function) and archive the output data set (normally on tape). This data set reflects the status of the database at a specific time.
- retain all protection log data written to the protection log data set for each nucleus session.

If you are using sequential protection logging (SIBA), you can use the ADARES COPY function to archive the log information. If you are using multiple data set protection logging (PLOG), use the ADARES PLCOPY function.

Each nucleus session is identified by a unique session number. This number is assigned to all SAVE and protection log data.

Example 1: Inactive Nucleus Save Operation

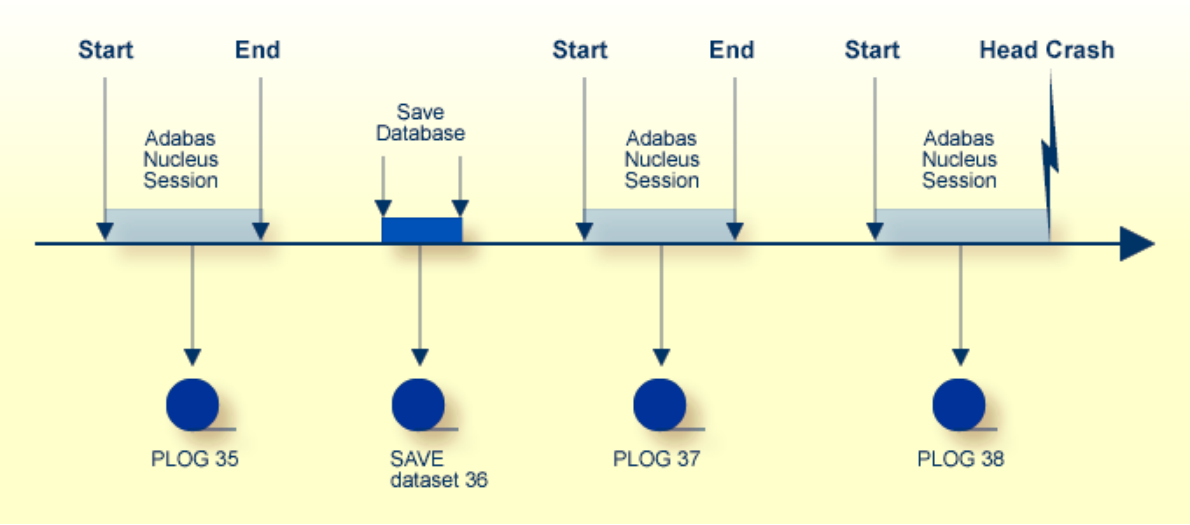
To recreate the database to the status of the failure

1 Restore the database using SAVE data set 36.

With the nucleus *not* active, execute

```
ADASAV RESTORE
```

2 Start the Adabas nucleus.



3 Reapply all modifications made in session 37.

With the nucleus active and running, execute

```
ADARES REGENERATE PLOGNUM=37
```

4 Reapply all modifications made in session 38.

With the nucleus active and running, execute

```
ADARES REGENERATE PLOGNUM=38
```

Example 2: Active Nucleus Save Operation

This example shows the assignment of session numbers to SAVE and protection log data sets when a SAVE database function is executed parallel to an active nucleus.

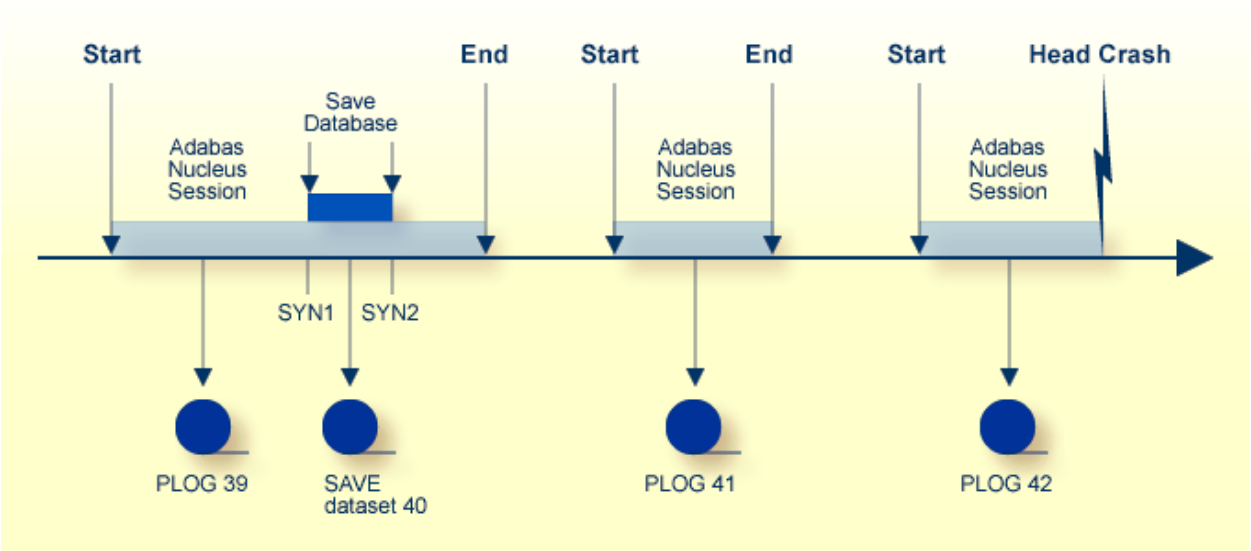
The nucleus writes a protection log number 39.

When ADASAV starts the SAVE function, the nucleus writes a SYN1 checkpoint to the data protection log.

At the end of the SAVE function, a SYN2 checkpoint is written. The SYN2 checkpoint is a synchronized checkpoint where all users are at ET status.

All writes of the nucleus to the Associator and Data Storage are written to the protection log as well (between SYN1 and SYN2).

The output of ADASAV is associated with session number 40. The nucleus still writes to protection log 39, even if the SYN2 checkpoint has been written.



To reset the database to the status of the failure

1 Restore the database using SAVE data set 40 as input.

   Modifications made to the database during the online SAVE operation are found on protection log 39, starting at the SYN1 checkpoint. ADASAV also requires the correct protection log input.

2 Stop the nucleus; then execute

```
ADASAV RESTONL PLOGNUM=39,SYN1=blocknumber
```

3 Start the Adabas nucleus.

4 Reapply modifications made in session 39, starting from the SYN2 checkpoint.

With the nucleus active, execute

```
ADARES REGENERATE PLOGNUM=39,FROMCP=SYN2,FROMBLK=blocknumber
```

5 Reapply modifications made in session 41.

With the nucleus active, execute

```
ADARES REGENERATE PLOGNUM=41
```

6 Reapply modifications made in session 42 up to the time of failure.

With the nucleus active, execute

```
ADARES REGENERATE PLOGNUM=42
```

To find the SYN1 and SYN2 block numbers needed as input for ADASAV and ADARES, check the output report written by the ADASAV SAVE function or perform the ET Checkpoint function in Adabas Online System.

Using the Adabas Recovery Aid (ADARAI)

Transaction recovery is provided whenever an Adabas session is abnormally terminated. The Adabas autobackout routine, which is automatically invoked at the beginning of every Adabas session, removes the effects of all interrupted transactions from the database.

However, when a database data set (ASSO, DATA, or WORK) is destroyed, it is necessary to restore and regenerate the database to recover the lost data.

The Adabas Recovery Aid helps automate and optimize database recovery. It records and reports all information needed to recover the database and builds the recovery job stream (JCL), which is the basis for reexecuting the jobs performed from the time of the last SAVE to the point of failure and error. For information, see the description of the ADARAI utility in the Adabas Utilities documentation.

13 Local Time Zone Change Support

- Time Stamp Overview 384
- Activating the Local Time Zone Change 384

The Adabas nucleus reflects dynamic changes made in the operating system to the local time zone (relative to machine or STCK time) without being shut down and restarted. The change becomes effective when the next operator command is issued from the console.

Time Stamp Overview

Time stamps on WORK and PLOG are in machine time rather than local time. This means that:

- Time stamps passed to user exit 2 as the second and third parameters are in machine time rather than local time. User exits that display the time stamps on the console or otherwise process the time stamps are thus affected.

  Note: The sample user exit 2 delivered by Software GmbH does *not* display or otherwise process these time stamps.

- Time stamps in PLOG records written WITH LOGINFO by the ADASEL utility to a DDEXPA*n* (or EXPA*n*) output data set are in machine time rather than local time.

  All other time stamps in Adabas data sets (for example, in save data sets, checkpoints, FCBs, the CLOG) are in local time.

Activating the Local Time Zone Change

To activate the local time zone change:

1. Change the local time zone in the operating system.
2. On z/OS, issue any command (for example **DSTAT**) from the operator console to the Adabas nucleus.

   Notes:

   1. The ADADBS OPERCOM function may *not* be used to activate the local time zone change. The command must be issued from the operator console.
   2. If commands were being logged on the CLOG when the local time zone was changed, we recommend that you use the FEOFCL command from the operator console to activate the change. In this way, the same dual CLOG will not contain time stamps from before the change as well as after its activation.

14 Adabas Timeout Feature

- Timeout Parameters 386
- Action Taken If Time Limit Exceeded 387

Each user who calls Adabas is assigned a user queue element (UQE) that indicates the nucleus resources allocated to the user. If a user calls Adabas and is then inactive for an extended period (for example, because a terminal operator leaves the terminal without logging off), that user's resources are locked and are not available to other users.

Adabas provides a timeout mechanism which prevents this potential waste of resources. The Adabas timeout function allows control of session-wide or individual user time limits for overall transaction time, and for specific user types.

Timeout Parameters

The following ADARUN timeout parameters can be set at nucleus startup:

Parameter	Specifies the maximum elapsed time (in units of 1.048576 seconds) ...
TNAA	an access-only user may be active without issuing an Adabas call.
TNAE	an ET-logic user may be active without issuing an Adabas call.
TNAX	an exclusive-control user may be active without issuing an Adabas call.
TT	permitted for a logical transaction by an ET-logic user.

If not specified, a timeout parameter value defaults to 900 or slightly over 15 minutes.

These parameter values, which apply to all users in the session, can be overridden for a single user with an Adabas OP command for the user session. The maximum values for such OP command overrides are specified by the following ADARUN timeout parameters set at nucleus startup:

Parameter	Specifies the maximum elapsed time (in units of 1.048576 seconds) ...
MXTNA	any individual user may be active without issuing an Adabas call.
MXTT	permitted for a logical transaction by an individual ET-logic user.

The Adabas ADADBS OPERCOM utility or Adabas Online System may also be used to change the timeout values.

The following section describes the specific actions taken for each type of timeout.

Action Taken If Time Limit Exceeded

This section lists the actions the nucleus *may* take when a time limit expires or the STOPU command is issued and describes the circumstances that determine the action.

Notes:

1. The DELUI/DELUF operator commands, the ADADBS OPERCOM STOPI/STOPF commands with the PURGE option, or their Adabas Online System (AOS) counterparts are handled like STOPU.
2. The STOPI/STOPF operator commands, the ADADBS OPERCOM STOPI/STOPF commands without the PURGE option, or their AOS counterparts are handled as a non-activity timeout.

Back Out Transaction

The nucleus backs out a transaction only for an ET user that has an open transaction.

The transaction is backed out for all kinds of timeout and for STOPU commands.

Release Command IDs

When the nucleus releases command IDs, it also erases the file list and deletes the ETID in the user queue element (UQE).

This action is taken for all TNA*x* timeouts and for the STOPU command.

It is *not* taken for a TT timeout.

Close User Queue Element

The nucleus closes the user queue element (UQE) for all TNAx timeouts when OPENRQ=YES and for the STOPU command.

Note: If OPENRQ=YES is in effect for the session and a user does not start with an OP command, the user receives response code 9 (ADARSP009).

This action is *not* taken for TNAx timeouts when OPENRQ=NO (with one exception described in the next paragraph) or for a TT timeout.

The nucleus closes the UQE for a TNAE timeout when OPENRQ=NO only when the user

- has no ETID;
- has no open transaction; and

- has not issued an OP command with R option.

Note: When OPENRQ=NO and the UQE is closed, the user's next call does not receive a response code 9 (ADARSP009) because the nucleus no longer knows that it timed out the user.

Set Timeout Status in the UQE

The nucleus sets the timeout status when the UQE is not closed.

Note: If a user issues a call and the timeout status is set in the UQE, the user receives response code 9 (ADARSP009).

Index

Symbols

64-bit virtual storage option, 222

A

ABDs
enable/disable logging of, 131
ACBX
enable/disable logging of, 131
Activating
audit of AOS and utility functions, 56
Adabas
overview of operating structure, 11
Adabas buffer descriptions (ABDs)
enable/disable logging of, 131
Adabas Caching Facility
adjusting shared cache size for, 85
Adabas Cluster Services
ADARUN parameter to enable, 92-94
Adabas command logged buffer size limit
ADARUN parameter, 83
Adabas control block
logging
command to start, 289
command to stop, 297
Adabas Delta Save Facility
parameter to enable/disable, 106
parameter to specify user exit routine, 107
Adabas Delta Save Facility Facility
display status
using operator command, 263
Adabas error handling and message buffering facility
commands, 318
Adabas Event Log
specifying size, 119
Adabas Fastpath
parameter to enable/disable, 112-113
Adabas Parallel Services
ADARUN parameter to enable, 92-94
Adabas Recovery Aid
function of, 381
Adabas Review
command to deactivate, 302
hub ID
command to set/modify, 302
local mode
command to switch to, 302
parameter to set, 186
Adabas SMF records
enabling, 198
enabling interval records, 201
enabling type 89, 198
selecting detail SMF record sections, 199
setting user-defined record number, 202
SMF user exit module name, 220
specifying subsystem name, 202
Adabas sysplex cluster
nucleus ID
parameter to specify, 167
Adabas Transaction Manager
ATM manager logging option, 213
external transaction coordinator support, 215
global transaction time limit, 212
identifying databases for ET data, 211
restart action control, 214
severity level for suppression of warning messages, 214
size of the ATM request queue, 210
TCID client ID prefix, 215
Adabas Triggers and Stored Procedures
parameter to activate, 204
Adabas Vista
parameter to enable/disable, 223
ADACHK utility
example of DSTAT command usage, 317
ADACICS
CICS command-level link routine for Adabas 7 and 8, 352
ADACICT
link routine Adabas 7 and 8 CICS task-related user exit (TRUE), 352
ADAEND
operator command, 248
ADAIOR
block size of sequential data set
parameter to specify, 182
ADALCO
link routine for Com-plete, 352
ADALCO8
link routine for Com-plete, 352
Adalink
definition of, 350
ADALNI link routine
for IMS TM, 352
ADALNI8 link routine
for IMS TM, 352
ADALNK link routine
for batch and TSO, 352
ADALNK8 link routine
for batch and TSO, 352

ADALNKR link routine
for batch and TSO, 352
ADALNKR8 link routine
for batch and TSO, 352
ADALOD utility
example of DSTAT command usage, 318
ADAORD utility
example of DSTAT command usage for, 318
ADARES utility
example of DSTAT command usage for, 318
ADARUN
directory of parameters, 17, 233
guidelines for setting parameters, 227
parameter syntax, 15
parameters specific to an operating system, 232
using session statistics to determine settings, 232
ADARUN parameter table, 49
ADARUN parameters
controlling RI command behavior, 191
logged buffer size limit for Review, 188
LRPL, 137
maximum number of records per buffer, 189
MLWTO, 143
REVBUFMF parameter, 185
REVLOGBMAX parameter, 188
REVLOGMAX parameter, 188
REVLOGNR parameter, 189
REVSTHMV parameter, 190
REVTIMER parameter, 190
RIAFTERUPDATE parameter, 191
RPLCONNECTCOUNT, 192
RPLCONNECTINTERVAL, 192
RPLPARMS, 192
RPLSORT, 193
RPWARNINCREMENT, 194
RPWARNINTERVAL, 195
RPWARNMESSAGELIMIT, 195
RPWARNPERCENT, 196
RVCLIENT, 196
SECUID, 197
send interval in seconds, 190
send to hub mode version, 190
size of the Review Buffer Pool as temporary storage, 185
total logged buffer size limit for a Review command, 188
ADASAV utility
abnormal termination
command to check status, 302
example of DSTAT command usage for, 318
ADASNAP data set, 321, 334
ADAULD utility
example of DSTAT command usage for, 318
ADAUSER
Adabas `stub' program, 351
link with Adabas API, 368
advance-lock
obtaining for a file, 248
releasing for a file, 301
releasing for all files, 301
alert messages
PLOG and CLOG, 291
alert messages for CLOG and PLOG, 133
ALOCKF
operator command, 248
ALOGDEV
ADARUN parameter, 49
ALOGDEV parameter, 22
ALOGSIZE
ADARUN parameter, 50
ALOGSIZE parameter, 22
AOSLOG
ADARUN parameter, 50
operator command
using console, 251
AOSLOG parameter, 22
API
batch operation, 368
build the ACBX, 359
build the control block, 356-357
code reentrancy requirements, 354
control block options, 356, 358
database IDs, 359
file numbers, 359
issue direct calls to Adabas, 359
link applications to Adabas, 350
one-byte file numbers, 356
passing parameters to, 353
pointer to the Adabas control block (ACB), 356
pointer to the extended Adabas control block (ACBX), 358
required work area, 353
two-byte file numbers, 357
versions of, 352
work area defined by DSECTs, 353
AREXCLUDE
ADARUN parameter, 51
AREXCLUDE parameter, 22
ARM
parameter to activate, 51
ARMNAME
ADARUN parameter, 51
ARMNAME parameter, 22
ARNWORKBUFFERS
ADARUN parameter, 52
ASSOCACHE
ADARUN parameter, 53
ASSOCACHE parameter, 22
Associator
automatic increase size command, 253
automatic increase threshold setting, 255
parameter to
specify device type for storing first block, 105
setting maximum size, 255
space notifications, 252
Associator (ASSO)
determining space defined, 276
Associator data set
parameter to enable/disable controller caching for, 53
space notifications, 54
ASSOSPACEWARN
ADARUN parameter, 54
operator command, 252
ASSOSPACEWARN parameter, 22
ASYTVS
ADARUN parameter, 55
operator command
using console, 253
ASYTVS parameter, 22
attached buffer
parameter to set time limit for hold, 98

attached buffers
command to display usage, 274
number of
parameter to specify, 155
AUDCONNECT
operator command, 253
audit log
dual
switch to other, 285
AUDITING
ADARUN parameter, 55
AUDITING parameter, 22
AUDITLOG
ADARUN parameter, 56
AUDITLOG parameter, 22
AUTCQENV parameter, 22
autobackout
definition of, 376
AUTOCQENV
ADARUN parameter, 56
AUTOCQTIME
ADARUN parameter, 57
AUTOCQTIME parameter, 23
AUTOINCASSOSIZE
ADARUN parameter, 58
operator command, 253
AUTOINCASSOSIZE parameter, 23
AUTOINCASSOTHRESHOLD
ADARUN parameter, 59
operator command, 255
AUTOINCASSOTHRESHOLD parameter, 23
AUTOINCASSOTOTAL
ADARUN parameter, 60
operator command, 255
AUTOINCASSOTOTAL parameter, 23
AUTOINCDATASIZE
ADARUN parameter, 61
operator command, 256
AUTOINCDATASIZE parameter, 23
AUTOINCDATATHRESHOLD
ADARUN parameter, 63
operator command, 257
AUTOINCDATATHRESHOLD parameter, 23
AUTOINCDATATOTAL
ADARUN parameter, 64
operator command, 258
AUTOINCDATATOTAL parameter, 23
Automatic Restart Management (ARM)
explanation, 377
autorestart
definition of, 376
parameter to activate ARM, 51
parameter to control WORK I/O buffers, 52
parameter to exclude files from processing, 51
progress message interval, 120

B

batch
ADALNK link routine, 352
ADALNK8 link routine, 352
ADALNKR link routine, 352
ADALNKR8 link routine, 352
API work area for, 353
process a request to Adabas, 351
using the Adabas API in, 368
bold, 8
braces ({}), 9
brackets ([]), 9
buffer flush
asynchronous
based on volume serial number, parameter to disable, 55
operator command to activate using console, 253
parameter to enable and set I/O pool size, 127
parameter to limit number of parallel I/O operations, 113
modify number of I/Os, 286
synchronous
parameter to specify duration of, 209
buffer pool
Adabas
parameter to set maximum size of, 123
attached
space allocation, 155
internal format
parameter to set size of, 128
buffers
Adabas command logged buffer size limit, 83
logged buffer size limit, 81

C

CACHE
ADARUN parameter, 65, 76
cache
shared cache, 89
CACHE parameter, 23
cache structure
name, 85
ratio
directory entries to elements, 105
statistics display, 280, 283
cache vector size, 85
caching
controller
parameter to enable/disable controller caching for Associator data set, 53
parameter to enable/disable controller caching for Data Storage data set, 102
parameter to enable/disable controller caching for Sort data set, 203
parameter to enable/disable controller caching for Temp data set, 208
parameter to enable/disable controller caching for Work data set, 223
caching facility
Associator cache space unit size, 68
Data Storage cache space unit size, 74
excluding files from caching, 101
lowest Adabas buffer efficiency before activation, 75
number of storage areas allocated, 95
parameter to activate RABN ranges, 65
parameter to control data space RABN caching, 72
parameter to control data space RABN caching for virtual 64 backed by 1M large pages, 72
parameter to control data space RABN caching for virtual 64 backed by 2G large pages, 72
parameter to control extended memory RABN caching, 72

parameter to control hiperspace RABN caching, 72
parameter to control RABNs cached in data space, 66
parameter to control RABNs cached in extended memory, 66
parameter to control RABNs cached in hiperspace, 66
parameter to control RABNs cached in virtual 64 storage, 66
parameter to control RABNs cached in virtual 64 storage backed by 1M large pages, 66
parameter to control RABNs cached in virtual 64 storage backed by 2G large pages, 66
parameter to control virtual 64 storage RABN caching for virtual 64, 72
parameter to enable, 65, 76, 79
parameter to exclude command types, 78
parameter to override Associator and Data Storage caching parameters, 97
parameter to set the cache space area or RABN inactivity time, 70
read-ahead buffer count parameter, 69
time between retry attempts, 96
work area cache type, 98
Work part 2 percentage cached, 99
Work part 3 percentage cached, 100
CACTIVATE
ADARUN parameter, 65
CACTIVATE parameter, 23
CANCEL
operator command, 259
CASSODSP
ADARUN parameter, 66
CASSODSP parameter, 23
CASSOEXT
ADARUN parameter, 66
CASSOEXT parameter, 24
CASSOG64
ADARUN parameter, 66
CASSOG64 parameter, 24
CASSOHSP
ADARUN parameter, 66
CASSOHSP parameter, 24
CASSOL64
ADARUN parameter, 66
CASSOL64 parameter, 24
CASSOMAXS
ADARUN parameter, 68
CASSOMAXS parameter, 24
CASSOP64 parameter, 24
CASSOV64
ADARUN parameter, 66
CASSOV64 parameter, 24
CBUFNO
ADARUN parameter, 69
CBUFNO parameter, 24
CCTIMEOUT
ADARUN parameter, 70
CCTIMEOUT parameter, 24
CDATADSP
ADARUN parameter, 72
CDATADSP parameter, 24
CDATAEXT
ADARUN parameter, 72
CDATAEXT parameter, 25
CDATAG64
ADARUN parameter, 72
CDATAG64 parameter, 25
CDATAHSP
ADARUN parameter, 72
CDATAHSP parameter, 25
CDATAL64
ADARUN parameter, 72
CDATAL64 parameter, 25
CDATAMAXS
ADARUN parameter, 74
CDATAMAXS parameter, 25
CDATAP64 parameter, 25
CDATAV64
ADARUN parameter, 72
CDATAV64 parameter, 25
CDEMAND
ADARUN parameter, 75
CDEMAND parameter, 26
CDISPSTAT parameter, 26
CDXnn
ADARUN parameter, 77
CDXnn parameter, 26
CEXCLUDE
ADARUN parameter, 78
CEXCLUDE parameter, 26
CFILE
ADARUN parameter, 79
CFILE parameter, 26
checkpoints
SYNS 60
parameter to set time interval between, 121
choices in syntax, 9
CICS
Adabas call using, 360
code reentrancy requirements, 355
issuing direct calls to Adabas through, 360
LNCSTUB link routine for command level, 352
CLOG
alert messages, 291
CLOGBMAX
ADARUN parameter, 81
CLOGBMAX parameter, 27
CLOGDEV
ADARUN parameter, 81
CLOGDEV parameter, 27
CLOGLAYOUT
ADARUN parameter, 82
CLOGLAYOUT parameter, 27
CLOGMAX
ADARUN parameter, 83
CLOGMAX parameter, 27
CLOGMRG
ADARUN parameter, 83
operator command, 259
CLOGMRG parameter, 27
CLOGSIZE
ADARUN parameter, 81
CLOGSIZE parameter, 27
CLUCACHEEXTRA
ADARUN parameter, 85
CLUCACHEEXTRA parameter, 27
CLUCACHENAME
ADARUN parameter, 85
CLUCACHENAME parameter, 27

CLUCACHESIZE
ADARUN parameter, 86
CLUCACHESIZE parameter, 27
CLUCACHETYPE
ADARUN parameter, 87
CLUCACHETYPE parameter, 27
CLUCACHEUNCHANGED
ADARUN parameter, 89
CLUCACHEUNCHANGED parameter, 28
CLUGROUPNAME
ADARUN parameter, 90
CLUGROUPNAME parameter, 28
CLULOCKNAME
ADARUN parameter, 91
CLULOCKNAME parameter, 28
CLULOCKSIZE
ADARUN parameter, 92
CLULOCKSIZE parameter, 28
CLUPUBLPROT
ADARUN parameter, 92
CLUPUBLPROT parameter, 28
CLUSTER
ADARUN parameter, 93
cluster
cache vector size, 85
group name, 90
redo updates
parameter to set size of pool, 135
CLUSTER parameter, 28
CLUWORK1CACHE
ADARUN parameter, 94
CLUWORK1CACHE parameter, 29
CMAXCSPS
ADARUN parameter, 95
CMAXCSPS parameter, 29
COBOL
set the control byte, 357
collation descriptor
user exit
parameter to specify ID and name, 77
Com-plete
ADALCO link routine, 352
ADALCO8 link routine, 352
API work area, 353
issuing direct calls to Adabas through, 360
reentrancy requirements, 355
command ID
count per user
parameter to set maximum, 163
requirement for OPEN with active user profile, 335
Command log
command to start, 289
command to stop, 297
dual
parameter to specify device type, 110
parameter to specify size, 110
multiple
parameter to specify device type, 81
parameter to specify size, 81
parameter to enable merge, 83
parameter to enable/disable, 131
parameter to specify format of, 82
sequential
parameter to set block size, 132
command log
dual
switch to other, 285
extension
enable/disable logging of, 131
command log (CLOG)
alert messages, 133
Command log data sets
maximum number of
parameter to specify, 158
Command log merge
command to switch on or off, 259
command queue
command to display usage, 274
returning unprocessed commands from, 263
setting command queue maximum wait time during online recovery, 57
specifying environment to return commands from, 56
command queue element
maximum number of
parameter to specify, 157
parameter to set time limit for hold, 98
command queue element (CQE)
display posted, 269
command queue element (DCQ)
display all posted
using operator command, 263
command routing, 313
commands
display total processed for a file, 266
DONLSTAT, 271
DRPLSTAT, 275
DSPACE, 276
DSTAT, 276
exclude from prefetch/multifetch
using parameter, 179
include in prefetch/multifetch
using parameter, 172
log update to DDPRINT
using console, 251
ONLRESUME, 300
ONLSTOP, 300
ONLSUSPEND, 301
operator, 243
parameter to
set time limit for completion, 98
RPLCHECK, 303
RPLCONNECT, 303
RPLCONNECTCOUNT, 304
RPLCONNECTINTERVAL, 304
communication
cluster nuclei
timeout threshold for canceled peer nucleus, 146
timeout threshold for command, 148
connection attempts
dynamically specifying the interval between, 304
dynamically specifying the number of, 304
forcing an attempt, 303
console messages
multi-line WTO messages, 143
Control scheduling of update commands during buffer flush, 220
conventions
syntax, 8

CRETRY
ADARUN parameter, 96
CRETRY parameter, 29
cross-checking active Event Replicator Servers, 303
CSTORAGE
ADARUN parameter, 97
CSTORAGE parameter, 29
CT
ADARUN parameter, 98
operator command, 261
CT parameter, 29
CWORK2FAC
ADARUN parameter, 99
CWORK2FAC parameter, 30
CWORK3FAC
ADARUN parameter, 100
CWORK3FAC parameter, 30
CWORKSTORAGE
ADARUN parameter, 98
CWORKSTORAGE parameter, 29
CXFILE
ADARUN parameter, 101
CXFILE parameter, 30

D

Data Integrity Block (DIB)
parameter to
ignore active nucleus entry in, 116
data integrity block (DIB)
operator command to display, 263
data protection area
parameter to allocate blocks for, 134
data sets
required for z/OS, 328
Data Storage
automatic increase size command, 256
automatic increase threshold setting, 257
setting maximum size, 258
space notifications, 261
Data Storage (DATA)
determining space defined, 276
Data Storage data set
parameter to enable/disable controller caching for, 102
Data Storage ociator data set
space notifications, 103
data storage record
index cross-checking, 118
index updating, 119
database
re-creating after failure, 377
repair after failure, 376
restarting after failure, 377
DATACACHE
ADARUN parameter, 102
DATACACHE parameter, 30
DATASPACEWARN
ADARUN parameter, 103
operator command, 261
DATASPACEWARN parameter, 30
DAUQ
operator command, 262
DBID
ADARUN parameter, 105
DBID parameter, 30
DCMDSTAT
operator command, 262
DCQ
operator command, 263
DDIB
operator command, 263
DDSF
operator command, 263
default parameter values, 8
DELCQ
operator command
using console, 263
DELUF
operator command, 264
DELUI
operator command, 265
descriptor
collation
parameter to specify user exit for, 77
unique
parameter to set size of pool, 125
detail sections
SMF records, 305
determining ASSO and DATA defined space, 276
DEVICE
ADARUN parameter, 105
DEVICE parameter, 30
DFILES
operator command, 265
DFILESTAT
operator command, 266
DFILUSE
operator command, 266
DHQ
operator command, 266
DHQA
operator command, 266
direct calls
issuing through IMS TM, 364
using API, 359
CICS, 360
CICS command level examples, 362-363
CICS macro level example, 361
Com-plete examples, 360
IMS TM, 364
IMS TM nonreentrant examples, 364
DIRRATIO / ELEMENTRATIO
ADARUN parameter, 105
DIRRATIO parameter, 31
displaying Adabas session parameters, 271
displaying initial-state request status, 271
displaying replay-related statistics, 276
displaying replication-related statistics, 275
distributed transaction processing
parameter to
ignore data in Work part 4, 117
parameter to enable/disable, 108
DLOCKF
operator command, 267
DMEMTB command
display member state table, 268
DNC
operator command, 269

DNFV command
displaying nucleus file variables, 269
DNH
operator command, 270
DNU
operator command, 270
DONLSTAT
operator command, 271
DONLSTAT operator command, 271
DPARM operator command, 271
DPPT command
displaying a nucleus PPT block, 273
DRES operator command
using utility, 274
DRPLSTAT operator command, 275
DSF
ADARUN parameter, 106
DSF parameter, 31
DSFEX1
ADARUN parameter, 107
DSFEX1 parameter, 31
DSPACE operator command
operator command, 276
DSTAT
operator command, 276
utilities, 317
DSTAT operator command, 276
DTARG command
display system image, 277
DTH
operator command, 277
DTP
ADARUN parameter, 108
DTP parameter, 31
DUALALD
ADARUN parameter, 109
DUALALD parameter, 31
DUALALS
ADARUN parameter, 109
DUALALS parameter, 31
DUALCLD
ADARUN parameter, 110
DUALCLD parameter, 31
DUALCLS
ADARUN parameter, 110
DUALCLS parameter, 31
DUALPLD
ADARUN parameter, 111
DUALPLD parameter, 31
DUALPLS
ADARUN parameter, 111
DUALPLS parameter, 31
dump
command to check status, 302
DUQ
operator command, 277
DUQA
operator command, 278
DUQE
operator command, 278
DUUQE
operator command, 278
DVOLIO
operator command, 279
DXCACHE command
display lock statistics, 280
DXFILE command
display file statistics, 281
DXLOCK command
display lock statistics, 282
DXSTAT command
display cache and lock statistics, 283
dynamically change MXCANCEL setting, 292
dynamically change MXCANCELWARN setting, 293
dynamically change MXMSGWARN setting, 295
dynamically change MXWTOR setting, 297
DZSTAT
operator command, 283

E

ELEMENTRATIO parameter, 31
ellipsis (...), 9
ENCRYPTION
ADARUN parameter, 112
ENCRYPTION parameter, 32
Entire Net-Work target connection
stop logging, 300
error handling and message buffering facility
parameter to activate, 203
ETID
use with user profile table, 335
exclusive control
obtaining for a file, 248
releasing for a file, 301
releasing for all files, 301
EXCPVR
controlling page fixing, 169
EXCPVR parameter, 32
extended Adabas control block (ACBX)
enable/disable logging of, 131
extended I/O list
enable/disable logging of, 132
logging, 291
stop logging, 299

F

failure
types of, 375
FASTPATH
ADARUN parameter, 112-113
FASTPATH parameter, 32
FEOFAL
operator command, 285
FEOFCL
operator command, 285
FEOFPL
operator command, 286
file
statistics display, 281
files
exclude from prefetch/multifetch
using parameter, 179
include in prefetch/multifetch
using parameter, 172
lock
using operator command, 288

locked
display using operator command, 267
obtaining an advance-lock, 248
parameter to exclude from
autorestart processing, 51
re-creating after failure, 378
releasing advance-lock, 301
stop users of
using operator command, 310
unlock
using operator command, 316
FMXIO
ADARUN parameter, 113
FMXIO parameter, 32
FMXIO=nn
operator command, 286
FORCE
ADARUN parameter, 114
FORCE parameter, 32
forcing a connection attempt, 303
forcing connection
force connection attempt command, 253
format buffer
enable/disable logging of, 131
logging
command to start, 289
command to stop, 298
format pool
command to display usage, 274

G

global cache area
construct type, 87
name, 85
ratio
directory entries to elements, 105
size, 86
Global lock area
statistics display, 282
global lock area
name, 91
size, 92

H

HALT
operator command, 286
HEXnn
ADARUN parameter, 115
HEXnn parameter, 32
hold queue
command to display usage, 274
count of ISNs in
parameter to set maximum, 160
hold queue element
display, 266
maximum number of
parameter to specify, 159
hyperdescriptor
parameter to
specify user exit ID and name, 115

I

I/O activity
enable/disable logging of, 131
logging
command to start, 290
command to stop, 298
ID Table
parameter to
allow nucleus to overwrite existing entry, 114
IGNDIB
ADARUN parameter, 116
IGNDTP
ADARUN parameter, 117
IGNDTP parameter, 32
IMS
Adabas call structure for use with, 364
API work area, 353
IMS TM
ADALNI link routine, 352
ADALNI8 link routine, 352
indentation, 9
index cross-checking controls, 118, 287
index updating controls, 119
INDEXCROSSCHECK
ADARUN parameter, 118
operator command, 287
INDEXCROSSCHECK parameter, 32
INDEXUPDATE
ADARUN parameter, 119
INDEXUPDATE parameter, 33
INFOBUFFERSIZE
ADARUN parameter, 119
INFOBUFFERSIZE parameter, 33
information buffer pool
see Adabas Event Log, 119
initial-state requests
displaying status, 271
resuming, 300
stopping, 300
suspending, 301
INTAUTO
ADARUN parameter, 120
INTAUTO parameter, 33
INTNAS
ADARUN parameter, 121
INTNAS parameter, 33
ISN
count in hold queue
operator command to display, 270
parameter to set maximum, 160
count per TBI element
parameter to set maximum, 164
list processing area
space allocation using parameter, 141
ISN buffer
enable/disable logging of, 131
logging
command to start, 290
command to stop, 298
italic, 8

L

LAP
    ADARUN parameter, 121
LAP parameter, 33
large page support, 122
LARGEPAGE
    ADARUN parameter, 122
LARGEPAGE parameter, 33
LBP
    ADARUN parameter, 86, 123
LBP parameter, 33
LCP
    ADARUN parameter, 125
LCP parameter, 33
LDEUQP
    ADARUN parameter, 125
LDEUQP parameter, 33
LDTP
    ADARUN parameter, 126
LDTP parameter, 34
LFIOP
    ADARUN parameter, 127
LFIOP parameter, 34
LFP
    ADARUN parameter, 128
LFP parameter, 34
LI
    ADARUN parameter, 129
LI parameter, 34
LICREFRESH
    operator command, 288
link globals table
    name, 130
link routines
    corresponding to TP monitors, 352
LNCSTUB
    link routine for CICS (command-level), 352
LNKGNAME
    ADARUN parameter, 130
LNKGNAME parameter, 34
LOCAL
    ADARUN parameter, 130
local buffer pool
    size parameter, 86
LOCAL parameter, 34
lock structure
    name, 91
    statistics display, 283
LOCKF
    operator command, 288
locking files
    in advance, 248
LOCKU
    operator command, 288
LOCKX
    operator command, 289
LOGABDX
    ADARUN parameter, 131
LOGABDX parameter, 34
LOGCB
    ADARUN parameter, 131
    operator command, 289
LOGCB parameter, 34
LOGCLEX
    ADARUN parameter, 131
LOGCLEX parameter, 34
LOGFB
    ADARUN parameter, 131
    operator command, 289
LOGFB parameter, 35
logged buffer size limit
    ADARUN parameter, 81
LOGGING
    ADARUN parameter, 131
    operator command, 289
LOGGING parameter, 35
LOGIB
    ADARUN parameter, 131
    operator command, 290
LOGIB parameter, 35
LOGIO
    ADARUN parameter, 131
    operator command, 290
LOGIO parameter, 35
LOGMB
    ADARUN parameter, 131
LOGMB parameter, 35
LOGRB
    ADARUN parameter, 132
    operator command, 290
LOGRB parameter, 35
LOGSB
    ADARUN parameter, 132
    operator command, 290
LOGSB parameter, 35
LOGSIZE
    ADARUN parameter, 132
LOGSIZE parameter, 35
LOGUX
    ADARUN parameter, 132
    operator command, 291
LOGUX parameter, 35
LOGVB
    ADARUN parameter, 132
    operator command, 291
LOGVB parameter, 35
LOGVOLIO
    ADARUN parameter, 132
    operator command, 291
LOGVOLIO parameter, 35
LOGWARN
    ADARUN parameter, 133
    operator command, 291
LOGWARN parameter, 35
lowercase, 8
LP
    ADARUN parameter, 134
LP parameter, 36
LQ
    ADARUN parameter, 135
LQ parameter, 36
LRDP
    ADARUN parameter, 135
LRDP parameter, 36
LRP1 parameter, 36
LRPL parameter, 137
LS

ADARUN parameter, 138
LS parameter, 36
LSF
ADARUN parameter, 139
LSF parameter, 37
LTZ parameter, 37, 139
LU
ADARUN parameter, 140
LU parameter, 37
LWKP2
ADARUN parameter, 141
LWKP2 parameter, 37
LWP
ADARUN parameter, 142
LWP parameter, 37

M

member state table
command to display, 268
message buffer area
parameter to set size of, 144
message case
parameter to set, 145
minimum keywords, 8
mixed case, 8
MLWTO parameter, 37, 143
MODE
ADARUN parameter, 143
MODE parameter, 37
modes of operation
parameter to specify, 143
MSGBUF
ADARUN parameter, 144
MSGBUF parameter, 37
MSGCONSL
ADARUN parameter, 145
MSGCONSL parameter, 37
MSGDRUCK
ADARUN parameter, 145
MSGDRUCK parameter, 37
MSGPRINT
ADARUN parameter, 145
MSGPRINT parameter, 37
multi-line console messages, 143
multifetch
automatic activation
parameter to specify, 171
buffer length per call
parameter to set, 174
count of records to obtain
parameter to set maximum, 173
data set for statistics, 175
exclude commands from
using parameter, 179
exclude files from
using parameter, 179
include commands in
using parameter, 172
include files in
using parameter, 172
total buffer length
parameter to set, 178
multifetch buffer
enable/disable logging of, 131
multifetch statistics
data set for, 175
multiprocessing
nucleus ID for
parameter to specify, 167
multiuser
operating mode
parameter to specify, 143
MXCANCEL
ADARUN parameter, 146
MXCANCEL operator command, 292
MXCANCEL parameter, 38
MXCANCELWARN
ADARUN parameter, 147
MXCANCELWARN operator command, 293
MXCANCELWARN parameter, 38
MXMSG
ADARUN parameter, 148
MXMSG parameter, 38
MXMSGWARN
ADARUN parameter, 149
MXMSGWARN operator command, 295
MXMSGWARN parameter, 38
MXSTATUS
ADARUN parameter, 150
MXSTATUS parameter, 38
MXTNA
ADARUN parameter, 151
MXTNA parameter, 38
MXTSX
ADARUN parameter, 152
MXTSX parameter, 38
MXTT
ADARUN parameter, 153
MXTT parameter, 39
MXWTOR
ADARUN parameter, 154
MXWTOR operator command, 297
MXWTOR parameter, 39

N

NAB
ADARUN parameter, 155
NAB parameter, 39
NALOG
ADARUN parameter, 156
NALOG parameter, 39
Natural
applications written in, 353
NC
ADARUN parameter, 157
NC parameter, 39
NCLOG
ADARUN parameter, 158
NCLOG parameter, 39
NH
ADARUN parameter, 159
NH parameter, 39
NISNHQ
ADARUN parameter, 160
NISNHQ parameter, 39
NOLOGCB

operator command, 297
NOLOGFB
operator command, 298
NOLOGGING
operator command, 297
NOLOGIB
operator command, 298
NOLOGIO
operator command, 298
NOLOGRB
operator command, 298
NOLOGSB
operator command, 299
NOLOGUX
operator command, 299
NOLOGVB
operator command, 299
NOLOGVOLIO
operator command, 299
NONDES
ADARUN parameter, 161
NONDES parameter, 39
normal font, 8
NPLOG
ADARUN parameter, 161
NPLOG parameter, 39
NPLOGBUFFERS parameter, 39, 162
NQCID
ADARUN parameter, 163
NQCID parameter, 40
NSISN
ADARUN parameter, 164
NSISN parameter, 40
NT
ADARUN parameter, 165
NT parameter, 40
NU
ADARUN parameter, 166
NU parameter, 40
NUCID
ADARUN parameter, 167
NUCID parameter, 40
Nucleus
TCP/IP URL
parameter to set, 207
nucleus
command to display command distribution statistics, 262
command to display current status, 276
command to display zIIP status, 283
control TCP/IP access using TCPIP parameter
using parameter, 206
ID for multiprocessing
parameter to specify, 167
isolated
parameter to define as a local nucleus, 130
number of threads per
parameter to set maximum, 165
operator commands, 245
overview in Adabas operation, 11
SVC for
parameter to specify, 205
TCP/IP link to
command to open/close, 314
nucleus file variables
displaying, 269
NWCONNECT
operator command, 300
NWORK1BUFFERS parameter, 40, 168

O

online operation
process a request to Adabas, 350
online process
display status of, 271
resume a suspended process, 300
stop cleanly, 300
suspend, 301
online recovery
setting environment for commands reviewed in the command queue, 56
setting wait time for commands in the command queue, 57
ONLRESUME
operator command, 300
ONLRESUME operator command, 300
ONLSTOP
operator command, 300
ONLSTOP operator command, 300
ONLSUSPEND
operator command, 301
ONLSUSPEND operator command, 301
open command
requirement for
parameter to specify, 168
OPENRQ
ADARUN parameter, 168
OPENRQ parameter, 40
operations
overview of Adabas, 11
Operator commands
DXLOCK
display lock statistics, 282
operator commands
connect to Adabas server or database ID, 253
controlling ASSO space use notifications, 252
controlling DATA space use notifications, 261
DMEMTB
display member state table, 268
DNFV, 269
DPARM, 271
DPPT, 273
DTARG, 277
DXCACHE
display lock statistics, 280
DXFILE
display file statistics, 281
DXSTAT
display cache and lock statistics, 283
MXCANCEL, 292
MXCANCELWARN, 293
MXMSGWARN, 295
MXWTOR, 297
setting automatic increase size for ASSO, 253
setting automatic increase size for DATA, 256
setting automatic increase threshold for ASSO, 255
setting automatic increase threshold for DATA, 257
setting maximum size for ASSO, 255
setting maximum size for DATA, 258

TARGET, 313
optional syntax elements, 9

P

page fixing, controlling
EXCPVR runs, 169
parameters, 49
PGFIX
ADARUN parameter, 169
PGFIX parameter, 40
PL/I
compiler code reentrancy, 355
PLOG
alert messages, 291
PLOG protection I/O buffers, 162
PLOGDEV
ADARUN parameter, 169
PLOGDEV parameter, 40
PLOGRQ
ADARUN parameter, 170
PLOGRQ parameter, 40
PLOGSIZE
ADARUN parameter, 169
PLOGSIZE parameter, 40
PPT
displaying a nucleus block, 273
PREFETCH
ADARUN parameter, 171
prefetch
automatic activation
parameter to set, 171
buffer length per call
parameter to set, 174
count of records to obtain
parameter to set maximum, 173
exclude commands from
using parameter, 179
exclude files from
using parameter, 179
include commands in
using parameter, 172
include files in
using parameter, 172
total buffer length
parameter to set, 178
PREFETCH parameter, 41
PREFICMD
ADARUN parameter, 172
PREFICMD parameter, 41
PREFIFIL
ADARUN parameter, 172
PREFIFIL parameter, 41
PREFNREC
ADARUN parameter, 173
PREFNREC parameter, 42
PREFSBL
ADARUN parameter, 174
PREFSTDD
ADARUN parameter, 175
PREFTBL
ADARUN parameter, 178
PREFTBL parameter, 42
PREFXCMD
ADARUN parameter, 179
PREFXCMD parameter, 42
PREFXFIL
ADARUN parameter, 179
PREFXFIL parameter, 42
print
statistics after refresh, 184
profile table
for user-related information, 335
maintenance, 336
values, 335
PROGRAM
ADARUN parameter, 180
PROGRAM parameter, 43
Protection log
description of, 373
dual
description of, 374
parameter to specify device type, 111
parameter to specify size, 111
switch to other, 374
multiple
parameter to specify device type, 169
parameter to specify size, 169
requirement for
parameter to specify, 170
sequential
description of, 373
protection log
dual
switch to other, 286
protection log (PLOG)
alert messages, 133
Protection log data sets
maximum number of
parameter to specify, 161
punctuation and symbols in syntax, 10

Q

QBLKSIZE
ADARUN parameter, 182
quasi-reentrant code
definition, 355

R

RALOCKF
operator command, 301
RALOCKFA
operator command, 301
RDUMPST
operator command, 302
read-only status
command to switch on or off, 302
parameter to set for session, 183
READONLY
ADARUN parameter, 183
operator command, 302
READONLY parameter, 43
record buffer
enable/disable logging of, 132
logging
command to start, 290

command to stop, 298
recovery, 371
redo pool
command to display usage, 274
parameter to set size of, 135
reentrant code
definition, 355
REFSTPRT
ADARUN parameter, 184
operator command, 302
REFSTPRT parameter, 43
region
address space as a, 12
release
advance-lock on a file, 301
advance-lock on all file, 301
repeated syntax elements, 9
REPLICATION
REPLICATION parameter, 184
REPLICATION parameter, 43
required syntax elements, 9
resources
display current usage
using utility, 274
statistics
command to display, 274
response code 145 (ADARSP145)
logging events, 119
restart, 371
resuming a suspended initial-state request, 300
returning commands from the command queue, 56-57
REVBUFMB parameter, 43, 185
REVFILTER
ADARUN parameter, 186
REVFILTER parameter, 43
REVIEW
ADARUN parameter, 186
operator command, 302
REVIEW parameter, 43
REVLOGBMAX parameter, 43, 188
REVLOGMAX parameter, 44, 188
REVLOGNR parameter, 44, 189
REVSTHMV parameter, 44, 190
REVTIMER parameter, 44, 190
RIAFTERUPDATE parameter, 44, 191
RPLCHECK operator command, 303
RPLCONNECT operator command, 303
RPLCONNECTCOUNT operator command, 304
RPLCONNECTCOUNT parameter, 44, 192
RPLCONNECTINTERVAL operator command, 304
RPLCONNECTINTERVAL parameter, 44, 192
RPLPARMS parameter, 44, 192
RPLSORT parameter, 193
RPWARNINCREMENT parameter, 45, 194
RPWARNINTERNVAL parameter, 45
RPWARNINTERVAL parameter, 195
RPWARNMESSAGELIMIT parameter, 45, 195
RPWARNPERCENT parameter, 45, 196
RVCLIENT parameter, 45, 196

S

search buffer
enable/disable logging of, 132
logging
command to start, 290
command to stop, 299
searches
complex S1/S2/S4
parameter to set time limit for, 209
non-descriptor fields
parameter to allow/disallow, 161
SECUID parameter, 45, 197
security pool
Adabas
parameter to set maximum size, 125
command to display usage, 274
security system user ID
requirement level, 197
SECUID ADARUN parameter, 197
security user ID
commands to stop and delete user, 310-311
sequential data sets
block size
parameter to specify, 182
session
Adabas
definition of, 371
displaying current parameters, 271
read-only status
parameter to specify, 183
sample z/OS job control for TCP/IP, 330
sample z/OS job control for UES, 330
sample z/OS job control to execute, 329
SVC for
parameter to specify, 205
unique identifier, 378
user
definition of, 371
utilities execution only
command to specify, 316
parameter to specify, 221
shared cache
unchanged block option, 89
single user
operating mode
parameter to specify, 143
requirements and restrictions, 335
SMF
ADARUN parameter, 198
SMF Interval records, 305
SMF parameter, 46
SMF records
detail sections, 305
enabling interval records, 201
enabling logging of, 198
enabling type 89, 198
selecting detail sections, 199
setting user-defined record number, 202
SMF user exit module name, 220
specifying subsystem name, 202
SMF Termination records, 305
SMF89
ADARUN parameter, 198
SMF89 parameter, 46
SMFDETAIL
ADARUN parameter, 199
SMFDETAIL parameter, 46

SMFINTERVAL
ADARUN parameter, 201
SMFINTERVAL parameter, 46
SMFRECNO
ADARUN parameter, 202
SMFRECNO parameter, 46
SMFSUBSYS
ADARUN parameter, 202
SMFSUBSYS parameter, 46
SMGT
ABNORMALTERM operand, 319
ACTPIN operand, 320
ADARUN parameter, 203
ADDPIN operand, 320
DEACTPIN operand, 320
DELPIN operand, 320
DISPLAY operand, 320
sample output, 323
DUMP operand, 321
MSGBUF operand, 319
operator command, 318, 321
SNAP operand, 321
XACTIVATE operand, 322
XCRITICAL operand, 322
XDEACTIVATE operand, 322
XLOAD operand, 322
XNOTCRITICAL operand, 322
SMGT parameter, 46
Sort
set size of internal processing area, 138-139
Sort data set
enable/disable controller caching for data set, 203
SORTCACHE
ADARUN parameter, 203
SORTCACHE parameter, 46
space notifications
Associator, 54, 252
Data Storage, 103, 261
spanned records
logging, 205
specifying connection attempt interval, 304
specifying number of connection attempts, 304
SPT
ADARUN parameter, 204
SPT parameter, 46
SRLOG
ADARUN parameter, 205
SRLOG parameter, 46
statistics
command to switch printing on or off, 302
control to print after refresh, 184
displaying, 275-276, 283
STOPF
operator command, 310
STOPI
operator command, 310
stopping an initial-state request, 300
STOPSU
operator command, 310
STOPSUR
operator command, 311
STOPU
operator command, 311
STOPUR
operator command, 312
subparameter syntax, 9
suspending an initial-state request, 301
SVC
ADARUN parameter, 205
SVC dump facility, 334
SVC parameter, 46
SYNCC
operator command, 312
syntax conventions, 8
bold, 8
braces ({}), 9
brackets ([]), 9
defaults, 8
ellipsis (...), 9
indentation, 9
italic, 8
lowercase, 8
minimum keywords, 8
mixed case, 8
mutually exclusive choices, 9
normal font, 8
optional elements, 9
punctuation and symbols, 10
repeated elements, 9
required elements, 9
subparameters, 9
underlining, 8
uppercase, 8
vertical bars (|), 9
system image display, 277

T

table of ISNs
number of ISNs per element
parameter to set maximum, 164
parameter to set size of, 129
table of ISNs pool
command to display usage, 274
table of sequential commands
parameter to set size of, 135
table of sequential commands pool
command to display usage, 274
TARGET command
route commands to specified target, 313
TARGETID
ADARUN parameter, 206
TARGETID parameter, 46
TCP/IP
link to Adabas nucleus
command to open/close, 314
session execution JCL, 330
TCPIP
ADARUN parameter, 206
issuing Entire Net-Work operator commands, 314
operator command, 314
TCPIP parameter, 47
TCPURL
ADARUN parameter, 207
TCPURL parameter, 47
Temp data set
controller caching for data set
parameter to enable/disable, 208

TEMPCACHE
ADARUN parameter, 208
TEMPCACHE parameter, 47
TFLUSH
ADARUN parameter, 209
TFLUSH parameter, 47
threads
number allowed per nucleus
parameter to set maximum, 165
status
command to display, 277
time zone pool
setting size, 139
time zones
setting time zone pool size, 139
timeout control
interregion communication limit
parameter to set, 98
limit for complex searches
parameter to set, 209
non-activity limit
command to set, 310
for access-only users parameter to set, 216
for ET logic users parameter to set, 217
for exclusive control users parameter to set, 217
parameter to set maximum time, 151
Sx command execution
parameter to set maximum time, 152
transaction limit
parameter to set for ET logic users, 218
parameter to set maximum time, 153
timeout facility, 385
action taken when time limit exceeded, 387
parameters, 386
TLSCMD
ADARUN parameter, 209
TLSCMD parameter, 47
TMDRQ
ADARUN parameter, 210
TMDRQ parameter, 47
TMETDATA
ADARUN parameter, 211
TMETDATA parameter, 47
TMGTT
ADARUN parameter, 212
TMGTT parameter, 47
TMLOG
ADARUN parameter, 213
TMLOG parameter, 47
TMMSGSEV
ADARUN parameter, 214
TMMSGSEV parameter, 47
TMRESTART
ADARUN parameter, 214
TMRESTART parameter, 47
TMSYNCMGR
ADARUN parameter, 215
TMSYNCMGR parameter, 47
TMTCIDPREF
ADARUN parameter, 215
TMTCIDPREF parameter, 48
TNAA
ADARUN parameter, 216
operator command, 314
TNAA parameter, 48
TNAE
ADARUN parameter, 217
operator command, 315
TNAE parameter, 48
TNAX
ADARUN parameter, 217
operator command, 315
TNAX parameter, 48
TP monitor
link routine corresponding to, 352
transaction
time limit for ET logic users
parameter to set, 218
transaction ID (XID) pool
command to display usage, 274
TSO
ADALNK link routine, 352
ADALNK8 link routine, 352
ADALNKR link routine, 352
ADALNKR8 link routine, 352
API work area, 353
TT
ADARUN parameter, 218
operator command, 315
TT parameter, 48
type 89 SMF records, 198
TZINFO data set, 334

U

UES
session execution JCL, 330
UEXn
ADARUN parameter, 219
UEXnn parameter, 48
UEXSMF
ADARUN parameter, 220
UEXSMF parameter, 48
underlining, 8
unique descriptor pool
command to display usage, 274
parameter to set size of, 125
UNLOCKF
operator command, 316
UNLOCKU
operator command, 316
UNLOCKX
operator command, 316
UPDATECONTROL
ADARUN parameter, 220
UPDATECONTROL parameter, 48
uppercase, 8
user
access-only
non-activity time limit parameter to set, 216
set non-activity time limit using operator command, 314
batch
how Adabas identifies, 372
commands to stop and delete, 311-312
definition of, 371
ET logic
non-activity time limit parameter to set, 217
set non-activity time limit using operator command, 315

set transaction time limit using operator command, 315
synchronize all using operator command, 312
transaction time limit parameter to set, 218
exclusive control
non-activity time limit parameter to set, 217
set non-activity time limit using operator command, 315
non-exclusive control
lock file for, using operator command, 289
unlock file for using operator command, 316
non-utility
lock file for, using operator command, 288
unlock file for using operator command, 316
number of command IDs per
parameter to set maximum, 163
number of ISNs in hold queue per
parameter to specify, 160
profile, 335
management of, 335
program
relationship to Adabas operation, 12
session
definition of, 371
User buffer
intermediate
parameter to set size of, 140
user exits
B
enable/disable logging of, 132
logging command to start, 291
logging command to stop, 299
collation descriptor
parameter to specify ID and name, 77
hyperdescriptor
parameter to specify ID and name, 115
link applications to Adabas, 351
parameter to specify ID and name, 219
SMF user exit module name, 220
user queue
command to display usage, 274
user queue element
display for recent user
using operator command, 262
display for specified user
using operator command, 278
display up to 100
using operator command, 278
display up to 5
using operator command, 277
display utility
using operator command, 278
maximum number of
parameter to set, 166
user queue element (UQE)
delete for a file
using operator command, 264
delete inactive
using operator command, 265
user queue file list pool
command to display usage, 274
users
count of current
operator command to display, 270
utilities
condition and termination codes, 340
data sets accessed on disk, 344
display status, 317
exclusive user of files required
conditions for termination, 346
executing, 339
file and database usage, 341
overview in Adabas operation, 12
session for utility execution only
parameter to specify, 221
utility control statement
parameter values
default, 8
UTIONLY
ADARUN parameter, 221
operator command, 316
UTIONLY parameter, 48

V

V64BIT parameter, 48, 222
value buffer
enable/disable logging of, 132
logging
command to start, 291
command to stop, 299
vertical bars (|), 9
virtual machines
term used to define storage space, 12
virtual storage
64-bit option, 222
VISTA
ADARUN parameter, 223
VISTA parameter, 48

W

Work
part 1
space allocation using parameter, 134
part 2
space allocation using parameter, 141
part 4
parameter to ignore data in, 117
work
part 1
function of, 372
Work data set
controller caching
parameter to enable/disable, 223
WORK I/O buffers
controlling in autorestart, 52
Work part 1
command to display usage, 274
protection I/O buffer controls, 168
Work part 2
command to display usage, 274
Work part 3
command to display usage, 274
Work pool
Adabas
parameter to set size of, 142
intermediate user buffer area
parameter to set size of, 140
sort processing area

parameter to set size of, 138-139
work pool
command to display usage, 274
WORKCACHE
ADARUN parameter, 223
WORKCACHE parameter, 48

X

XCF group
name, 90
XID pool
command to display usage, 274

Z

z/OS
Adabas data sets required for, 328
TCP/IP URL
parameter to set, 207
ZIIP
ADARUN parameter, 224
operator command, 317
parameter to enable/disable, 224
ZIIP parameter, 49