

**Adabas**

**Operations**

Version 8.1.4

June 2014

This document applies to Adabas Version 8.1.4.

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

Copyright © 1971-2014 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors..

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. 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 AG and/or its subsidiaries is located at <http://documentation.softwareag.com/legal/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> 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 and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

**Document ID: ADAMF-OPERATOR-814-20140626**

## Table of Contents

Operations .....	ix
1 Conventions .....	1
Syntax Conventions .....	2
Data Set Names .....	4
2 Controlling the Adabas Environment .....	5
I .....	7
3 Software AG Mainframe Product Licensing .....	9
Distributed License Libraries .....	10
License File .....	11
Transferring a License File from PC to a z/OS Host Using FTP .....	12
Transferring a License File from PC to a z/VSE Host Using FTP .....	13
Transferring a License File to a z/VM Host .....	15
Transferring a License File from PC to a BS2000 Host Using FTP .....	16
License File Installation .....	17
Product License Check FAQs .....	17
Using The License Utility: LICUTIL .....	22
II Adabas Initialization (ADARUN Statement) .....	31
4 ADARUN Parameter Syntax .....	33
5 ADARUN Parameter Directory .....	35
AOSLOG : Log Active Nucleus Update Commands .....	62
AREXCLUDE : Exclude Files from Autorestart Processing .....	62
ARMNAME : Activate Automatic Restart Management (ARM) Program .....	63
ASSOCACHE : Associator Controller Caching .....	63
ASYTVS : Asynchronous Bufferflush .....	64
CACHE : Adabas Caching Facility Control .....	64
CACTIVATE : Adabas Caching Facility RABN Activation .....	65
CASSODSP / CASSOEXT / CASSOHSP / CASSOL64 / CASSOV64 : Adabas Caching Facility Associator RABN Cache Space Control .....	66
CASSOMAXS : Associator Cache Space Unit Size .....	67
CBUFNO: Number of Read-Ahead Buffers for Concurrent I/O Processing .....	69
CCTIMEOUT : Cache Space Area or RABN Inactivity Time .....	70
CDATADSP / CDATAEXT / CDATAHSP / CDATAL64 / CDATAV64 : Adabas Caching Facility Data Storage RABN Cache Space Control .....	71
CDATAMAXS : Data Storage Cache Space Unit Size .....	72
CDEMAND : Lowest Adabas Buffer Efficiency for Caching .....	74
CDISPSTAT : Adabas Caching Facility RABN Range Statistics Display Control .....	75
CDXnn : Collation Descriptor User Exit .....	75
CEXCLUDE : Adabas Caching Facility Excluded Command Types .....	76
CFILE : Adabas Caching Facility File Range .....	77
CLOGBMAX: Logged Buffer Size Limit .....	79

CLOGDEV : Multiple Command Log Device Type	CLOGSIZE : Multiple Command Log Size .....	79
CLOGLAYOUT : Command Logging Format .....		80
CLOGMAX: Total Logged Buffer Size Limit for a Command .....		81
CLOGMRG : Cluster Command Log Merge .....		82
CLUCACHENAME : Cluster Cache Structure/Area Name .....		83
CLUCACHESIZE : Size of the Global Cache Area .....		84
CLUCACHETYPE: Global Cache Area Storage Type Option .....		85
CLUCACHEUNCHANGED : Global Cache Unchanged Block Option .....		86
CLUGROUPNAME : Cluster Group Name .....		87
CLULOCKNAME : Cluster Lock Structure/Area Name .....		88
CLULOCKSIZE : Size of the Global Lock Area .....		89
CLUSTER : Adabas Nucleus Cluster Session Control .....		89
CMADDR and CMLADDR : GETMAIN Memory Pool Start Address .....		90
CMAXCSPS : Adabas Caching Facility Storage Area Quantities .....		91
CMDQMODE : Command Queue Mode .....		92
CMFIX or CMLFIX : GETMAIN Memory Pool Fixed Location .....		92
CMLxxxxx Parameters .....		93
CMSCOPE or CMLSCOPE : GETMAIN Memory Pool Scope .....		94
CMSIZE or CMLSIZE : GETMAIN Memory Pool Use and Size .....		95
CRETRY : Adabas Caching Facility Retry Time .....		96
CSTORAGE : Adabas Caching Facility Override Cache Type .....		97
CT : Command Timeout Limit .....		98
CWORKSTORAGE : Adabas Caching Facility Work Area Cache Type .....		98
CWORK2FAC : Adabas Caching Facility Work Part 2 Percentage .....		99
CWORK3FAC : Adabas Caching Facility Work Part 3 Percentage .....		100
CXFILE : Adabas Caching Facility Excluded Files .....		101
DATACACHE : Data Storage Controller Caching .....		102
DBID : Database Identification .....		103
DEVICE : Device Type .....		103
DIRRATIO / ELEMENTRATIO : Cluster Cache Directory-to-Data Elements Ratio .....		104
DSF : Adabas Delta Save Facility Control .....		105
DSFEX1 : Delta Save Facility Exit .....		106
DTP : Distributed Transaction Processing Control .....		106
DUALCLD : Dual Command Log Device Type	DUALCLS : Dual Command Log Size .....	107
DUALPLD : Dual Protection Log Device Type	DUALPLS : Dual Protection Log Size .....	108
EXCPVR : Control EXCP or EXCPVR Use .....		109
FASTPATH : Adabas Fastpath Control .....		109
FMXIO : Parallel LFIOP I/O Operations .....		110
FORCE : Allow Nucleus Database ID or Review Hub Table Entry Overwrite .....		111
GROUPS : User Group Interprocess Communication .....		112

HEXnn :Hyperdescriptor User Exit .....	113
IDTNAME : Define ID Table Name .....	113
IDTPSUP : Suppress IDTNAME in Adabas Output Messages .....	114
IGNDIB : Ignore DIB Entry for Adabas Nucleus .....	114
IGNDTP : Ignore Two-Phase Commit Area (Work Part 4) .....	115
INTNAS : Time Interval Between SYNS 60 Checkpoints .....	116
LBP : Length of Buffer Pool .....	116
LCP : Length of Security Pool .....	117
LDEUQP : Length of Unique Descriptor Pool .....	118
LDTP : Length of Distributed Transaction Processing Area (Work Part 4) .....	118
LFIOp : Asynchronous Buffer Pool Control .....	119
LFP : Length of Internal Format Buffer Pool .....	120
LI : Length of Table of ISNs (TBI) .....	121
LNKGNAME: Link Globals Table Name .....	122
LOCAL : Local Nucleus or Adabas Review Hub .....	122
LOGGING / LOGxxxx : Command Logging Control .....	123
LOGSIZE : Maximum Command Log Size .....	124
LP : Length of Data Protection Area (Work Part 1) .....	125
LQ : Length of Sequential Command Table .....	125
LRDP : Length of the Redo Pool .....	126
LRPL :Adabas or Event Replicator Replication Pool Size .....	127
LS : Length of Internal Sort Area .....	128
LU : Length of Intermediate User Buffer Area .....	129
LWKP2 : Length of ISN List Processing Area (Work Part 2) .....	130
LWP : Length of Adabas Work Pool .....	131
MODE : Mode of Operation .....	131
MSGBUF : Size of Message Buffer .....	132
MSGCONSL : Case of Messages Sent to the Console MSGDRUCK : Case of Messages Sent to DD/DRUCK MSGPRINT : Case of Messages Sent to DD/PRINT .....	133
MXCANCEL: Timeout Threshold for Canceled Peer Nucleus .....	134
MXCANCELWARN : Timeout Threshold for Canceled Peer Nucleus Warning .....	135
MXMSG: Timeout Threshold for Internucleus Command Processing .....	136
MXMSGWARN : Timeout Threshold for Internucleus Command Processing Warning .....	137
MXSTATUS : Member-Level XCF Status Monitoring Heartbeat Interval .....	138
MXTNA : Maximum Non-activity Time .....	139
MXTSX : Maximum Sx Command Execution Time .....	140
MXTT : Maximum Transaction Time .....	141
MXWTOR : Self-Termination Operator Query Interval .....	142
NAB : Number of Attached Buffers .....	143
NC : Number of Command Queue Elements .....	144
NCLOG : Number of Command Log Data Sets .....	145

NH : Number of Hold Queue Elements .....	146
NISNHQ : Number of ISNs in Hold Queue per User .....	146
NONDES : Nondesoriptor Searches .....	147
NPLOG : Number of Protection Log Data Sets .....	148
NQCID : Number of Active Command IDs per User .....	149
NSISN : Number of ISNs per ISN Table Element .....	149
NT : Number of Threads .....	151
NU : Number of User Queue Elements .....	152
NUCID : Adabas Cluster Nucleus ID .....	152
OPENRQ : Open Command Required .....	153
PGFIX: EXCPVR Page Fixing Control .....	154
PLOGDEV: Multiple Protection Log Device TypePLOGSIZE: Multiple Protection Log Size .....	154
PLOGSEGSDS: Spanned Record PLOG Control .....	155
PLOGRQ : Protection Log Required .....	155
PREFETCH : Activate Prefetch Feature .....	156
PREFICMD : Include Commands in Prefetch/Multifetch ProcessingPREFIFIL : Include Files in Prefetch/Multifetch Processing .....	157
PREFNREC : Multifetch Record Count .....	159
PREFSBL : Prefetch Buffer Length per Call .....	159
PREFTBL : Total Size of Prefetch Buffer .....	160
PREFXCMD (PREFXFIL) : Exclude Commands (Files) from Prefetch/Multifetch Processing .....	161
PROGRAM : Program to Run .....	162
QBLKSIZE : Block Size for Sequential Data Sets .....	164
READONLY : Read-Only Session Control .....	165
REPLICATION Parameter .....	166
REVFILTER : Review Record Filtering Control .....	167
REVIEW : Adabas Review Control .....	167
REVLOGBMAX: Logged Buffer Size Limit for Review .....	168
REVLOGMAX: Total Logged Buffer Size Limit for a Review Command .....	169
RPLCONNECTCOUNT Parameter .....	169
RPLCONNECTINTERVAL Parameter .....	170
RPLPARMS Parameter .....	170
RPLSORT Parameter .....	171
RPWARNINCREMENT Parameter .....	172
RPWARNINTERVAL Parameter .....	172
RPWARNMESSAGELIMIT Parameter .....	173
RPWARNPERCENT Parameter .....	173
SMGT : Error Handling and Message Buffering Facility .....	174
SORTCACHE : Sort Area Controller Caching .....	174
SPT : Activate Adabas Triggers and Stored Procedures .....	175
SUBMPSZ : GETMAIN Memory Pool for Subtasks .....	176
SVC : SVC Number .....	176
TAPEREL : Tape Handling Control (BS2000 Only) .....	177

TARGETID : Entire Net-Work Target ID .....	178
TASKCTGY : Adabas Batch/TP Task Category Control .....	178
TCPIP : TCP/IP Access Control (z/OS Only) .....	179
TCPURL : TCP/IP Universal Resource Locator .....	179
TEMPCACHE : Temp Area Controller Caching .....	180
TFLUSH : Synchronous Buffer Flush Duration .....	181
TLSCMD : Time Limit for S1, S2, and S4 Complex Searches .....	181
TMDRQ : Size of ATM Request Queue .....	182
TMETDATA : ET Data Control .....	183
TMGTT : Global Transaction Time Limit .....	184
TMLOG : ATM Manager Logging Control .....	185
TMMMSGSEV : Severity Threshold for Warning Messages .....	186
TMRESTART : ATM Restart Control .....	186
TMSYNCMGR : Support for External Transaction Coordinators .....	187
TMTCIDPREF : Dynamic Client ID Prefix .....	188
TNAA : Nonactivity Time Limit for Access-Only Users .....	188
TNAE : Nonactivity Time Limit for ET Logic Users .....	189
TNAX : Nonactivity Time Limit for Exclusive Control Users .....	190
TT : Transaction Time Limit .....	191
UEXn : User Exit .....	192
UTIONLY : Utilities-Only Session Control .....	193
VISTA : Adabas Vista Control .....	193
WORKCACHE : Work Area Controller Caching .....	194
6 Setting Adabas Session Parameters .....	195
General Guidelines .....	196
Using Session Statistics to Determine Parameter Settings .....	200
Parameters for Operating Systems .....	200
7 ADARUN Parameters Pertinent to Adabas Add-on Products .....	203
Adabas Caching Facility ADARUN Parameters .....	204
Adabas Fastpath ADARUN Parameters .....	205
Adabas Cluster Services ADARUN Parameters .....	205
Adabas Delta Save ADARUN Parameters .....	206
Adabas Parallel Services ADARUN Parameters .....	206
Adabas Review ADARUN Parameters .....	207
Adabas Transaction Manager ADARUN Parameters .....	207
Adabas Vista ADARUN Parameters .....	208
Event Replicator for Adabas ADARUN Parameters .....	208
III .....	211
8 Operator Commands .....	213
Entering Operator Commands .....	214
Operator Commands .....	216
SMGT Display Output Samples .....	248
9 Adabas Session Execution .....	251
Program Synchronization Using Operating System Services .....	252
BS2000 Session .....	253

z/OS Session .....	255
z/VM Session .....	259
VSE Session .....	260
Notes .....	262
Single-User Mode .....	264
User Profile .....	265
10 Adabas Utility Execution .....	267
Condition and Termination Codes .....	268
File and Database Usage .....	268
Disk Accesses by Utility and Function .....	271
Possible Resource Conflicts .....	272
Repetitive Utility Execution Under z/VM .....	274
11 Linking Applications to Adabas .....	275
How the Adabas API Works .....	276
Available Link Routines .....	278
Required Work Area .....	279
Required Application Reentrancy Properties .....	281
Adabas Control Block (ACB) Options .....	282
Extended Adabas Control Block (ACBX) Options .....	284
Programming Conventions for Issuing Direct Calls .....	286
Using the CICS COMMAREA or TWA with COBOL Programs .....	291
Using the Adabas API in Batch Mode .....	295
Support for OpenEdition z/OS Adabas Clients .....	299
12 Adabas Restart and Recovery .....	303
Work Data Set .....	304
Protection Log .....	305
Restart Operations .....	307
Database Recovery .....	309
13 Local Time Zone Change Support .....	315
Time Stamp Overview .....	316
Activating the Local Time Zone Change .....	316
14 Adabas Timeout Feature .....	319
Timeout Parameters .....	320
Action Taken If Time Limit Exceeded .....	321
Index .....	323



---

# 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:

- Controlling the Adabas Environment*** Provides an overview of the Adabas operating structure.
- Software AG Mainframe Product Licensing*** Describes Software AG'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 the BS2000, z/OS, z/VM, and VSE operating systems, and shows examples of the job streams for each system. 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 AG 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 Conventions

---

▪ Syntax Conventions .....	2
▪ Data Set Names .....	4

This document covers the following topics:

- [Syntax Conventions](#)
- [Data Set Names](#)

## 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.	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>ADADBS CHANGE FILE = file-number</b> </div> <p>The syntax elements ADADBS, CHANGE, and FILE are Adabas keywords.</p>
lowercase, italic, normal font	Syntax elements appearing in lowercase and normal, italic font identify items that you must supply.	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>ADADBS CHANGE FILE = file-number</b> </div> <p>The syntax element <i>file-number</i> 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.</p>
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.	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>[SORTDEV = { device-type   ADARUN-device }]</b> </div> <p>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 <i>not</i> be specified for the SORTDEV parameter.</p>
underlining	Underlining is used for two purposes:  1. To identify default values, wherever appropriate. Otherwise, the defaults are explained in the accompanying parameter descriptions.	<div style="border: 1px solid black; padding: 5px; text-align: center;"> <b>[LRECL = { record-buffer-length   <u>4000</u> }]</b> </div> <p>In the example above, 4000 is the default that will be used for the LRECL parameter if no other record buffer length is specified.</p> <div style="border: 1px solid black; padding: 5px; text-align: center; margin-top: 10px;"> <b>DEVICE</b> </div>

Convention	Description	Example
	<p>2. To identify the short form of a keyword.</p>	<p>In the example above, the short version of the DEVICE parameter is DE.</p>
<p>vertical bars ( )</p>	<p>Vertical bars are used to separate mutually exclusive choices.</p> <p><b>Note:</b> In more complex syntax involving the use of large brackets or braces, mutually exclusive choices are stacked instead.</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p><b>ADAORD { RESTRUCTUREF   REF }</b></p> </div> <p>In the example above, you must select RESTRUCTUREF or REF for this ADAORD function. There are no defaults.</p>
<p>brackets ([ ])</p>	<p>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.</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p><b>SORTSEQ = {</b>  <span style="margin-left: 20px;"><i>descriptor</i> [MU] [NU]</span>  <span style="margin-left: 20px;"><b>ISN [ STARTISN = value ]</b></span>  <span style="margin-left: 20px;"><i>physical-sequence</i></span>  <b>}</b></p> </div> <p>In this example, the SORTSEQ parameter and the MU, NU, and STARTISN subparameters are optional.</p> <p><b>Note:</b> Note that the mutually exclusive choices for the SORTSEQ parameter are stacked.</p>
<p>braces ({} )</p>	<p>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.</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p><b>SUSPEND</b>  <span style="margin-left: 20px;"><b>[TTSYN = { <i>time-available-to-sync</i>   ADARUN-TT } ]</b></span>  <span style="margin-left: 20px;"><b>[TRESUME = { <i>time-until-resume</i>   120 } ]</b></span>  <b>RESUME</b></p> </div> <p>In this example, either the SUSPEND or RESUME parameter is required.</p>
<p>indentation</p>	<p>Indentation is used to identify subparameters of a parameter.</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p><b>SUSPEND</b>  <span style="margin-left: 20px;"><b>[TTSYN = { <i>time-available-to-sync</i>   ADARUN-TT } ]</b></span>  <span style="margin-left: 20px;"><b>[TRESUME = { <i>time-until-resume</i>   120 } ]</b></span>  <b>RESUME</b></p> </div> <p>In this example, TTSYN and TRESUME are subparameters of the SUSPEND parameter.</p>
<p>ellipsis (...)</p>	<p>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.</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p><b>[FIELD = ' <i>field-name</i> [ , option ] ... ' ] ...</b></p> </div> <p>In this example, the FIELD parameter can be repeated. In addition, more than one option can be associated with a field.</p>

Convention	Description	Example
other punctuation and symbols	All other punctuation and symbols must be entered exactly as shown.	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <code>[FIELD = 'field-name [ , option]... ' ] ...</code> </div> <p>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.</p>

## Data Set Names

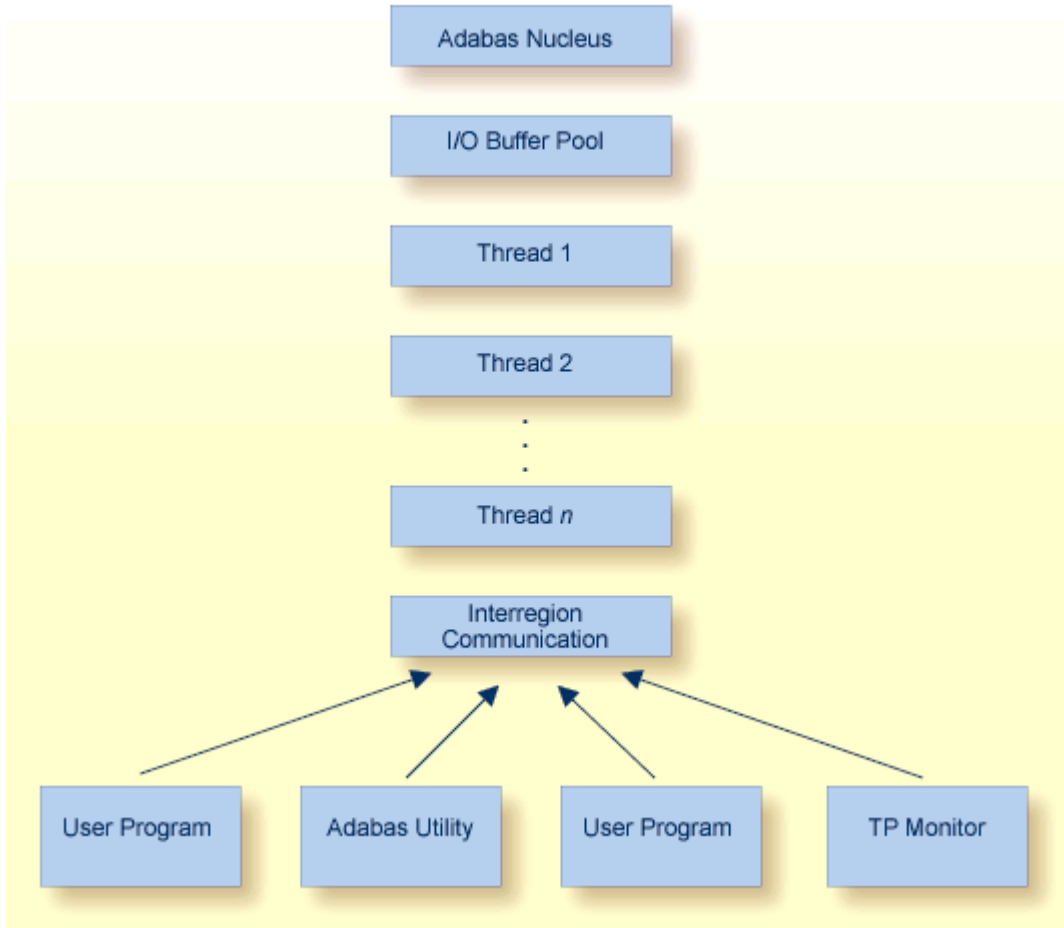
---

Data set names starting with DD are referred to in Adabas manuals with a slash separating the DD from the remainder of the data set name to accommodate VSE data set names that do not contain the DD prefix. The slash is not part of the data set name.

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*.

## 2 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:

Operating System	Storage Area Name
BS2000	task
z/OS	address space / data space / hiperspace
z/VM	virtual machine
VSE	partition / address space / data space

For consistency and simplification, this manual refers to all BS2000, z/OS, and VSE areas (task, address space, partition, region, etc.) as *regions*. z/VM areas are called *virtual machines*.



I

---



# 3 Software AG Mainframe Product Licensing

---

- Distributed License Libraries ..... 10
- License File ..... 11
- Transferring a License File from PC to a z/OS Host Using FTP ..... 12
- Transferring a License File from PC to a z/VSE Host Using FTP ..... 13
- Transferring a License File to a z/VM Host ..... 15
- Transferring a License File from PC to a BS2000 Host Using FTP ..... 16
- License File Installation ..... 17
- Product License Check FAQs ..... 17
- Using The License Utility: LICUTIL ..... 22

This document describes Software AG's mainframe product licensing procedures, utilities, and license file.

- [Distributed License Libraries](#)
- [License File](#)
- [Transferring a License File from PC to a z/OS Host Using FTP](#)
- [Transferring a License File from PC to a z/VSE Host Using FTP](#)
- [Transferring a License File to a z/VM Host](#)
- [Transferring a License File from PC to a BS2000 Host Using FTP](#)
- [License File Installation](#)
- [Product License Check FAQs](#)
- [Using The License Utility: LICUTIL](#)

## Distributed License Libraries

---

All Software AG mainframe products are distributed with licensing libraries that contain the files necessary to license your Software AG mainframe software. This section describes these data sets. All licensing data sets have names starting with the characters "MLC".

### z/OS Licensing Data Sets

The following licensing data sets are provided in z/OS environments:

Data Set Name	Description
MLC <i>vrs</i> .LOAD	Mainframe license check load library.
MLC <i>vrs</i> .JOBS	Mainframe license check example jobs.

### z/VSE Licensing Data Sets

The following licensing data sets are provided in z/VSE environments:

Data Set Name	Description
MLC <i>vrs</i> .LIBR	Mainframe license check load library.
MLC <i>vrs</i> .LIBJ	Mainframe license check example jobs.

## z/VM Licensing Data Sets

The following licensing data set is provided in z/VM environments:

Data Set Name	Description
MLC <i>vr</i> s.TAPE	Mainframe license check library.



**Note:** Software AG licensing software is not provided for VM/GCS environments.

## BS2000 Licensing Data Sets

The following licensing data sets are provided in BS2000 environments:

Data Set Name	Description
MLC <i>vr</i> s.MOD	Mainframe license check load library.
MLC <i>vr</i> s.JOBS	Mainframe license check example jobs.

## License File

You must install a valid license file on all mainframe platforms in which your Software AG mainframe product is installed. Your individual product license is shipped on the installation tape. It can also be shipped by e-mail, if desired. License files have file names in the format "*pppvr.xml*", where *ppp* is the three-character product code of the Software AG product (for example, ADA for Adabas, NAT for Natural, or WCP for Entire Net-Work) and *vr* represent the *major and minor version numbers* (respectively) of the product.

Once the license file is received, you must use native FTP commands to transfer it to your mainframe host before using it during installation.



**Caution:** Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

The license file is provided as an XML document (encoding is US-ASCII). This document can be viewed using a browsing tool or text editor on a PC. It contains text, which represents the licensing information and a digital signature, the license key. Among other things, it displays Software AG legal notices and copyright information. The product license is comprised of environmental information.




**Important:** Please note that the license file must remain in ASCII format -- even on the mainframe. It must not be modified. Any modification of the license file will invalidate the

digital signature and the license check will fail. If the check fails, you will not be able to run the product. In the event of a check failure, please contact your Software AG technical support representative.

## Transferring a License File from PC to a z/OS Host Using FTP

---

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.

 **Caution:** Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

▶ **To transfer a license file from the PC to a mainframe z/OS host, perform the following steps:**

- 1 Save the product license file e-mail attachment on your PC's hard disk.
- 2 Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.
- 3 Start an FTP session for communication with the z/OS host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the z/OS host.

- 4 Enter your z/OS host login ID and password.
- 5 Once the FTP session has been established, you can optionally delete the default working prefix. To determine what the default working prefix is, enter the following command:

```
ftp>pwd
```

To delete the default working prefix, enter the following command:

```
ftp>cd ..
```

- 6 Specify the z/OS file system and data set prefix (for example, "ADABAS", "NATURAL", "NETWORK", or any other convenient prefix) for the license file. The following example sets the data set prefix to "ADABAS":

```
ftp>cd 'ADABAS'
```

- 7 Switch to binary data mode (the license file must retain its format as ASCII during the transfer):

```
ftp>binary
```

- 8 Specify that the data set for the license file must be written with RECFM=FB and LRECL=80.

```
ftp>quote site RECFM=FB LRECL=80 BLKSIZE=4000
```

RECFM=F is also supported for product license files.

- 9 Write the license as a data set on the z/OS system. For example, if the license file name is *ada81.xml*, you might enter:

```
ftp>put ada81.xml
```

This command will create a data set called "*prefix*.ADA81.XML", where *prefix* is the data set prefix you specified in Step 6. For example, if you requested a prefix of "ADABAS" in Step 6, the data set name would be "ADABAS.ADA81.XML". The data set will have RECFM=FB, LRECL=80, and the license information stored in the data set will be in ASCII format.

- 10 Stop your FTP session.

```
ftp>quit
```

## Transferring a License File from PC to a z/VSE Host Using FTP

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.



**Caution:** Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

The FTP sample session described in these steps loads the license file onto a member of a z/VSE library. It is possible to load the license file onto a sequential z/VSE data set (with RECFM=F or FB and LRECL=80) instead. For more information, refer to your FTP documentation.

### ▶ To transfer a license file from the PC to a mainframe z/VSE host, perform the following steps:

- 1 Save the product license file e-mail attachment on your PC's hard disk.
- 2 Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.
- 3 Start an FTP session for communication with the z/VSE host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the z/VSE host.

- 4 Enter your z/VSE host login ID and password.

- 5 Once the FTP session has been established, you can optionally delete the default working prefix. To determine what the default working prefix is, enter the following command:

```
ftp>pwd
```

To delete the default working prefix, enter the following command:

```
ftp>cd ..
```

- 6 Point to the z/VSE library for the license file, (for example, "USERLIB"):

```
ftp>cd USERLIB
```

- 7 Point to the z/VSE sublibrary for the license file, (for example, "MYSUBLIB"):

```
ftp>cd MYSUBLIB
```

- 8 Switch to binary data mode (the license file must retain its format as ASCII during the transfer):

```
ftp>quote type I
```

- 9 Specify that the data set for the license file must be written with RECFM=FB and LRECL=80.

```
ftp>quote site recfm fb 80
```

RECFM=F is also supported for product license files.

- 10 Write the license as a member in the libraries on the z/VSE system you identified in Steps 6 and 7. For example, if the license file name is ada81.xml, you might enter:

```
ftp>put ada81.xml LICENSE.XML
```

This command will write the license file ada81.xml to a member called "LICENSE.XML". If you requested a library and sublibrary of "USERLIB" and "MYSUBLIB" in Steps 6 and 7, the LICENSE.XML file would reside in the USERLIB.MYSUBLIB library. The data set will have RECFM=FB, LRECL=80, and the license information stored in the member will be in ASCII format.

- 11 Stop your FTP session.

```
ftp>quit
```



## Transferring a License File to a z/VM Host

If a license file is supplied as an e-mail attachment or on tape, you must transfer it to the mainframe. This section describes how to transfer the license file from tape and from e-mail (using FTP).



**Note:** Software AG licensing software is not provided for VM/GCS environments.

▶ **To transfer the product license file from tape to a source library, complete the following steps:**

- 1 Refer to the *Report of Tape Creation* to accurately locate the license file on the tape.
- 2 If the tape is positioned somewhere other than the beginning of the tape, rewind it:

```
TAPE REW
```

- 3 Position the tape to the license file location, as shown in the *Report of Tape Creation*. For example, if the sequence number of ADA<sub>vr</sub>ZVM file, as shown by the *Report of Tape Creation*, is *n*, you must position over 3 *n* - 2 tape marks (that is, FSF 1 for the first data set, FSF 4 for the second, etc.).
- 4 Run the following z/VM commands in sequence to transfer the file off of the tape.

```
FILEDEF IN TAP1 (RECFM FB LRECL 80 BLKSIZE 3120
FILEDEF OUT DISK fn ft A (RECFM F LRECL 80 BLKSIZE 80
MOVEFILE IN OUT
```

where *fn* is the file name of the license file and *ft* is the file type of the license file.

▶ **To transfer the product license file e-mail attachment from Windows to a z/VM host, complete the following steps (make sure to switch to binary transfer and verify that the resulting data set has RECFM=F or FB and LRECL=80):**

- 1 Save the product license file e-mail attachment on your PC's hard disk.
- 2 Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.

```
cd dir
```

- 3 Start an FTP session with the z/VM host:

```
ftp host-name
```

where *host-name* is the name of the z/VM host.

- 4 Enter your FTP logon userid and password, as appropriate.
- 5 Switch the FTP data mode to binary:

```
binary
```

- 6 Request that the license data set to be written has RECFM=FB and LRECL=80:

```
quote site fix 80
```

- 7 Write the license file to the z/VM host. For example:

```
put fn.ft
```

where *fn* is the file name of the license file and *ft* is the file type of the license file.

The license file is written to your VM host.

- 8 Stop your FTP session:

```
quit
```



**Caution:** Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

## Transferring a License File from PC to a BS2000 Host Using FTP

---

If a license file is supplied as an e-mail attachment, you must transfer the attached license file from the PC to the mainframe using native FTP commands.



**Caution:** Using utilities instead of native FTP commands for the license file transfer may corrupt the license key and thus prevent the later execution of your Software AG mainframe product. This applies, for example, to file transfer based on 3270 terminal emulations that do not provide a true binary file transfer, but convert specific characters.

### ▶ To transfer a license file from the PC to a BS2000 host, perform the following steps:

Make sure to switch to binary transfer and verify that the resulting data set has file type SAM.

- 1 Save the product license file e-mail attachment on your PC's hard disk.
- 2 Open a command prompt window. In the command prompt window, change to the directory where you saved the license file.

```
cd dir
```

- 3 Start an FTP session for communication with the BS2000 host using the following FTP command:

```
ftp host-name
```

where *host-name* is the name of the BS2000 host.

- 4 Enter your FTP login ID and password.
- 5 Switch the FTP data mode to binary:

```
binary
```

- 6 Request that the license data set has a file type of SAM:

```
quote file *,fcctype=sam ↵
```

- 7 Write the license file to the BS2000 host. For example, to write the *ada81.xml* license file, you would enter:

```
put ada81.xml
```

The license file is written to your BS2000 host.

- 8 Stop your FTP session:

```
quit
```

## License File Installation

---

During the mainframe product installation process, the license file is loaded and processed as required by your Software AG mainframe product. Refer to the appropriate installation procedure for your Software AG mainframe product for further information.

## Product License Check FAQs

---

### 1. Why has Software AG introduced a license check for products on the mainframe?

Software AG wants to ensure that customers run our software products only on mainframe machines for which they have valid and sufficient product license. This ensures that we have better control of the use of our software products. On other platforms (UNIX, Windows), the product licensing has been established for many years.

### 2. What information is contained in a product license?

A product license is a sequential file with US-ASCII text in XML format containing the following items:

- Software AG header
- customer information (Name, ID)

- encrypted license key
- license expiration date (or unlimited)
- product information (product code, version, name)
- environment information (operating system type, CPU ID, system name, capacity)



**Caution:** Any modification of the license file will invalidate the digital signature and the license key check will fail. If the check fails, you will not be able to run the product. In the event of a check failure, please contact your Software AG technical support representative.

### 3. What exactly is meant by the CPU ID?

IBM defines the CPU ID as the "central processing complex node descriptor sequence number". It is the unique hexadecimal machine serial number without the machine model number.

#### z/OS

On z/OS systems, information about the CPU can be displayed on the operator console by entering the command:

```
D M=CPU
```

For example, suppose the console output from this command contains these lines:

```
EE174I 16.38.50 DISPLAY M 951
PROCESSOR STATUS
ID  CPU              SERIAL
00  +                0FA10E2096
01  +                0FA10E2096
CPC ND = 002096.S07.IBM.83.00000007A10E
CPC SI = 2096.V03.IBM.83.00000000007A10E
....
```

In this example, the CPU ID is "7A10E", as highlighted in the example above.

#### z/VSE

On z/VSE systems, information about the CPU can be displayed on the operator console by entering the operator command:

```
sir
```

For example, suppose the output from this command contains this line:

```
...
AR 0015 PROCESSOR = IBM 2096-V03 83 (7A10E83) LPAR = DAEX No. = 0007
....
```

The CPU ID is highlighted in the example.

## BS2000

On BS2000/OSD systems, the CPU ID (8-byte hexadecimal) consists of the machine serial number, the processor ID and the machine model number. The processor ID in the second byte of the CPU ID is ignored for the license check. Information about the CPU can be displayed by entering the following BS2000/OSD command:

```
/SHOW-SYSTEM-INFORMATION INFORMATION=*CPU-ID-LIST
```

For example, suppose the output from this command contains these lines:

```
...
%CONFIGURATION           = 7.500- S140-20A      <<
%CPU-ID-LIST :  ADR      0      = 1D02301375000000 <<
...
```

The most relevant information is shown in these lines. The first value identifies the machine type as "7.500- S140-20A", the second value identifies the physical CPU ID as "1D02301375" (trailing zeros can be omitted).

## z/VM

On z/VM systems, the CPU ID can be determined entering the command:

```
q cpu
```

Suppose the output from this command is:

```
CPUID = FF07A10E20968000
```

The first byte will always be "FF", followed by the CPU-ID (highlighted in the example).

#### 4. Which mainframe products use the Software AG common mainframe license routines?

- Adabas (version 8.1.4 or later)
- Com-plete (planned)
- Entire Net-Work (version 6.2.1 or later)
- Entire System Services (planned)

- Natural (planned)
- webMethods EntireX (planned)

#### 5. How is the product license file installed?

The product license file is supplied on the individual customer installation tape or separately via an e-mail attachment. For information on its installation, refer to the installation documentation for your product. For information on transferring the license file from a PC to a z/OS, z/VSE, z/VM, or BS2000 host, read the various *Transferring a License File from PC to a xxx Host* sections, elsewhere in this section.

In addition, steps for actually installing the license file are provided in the installation steps of your software product.

#### 6. How can I read the product license file?

- The product license file can be read on a PC using an XML editor (file type *.xml*) or any PC-based text editor.
- The license utility, LICUTIL, provided with the Software AG mainframe license check software includes a function that allows you to display the license (Display License function). For more information about the license utility, read *The License Utility: LICUTIL*, elsewhere in this section.
- When Natural is installed and runs, the license information can be displayed using function "L" of the Natural SYSTP utility.
- When Adabas, Complete, Entire Net-Work, Entire System Services, and webMethods EntireX are installed and run, they print a formatted license file into their logging files.

#### 7. How can I display machine-specific data?

Use the license utility, LICUTIL, to display machine-specific data. For more information about the license utility, read *The License Utility: LICUTIL*, elsewhere in this section.

#### 8. When is the product license checked?

The product license is checked during every time the product is initialized. In addition, the product license is checked once a day.

Thirty days before the license expires, license check failure messages are produced. Your software product will still function, but these messages warn you that it is time to obtain a new license.

#### 9. Which items of a product license are checked?

Product license items are divided into three logical license groups (License Key, Product-Specific Information, and Machine-Specific Information) and checked in that order. If an inconsistency in any one of these groups is detected by the license checker, the succeeding items of that logical license group are also checked. Then the check terminates with corresponding error messages. For example, if the license key expires, the license key check terminates with an error before the license checker checks any product-specific or machine-specific information.

The following table identifies the product license items comprising each logical group:

Checking Order	Logical License Group Name	Product License Items Included
1	License Key	The encrypted license key
2	Product-Specific Information	<ul style="list-style-type: none"> <li>■ The expiration date (if any)</li> <li>■ The operating system (z/OS, z/VSE, z/VM, or BS2000/OSD)</li> <li>■ The product code (for example, ADA, COm, EXX, NAT, or WCP)</li> <li>■ The product version, in either "v.r" or "v.r.s" format. For further information on product versions, refer to the description of "v.r" or "v.r.s" in the product documentation Glossary.</li> </ul>
3	Machine-Specific Information	<ul style="list-style-type: none"> <li>■ The machine CPU ID</li> <li>■ The machine capacity (only in z/OS, z/VSE, and z/VM environments)</li> <li>■ The logical partition</li> </ul>

**10. Does the Software AG mainframe license checker software support a disaster recovery scenario?**

Yes. Please contact your Software AG sales representative to obtain a license file which is able to support disaster recovery scenarios. The parameter `DisasterRecoveryCPUIDs` in this license file allows you to specify one or more CPU IDs of the machines to be used when your production machine fails. If you do not have these CPU IDs available, it is possible to specify the value "Unknown". In this case, the license checker issues a warning message when a disaster recovery scenario is assumed.

In a disaster recover scenario, no machine data besides the CPU ID is checked.

**11. What happens if the product license expires or is incorrect, insufficient, or not installed?**

The reaction varies by Software AG mainframe product. For more information, refer to the documentation for the product.

Thirty days before the license expires, license check failure messages are produced. Your software product will still function, but these messages warn you that it is time to obtain a new license.

**12. How can I get a new product license file if the delivered license file is insufficient for my environment?**

Contact your Software AG sales representative to get a correct product license file according to your contract. It can be sent by e-mail either in ASCII format or in the converted assembler input format.

**13. Do I need separate product licenses for different machines?**

No, it is possible to have multiple CPU IDs defined in one license file.

## Using The License Utility: LICUTIL

The license utility, LICUTIL, is a utility provided with Software AG's mainframe license checking software. It can perform the following functions:

- Check the license file. LICUTIL is mainly intended to be used to check the license file during the installation of the product. The original license file must be provided as input along with the product code and version. If the license file is not valid, appropriate error messages are returned.
- Display the license file and machine data. LICUTIL is able to display the contents of the license file and machine data, such as the CPU ID, LPAR, or MSU information.
- Convert the license file into an assembler source module. LICUTIL can be used to convert the license file into an assembler source module that can then be used to assemble and link a license module. For more information about using license key modules, refer to the documentation provided with your Software AG product.

The following table briefly describes the functions and parameters in the license utility. More detailed information of each function is provided later in this section:

Function	Parameters	Description
<b>CHECK</b>	---	Check the product license.
<b>DISPLAY</b>	ALL	Display the product license items and machine data.
	LICENSE	Display only the product license items.
	MACHINE	Display only the machine data.
<b>END   EXIT</b>	---	Stop the license utility.
<b>HELP</b>	---	Produce a list of available LICUTIL functions and the syntax of the functions.
<b>MAKE</b>	---	Create an assembler source module from the license file.
<b>SET</b>	DATE= <i>yyyy-mm-dd</i> or DATE= <i>yyyymmdd</i>	Set the date for a license check. The default is today.
	FORMAT= {FREE   LEFT   ← [STRUCTURED]}	Produce output without XML tags (FREE), with each XML tag in a new line (LEFT), or with each XML tag indented (STRUCTURED). The default is FREE.
	PRODVER= <i>pppv.r.s</i>	Set the product code and version, where <i>ppp</i> is the Software AG 3-character product code and <i>v.r.s</i> is the three-digit version number.



Error messages for the license utility all begin with the prefix MLC. They are documented with the rest of your Software AG product messages.

This section covers the following topics:

- [CHECK Function: Check the Product License](#)
- [DISPLAY Function: Display License Items and Machine Data](#)
- [END/EXIT Functions: Terminate the License Utility](#)
- [HELP Function: View LICUTIL Help](#)
- [MAKE Function: Convert the License File to Assembler Source](#)
- [SET Function: Set LICUTIL Options](#)
- [JCL Examples and Requirements](#)

### **CHECK Function: Check the Product License**

The CHECK function can be used to check the license file against the product's execution environment. There are no parameters, however to run the CHECK function successfully, you must have previously specified the [SET PRODVER](#) function to provide the product code and version number.

The following items are checked, in the following order:

1. License key
2. Product-specific data, including the product code, version, operating system, and expiration date.
3. Machine-specific data, including the CPU IDs, the capacity (on only z/OS, z/VSE and z/VM systems), and the logical partition.

The syntax of the CHECK function is:



CHECK

Sample JCL or JCS is provided for the CHECK function in member LICCHECK of the Software AG mainframe license JOBS data set (MLC<sub>VRS</sub>.JOBS).

#### **Example:**

In the following example, the product code and version are set to ADA8.1.4 (Adabas 8.1.4) and then the product license is checked.

```
SET PRODVER=ADA8.1.4  
CHECK
```

## DISPLAY Function: Display License Items and Machine Data

Use the DISPLAY function to display the contents of the product license items (license file, product-specific information, and machine-specific information) and to display machine data (CPU ID, machine type, model ID, LPAR, MSU) about the machine on which the utility is being run.

The syntax of the DISPLAY function is:

```
DISPLAY { ALL | LICENSE | MACHINE }
```

The parameters are described in the following table. At least one parameter is required.

Parameters	Description
ALL	Display the product license items and machine data.
LICENSE	Display only the product license items.
MACHINE	Display only the machine data.

Sample JCL or JCS is provided for the DISPLAY function in member LICMDATA of the Software AG mainframe license JOBS data set (MLC*vrs*.JOBS).

### Examples

In the following example, both license and machine data is displayed:

```
DISPLAY ALL
```

In the following example, only license data is displayed:

```
DISPLAY LICENSE
```

In the following example, the license data is displayed in structured format:

```
SET FORMAT=STRUCTURED  
DISPLAY LICENSE
```

In the following example, only machine data is displayed:

```
DISPLAY MACHINE
```

## END/EXIT Functions: Terminate the License Utility

Use the END or EXIT functions to terminate the license utility. These functions should be used if LICUTIL is running interactive in environments other than batch (for example TSO under z/OS).

The syntax of the END and EXIT functions is:

```
{ END | EXIT }
```

## HELP Function: View LICUTIL Help

Use the HELP function to review a list of the available LICUTIL functions and their syntax.

The syntax of the HELP function is:

```
HELP
```

## Sample Output

Here is some sample output from this function:

```
>>>HELP
LICUTIL reads a Software AG product license from SYSUT1.
It has the following functions:
  CHECK   : Check license read from SYSUT1
  DISPLAY : Display license and machine
  END     : Terminate LICUTIL
  MAKE    : write license for assembly to SYSUT2
  SET     : sets execution defaults

SET keyword=value accepts the following keyword(s):
DATE=YYYY-MM-DD or YYYYMMDD - Set date for license check (default TODAY)
PRODVER=zzzv.r.s             - Set product code zzz and version release
                               SM-level v.r.s for license check
FORMAT=LEFT/STRUCTURED/FREE - Set format of license display
```

## MAKE Function: Convert the License File to Assembler Source

The MAKE function can be used to convert the license file into an assembler source. After assembling this source (and linking it), the resulting license module can be used by defined Software AG products (for example Adabas or Natural) instead of the license files. For more information, please read the documentation for your specific Software AG mainframe product.

The syntax of the MAKE function is:



MAKE

Sample JCL or JCS is provided for the CHECK function in member LICMAKE of the Software AG mainframe license JOBS data set (MLC*vrs*.JOBS).

### Sample Output

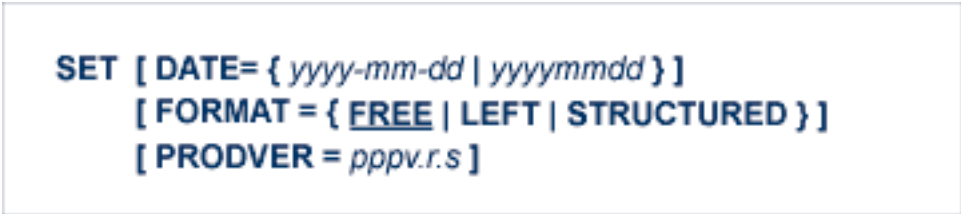
Here is some sample output from this function when run on a z/OS system:

```
MLC9006 License converted to assembler output. 042 records written  
to SYSUT2.
```

## SET Function: Set LICUTIL Options

Use the SET function to set options for the license utility.

The syntax of the SET function is:



```
SET [ DATE= { yyyy-mm-dd | yyyymmdd } ]  
    [ FORMAT = { FREE | LEFT | STRUCTURED } ]  
    [ PRODVER = pppv.r.s ]
```

The parameters are described in the following table. At least one parameter is required.

Parameters	Description
DATE	Sets the date for the license check. The default is today. The format of the date can be either <i>yyyy-mm-dd</i> or <i>yyyymondd</i> , where <i>yyyy</i> is the four-character year, <i>mm</i> is the two-digit month, <i>mon</i> is the three-character month abbreviation, and <i>dd</i> is the two-digit day.
FORMAT	Sets the display format for the <b>DISPLAY</b> function. There are three different formats: <ul style="list-style-type: none"> <li>■ Specify FREE (the default) to produce output with XML tags.</li> <li>■ Specify LEFT to produce output with XML tags</li> <li>■ Specify STRUCTURED to produce output with each XML tag indented.</li> </ul>
PRODVER	Sets the product code and version for the <b>CHECK</b> function. The format in which the product code and version should be specified is <i>pppv.r.s</i> , where <i>ppp</i> is the three-character Software AG product code (such as ADA, NAT or WCP) and <i>v.r.s</i> is the three-digit version number.

Sample JCL or JCS is provided for the DISPLAY function in member LICCHECK of the Software AG mainframe license JOBS data set (MLC*vrs*.JOBS).

### Example

In the following example, the format of the DISPLAY output will not include XML tags and the product and version are set as Adabas 8.1.4:

```
SET FORMAT=FREE
SET PRODVER=ADA8.1.4
```

### JCL Examples and Requirements

Sample JCL of the license utility functions can be found in the following members of the Software AG mainframe license example libraries (MLC*vrs*.JOBS in z/OS and BS2000 environments, MLC*vrs*.LIBJ in z/VSE environments, and MLC*vrs*.TAPE in VM environments).

Member	Functions Included	Demonstrates how to...
LICCHECK	SET PRODVER=PPP <i>v.r.s</i> DISPLAY CHECK	Check a license file with LICUTIL.
LICMAKE	MAKE	Convert the license file into an assembler source.
LICMDATA	DISPLAY MACHINE	Display machine data. This function can be used to send Software AG the required data for a license file.

This section covers the following topics:

- [z/OS and z/VM JCL Requirements](#)
- [z/VSE JCL Requirements](#)

- [BS2000/OSD JCL Requirements](#)

### z/OS and z/VM JCL Requirements

The following file assignments are used by the license utility in z/OS and z/VM environments:

DD Name	Description
SYSIN	Input of LICUTIL control parameters.
SYSPRINT	Output of LICUTIL run.
SYSUT1	Input of original license file.
SYSUT2	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	DD Name			
	SYSIN	SYSPRINT	SYSUT1	SYSUT2
CHECK	X	X	X	
DISPLAY ALL	X	X	X	
DISPLAY LICENSE	X	X	X	
DISPLAY MACHINE	X	X		
MAKE	X	X	X	X
HELP	X	X		

### z/VSE JCL Requirements

The following file assignments are used by the license utility in z/VSE environments:

File Assignment	Description
SYSIPT	Input of LICUTIL control parameters and the input of original license file. <b>Note:</b> The license file has to be stored in a sublibrary.
SYSLIST	Output of LICUTIL run.
SYSPCH	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	File Assignments		
	SYSIPT	SYSLIST	SYSPCH
CHECK	X	X	
DISPLAY ALL	X	X	
DISPLAY LICENSE	X	X	
DISPLAY MACHINE	X	X	
MAKE	X	X	X
HELP	X	X	

### BS2000/OSD JCL Requirements

The following file assignments are used by the license utility in BS2000 environments:

System File	Description
SYSDTA	Input of LICUTIL control parameters.
SYSOUT	Output of LICUTIL run.

Link Name	Description
SYSUT1	Input of original license file.
SYSUT2	Output of license converted to assembler source.

The following table shows which file assignments are required in the JCL by each LICUTIL function (an X in a table cell indicates that the file assignment is required; a blank cell indicates that file assignment is not required):

License Utility Function	File Assignment			
	SYSDTA	SYSOUT	SYSUT1	SYSUT2
CHECK	X	X	X	
DISPLAY ALL	X	X	X	
DISPLAY LICENSE	X	X	X	
DISPLAY MACHINE	X	X		
MAKE	X	X	X	X
HELP	X	X		





# 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 DDCARD ( z/OS, z/VM, or BS2000) or VSE CARD 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, 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

---

▪ AOSLOG : Log Active Nucleus Update Commands .....	62
▪ AREXCLUDE : Exclude Files from Autorestart Processing .....	62
▪ ARMNAME : Activate Automatic Restart Management (ARM) Program .....	63
▪ ASSOCACHE : Associator Controller Caching .....	63
▪ ASYTVS : Asynchronous Bufferflush .....	64
▪ CACHE : Adabas Caching Facility Control .....	64
▪ CACTIVATE : Adabas Caching Facility RABN Activation .....	65
▪ CASSODSP / CASSOEXT / CASSOHSP / CASSOL64 / CASSOV64 : Adabas Caching Facility Associator RABN Cache Space Control .....	66
▪ CASSOMAXS : Associator Cache Space Unit Size .....	67
▪ CBUFNO: Number of Read-Ahead Buffers for Concurrent I/O Processing .....	69
▪ CCTIMEOUT : Cache Space Area or RABN Inactivity Time .....	70
▪ CDATADSP / CDATAEXT / CDATAHSP / CDATAL64 / CDATAV64 : Adabas Caching Facility Data Storage RABN Cache Space Control .....	71
▪ CDATAMAXS : Data Storage Cache Space Unit Size .....	72
▪ CDEMAND : Lowest Adabas Buffer Efficiency for Caching .....	74
▪ CDISPSTAT : Adabas Caching Facility RABN Range Statistics Display Control .....	75
▪ CDXnn : Collation Descriptor User Exit .....	75
▪ CEXCLUDE : Adabas Caching Facility Excluded Command Types .....	76
▪ CFILE : Adabas Caching Facility File Range .....	77
▪ CLOGBMAX: Logged Buffer Size Limit .....	79
▪ CLOGDEV : Multiple Command Log Device TypeCLOGSIZE : Multiple Command Log Size .....	79
▪ CLOGLAYOUT : Command Logging Format .....	80
▪ CLOGMAX: Total Logged Buffer Size Limit for a Command .....	81
▪ CLOGMRG : Cluster Command Log Merge .....	82
▪ CLUCACHENAME : Cluster Cache Structure/Area Name .....	83
▪ CLUCACHESIZE : Size of the Global Cache Area .....	84
▪ CLUCACHETYPE: Global Cache Area Storage Type Option .....	85
▪ CLUCACHEUNCHANGED : Global Cache Unchanged Block Option .....	86
▪ CLUGROUPNAME : Cluster Group Name .....	87
▪ CLULOCKNAME : Cluster Lock Structure/Area Name .....	88
▪ CLULOCKSIZE : Size of the Global Lock Area .....	89

▪ CLUSTER : Adabas Nucleus Cluster Session Control .....	89
▪ CMADDR and CMLADDR : GETMAIN Memory Pool Start Address .....	90
▪ CMAXCSPS : Adabas Caching Facility Storage Area Quantities .....	91
▪ CMDQMODE : Command Queue Mode .....	92
▪ CMFIX or CMLFIX : GETMAIN Memory Pool Fixed Location .....	92
▪ CMLxxxx Parameters .....	93
▪ CMSCOPE or CMLSCOPE : GETMAIN Memory Pool Scope .....	94
▪ CMSIZE or CMLSIZE : GETMAIN Memory Pool Use and Size .....	95
▪ CRETRY : Adabas Caching Facility Retry Time .....	96
▪ CSTORAGE : Adabas Caching Facility Override Cache Type .....	97
▪ CT : Command Timeout Limit .....	98
▪ CWORKSTORAGE : Adabas Caching Facility Work Area Cache Type .....	98
▪ CWORK2FAC : Adabas Caching Facility Work Part 2 Percentage .....	99
▪ CWORK3FAC : Adabas Caching Facility Work Part 3 Percentage .....	100
▪ CXFILE : Adabas Caching Facility Excluded Files .....	101
▪ DATACACHE : Data Storage Controller Caching .....	102
▪ DBID : Database Identification .....	103
▪ DEVICE : Device Type .....	103
▪ DIRRATIO / ELEMENTRATIO : Cluster Cache Directory-to-Data Elements Ratio .....	104
▪ DSF : Adabas Delta Save Facility Control .....	105
▪ DSFEX1 : Delta Save Facility Exit .....	106
▪ DTP : Distributed Transaction Processing Control .....	106
▪ DUALCLD : Dual Command Log Device TypeDUALCLS : Dual Command Log Size .....	107
▪ DUALPLD : Dual Protection Log Device TypeDUALPLS : Dual Protection Log Size .....	108
▪ EXCPVR : Control EXCP or EXCPVR Use .....	109
▪ FASTPATH : Adabas Fastpath Control .....	109
▪ FMXIO : Parallel LFIOP I/O Operations .....	110
▪ FORCE : Allow Nucleus Database ID or Review Hub Table Entry Overwrite .....	111
▪ GROUPS : User Group Interprocess Communication .....	112
▪ HEXnn :Hyperdescriptor User Exit .....	113
▪ IDTNAME : Define ID Table Name .....	113
▪ IDTPSUP : Suppress IDTNAME in Adabas Output Messages .....	114
▪ IGNDIB : Ignore DIB Entry for Adabas Nucleus .....	114
▪ IGNDTP : Ignore Two-Phase Commit Area (Work Part 4) .....	115
▪ INTNAS : Time Interval Between SYNS 60 Checkpoints .....	116
▪ LBP : Length of Buffer Pool .....	116
▪ LCP : Length of Security Pool .....	117
▪ LDEUQP : Length of Unique Descriptor Pool .....	118
▪ LDTP : Length of Distributed Transaction Processing Area (Work Part 4) .....	118
▪ LFIOP : Asynchronous Buffer Pool Control .....	119
▪ LFP : Length of Internal Format Buffer Pool .....	120
▪ LI : Length of Table of ISNs (TBI) .....	121
▪ LNKGNAME: Link Globals Table Name .....	122
▪ LOCAL : Local Nucleus or Adabas Review Hub .....	122
▪ LOGGING / LOGxxxx : Command Logging Control .....	123

▪ LOGSIZE : Maximum Command Log Size .....	124
▪ LP : Length of Data Protection Area (Work Part 1) .....	125
▪ LQ : Length of Sequential Command Table .....	125
▪ LRDP : Length of the Redo Pool .....	126
▪ LRPL : Adabas or Event Replicator Replication Pool Size .....	127
▪ LS : Length of Internal Sort Area .....	128
▪ LU : Length of Intermediate User Buffer Area .....	129
▪ LWKP2 : Length of ISN List Processing Area (Work Part 2) .....	130
▪ LWP : Length of Adabas Work Pool .....	131
▪ MODE : Mode of Operation .....	131
▪ MSGBUF : Size of Message Buffer .....	132
▪ MSGCONSL : Case of Messages Sent to the Console	
MSGDRUCK : Case of Messages Sent to	
DD/DRUCKMSGPRINT : Case of Messages Sent to DD/PRINT .....	133
▪ MXCANCEL: Timeout Threshold for Canceled Peer Nucleus .....	134
▪ MXCANCELWARN : Timeout Threshold for Canceled Peer Nucleus Warning .....	135
▪ MXMSG: Timeout Threshold for Internucleus Command Processing .....	136
▪ MXMSGWARN : Timeout Threshold for Internucleus Command Processing Warning .....	137
▪ MXSTATUS : Member-Level XCF Status Monitoring Heartbeat Interval .....	138
▪ MXTNA : Maximum Non-activity Time .....	139
▪ MXTSX : Maximum Sx Command Execution Time .....	140
▪ MXTT : Maximum Transaction Time .....	141
▪ MXWTOR : Self-Termination Operator Query Interval .....	142
▪ NAB : Number of Attached Buffers .....	143
▪ NC : Number of Command Queue Elements .....	144
▪ NCLOG : Number of Command Log Data Sets .....	145
▪ NH : Number of Hold Queue Elements .....	146
▪ NISNHQ : Number of ISNs in Hold Queue per User .....	146
▪ NONDES : Nondesoriptor Searches .....	147
▪ NPLOG : Number of Protection Log Data Sets .....	148
▪ NQCID : Number of Active Command IDs per User .....	149
▪ NSISN : Number of ISNs per ISN Table Element .....	149
▪ NT : Number of Threads .....	151
▪ NU : Number of User Queue Elements .....	152
▪ NUCID : Adabas Cluster Nucleus ID .....	152
▪ OPENRQ : Open Command Required .....	153
▪ PGFIX: EXCPVR Page Fixing Control .....	154
▪ PLOGDEV: Multiple Protection Log Device Type	
PLOGSIZE: Multiple Protection Log Size .....	154
▪ PLOGSEGSDS: Spanned Record PLOG Control .....	155
▪ PLOGRQ : Protection Log Required .....	155
▪ PREFETCH : Activate Prefetch Feature .....	156
▪ PREFICMD : Include Commands in Prefetch/Multifetch Processing	
PREFIFIL : Include Files in Prefetch/Multifetch	
Processing .....	157
▪ PREFNREC : Multifetch Record Count .....	159
▪ PREFSBL : Prefetch Buffer Length per Call .....	159
▪ PREFTBL : Total Size of Prefetch Buffer .....	160

▪ PREFXCMD (PREFXFIL) : Exclude Commands (Files) from Prefetch/Multifetch Processing .....	161
▪ PROGRAM : Program to Run .....	162
▪ QBLKSIZE : Block Size for Sequential Data Sets .....	164
▪ READONLY : Read-Only Session Control .....	165
▪ REPLICATION Parameter .....	166
▪ REVFILTER : Review Record Filtering Control .....	167
▪ REVIEW : Adabas Review Control .....	167
▪ REVLOGBMAX: Logged Buffer Size Limit for Review .....	168
▪ REVLOGMAX: Total Logged Buffer Size Limit for a Review Command .....	169
▪ RPLCONNECTCOUNT Parameter .....	169
▪ RPLCONNECTINTERVAL Parameter .....	170
▪ RPLPARMS Parameter .....	170
▪ RPLSORT Parameter .....	171
▪ RPWARNINCREMENT Parameter .....	172
▪ RPWARNINTERVAL Parameter .....	172
▪ RPWARNMESSAGELIMIT Parameter .....	173
▪ RPWARNPERCENT Parameter .....	173
▪ SMGT : Error Handling and Message Buffering Facility .....	174
▪ SORTCACHE : Sort Area Controller Caching .....	174
▪ SPT : Activate Adabas Triggers and Stored Procedures .....	175
▪ SUBMPSZ : GETMAIN Memory Pool for Subtasks .....	176
▪ SVC : SVC Number .....	176
▪ TAPEREL : Tape Handling Control (BS2000 Only) .....	177
▪ TARGETID : Entire Net-Work Target ID .....	178
▪ TASKCTGY : Adabas Batch/TP Task Category Control .....	178
▪ TCPIP : TCP/IP Access Control (z/OS Only) .....	179
▪ TCPURL : TCP/IP Universal Resource Locator .....	179
▪ TEMPCACHE : Temp Area Controller Caching .....	180
▪ TFLUSH : Synchronous Buffer Flush Duration .....	181
▪ TLSCMD : Time Limit for S1, S2, and S4 Complex Searches .....	181
▪ TMDRQ : Size of ATM Request Queue .....	182
▪ TMETDATA : ET Data Control .....	183
▪ TMGTT : Global Transaction Time Limit .....	184
▪ TMLOG : ATM Manager Logging Control .....	185
▪ TMMMSGSEV : Severity Threshold for Warning Messages .....	186
▪ TMRESTART : ATM Restart Control .....	186
▪ TMSYNCMGR : Support for External Transaction Coordinators .....	187
▪ TMTCIDPREF : Dynamic Client ID Prefix .....	188
▪ TNAA : Nonactivity Time Limit for Access-Only Users .....	188
▪ TNAE : Nonactivity Time Limit for ET Logic Users .....	189
▪ TNAX : Nonactivity Time Limit for Exclusive Control Users .....	190
▪ TT : Transaction Time Limit .....	191
▪ UEXn : User Exit .....	192
▪ UTIONLY : Utilities-Only Session Control .....	193
▪ VISTA : Adabas Vista Control .....	193



- WORKCACHE : Work Area Controller Caching ..... 194

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

Parameter	Use to	Values	Default
A0slog	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.	<i>arm-pgm-name</i>	---
ASSocache	Activate or deactivate controller caching for the Associator component (z/OS and z/VM only).	YES   NO	YES
ASYtvs	Flush buffers asynchronously based on volume serial number.	YES   NO	YES
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)	<i>rabn   rabn1 - rabnx</i>	---
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)	<i>rabn   rabn1 - rabnx</i>	---
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)	<i>rabn   rabn1 - rabnx</i>	---
CASSOV64	Cache an Associator RABN or RABN range in the virtual 64 storage cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters.  This parameter is available only in z/OS environments.  (Adabas Caching Facility)	<i>rabn   rabn1 - rabnx</i>	---
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

Parameter	Use to	Values	Default
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
CDATADsp	Cache a Data Storage RABN or RABN range in the data space cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters. (Adabas Caching Facility)	<i>rabn</i>   <i>rabn1 - rabnx</i>	---
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.  Only extended memory is available for BS2000 RISC machines.  (Adabas Caching Facility)	<i>rabn</i>   <i>rabn1 - rabnx</i>	---
CDATAHsp	Cache a Data Storage RABN or RABN range in the hiperspace cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters.  This parameter is available only in z/OS environments.  (Adabas Caching Facility)	<i>rabn</i>   <i>rabn1 - rabnx</i>	---
CDATAV64	Cache a Data Storage RABN or RABN range in the virtual 64 storage cache. This parameter is mutually exclusive with the CFILE and CXFILE parameters.  This parameter is available only in z/OS environments.  (Adabas Caching Facility)	<i>rabn</i>   <i>rabn1 - rabnx</i>	---
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 memory, data spaces, or hiperspaces)  1M (virtual 64 memory)
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

Parameter	Use to	Values	Default
CDIsostat	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
CDXnn	Specify the user routine of collation descriptor user exits. The <i>nn</i> 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:  <pre>CFILE=({ <i>fnr</i> ↵   <i>fnr1</i>-<i>fnrx</i>}[ ,<i>class</i>][ ,<i>scope</i>][ ,<i>cachetype</i>])</pre> where: <ul style="list-style-type: none"><li>■ <i>fnr</i>, <i>fnr1</i>, <i>fnrx</i>: valid file numbers. When a range is specified, the lower file number must be specified first (<i>fnr1</i>).</li><li>■ <i>class</i>: 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.</li><li>■ <i>scope</i>: Identifies the type of caching used: Associator (A), Data Storage (D), or both (B).</li><li>■ <i>cachetype</i>: Identifies the cache storage type used: data space (D), extended (E), hiperspace (H), or virtual 64 (V). Virtual 64 storage cache is only available in z/OS environments.</li></ul> This parameter is mutually exclusive with the CASS0xxx, CDATAxxx and CSTORAGE parameters.  (Adabas Caching Facility)	<i>fnr</i> , <i>fnr1</i> , <i>fnrx</i> : 0 - n  <i>class</i> : 1   2   3   4   5  <i>scope</i> : A   D   B  <i>cachetype</i> : D   E   H   V	<i>fnr</i> , <i>fnr1</i> , <i>fnrx</i> : ---  <i>class</i> : 3  <i>scope</i> : B  <i>cachetype</i> : D
CLOGBmax	Specify the maximum size (bytes) of a logged buffer.	368 - 4294967295	4096

Parameter	Use to	Values	Default
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 <i>Adabas Device Types and Block Sizes</i> .	value of the DEV parameter
CLOGLayout	Specify the format of the Adabas command log.	5   8	5
CLOGMax	Specify the maximum size (bytes) of all of the logged buffers allowed for an Adabas command.	368 - 4294967295	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	---
CLUCACHEName	Identifies the name of the Adabas cluster cache structure/area in which the current nucleus participates.  (Adabas Cluster Services)	<i>cluster-name</i>	---
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) or virtual 64 cache (V64). The V64 value for this parameter is available in z/OS environments only.  (Adabas Parallel Services)	DSP   V64	DSP
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 Entire Net-Work Client.  (Adabas Cluster Services)	<i>groupname</i>	---
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)	<i>lockstructurename</i>	---

Parameter	Use to	Values	Default
CLULOCKSize	Specify the amount of storage allocated for the global lock area. Specify values for this parameter using the following syntax:  <code>nnn[K   M   G]</code>  where <i>nnn</i> : 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	---
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
CMAddr	Identify the starting address of the GETMAIN common memory pool <i>above</i> the 16M line. This parameter is valid only in BS2000 environments running Sockets versions less than 2.2.	X'1000000' - X'FE000000'	0(X'1000000')
CMAxcsp	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
CMDqmode	Indicate where the command queue memory pool should be allocated: below the 16 MB line (BELOW) or above it (ABOVE). This parameter is valid only in BS2000 environments.	ABOVE   BELOW	ABOVE (BELOW for Adabas versions prior to Version 8)
CMFix	Indicate whether the GETMAIN common memory pool must have a fixed location <i>above</i> the 16 MB line. This parameter is valid only in BS2000 environments running Sockets versions less than 2.2.	YES   NO	NO
CMLAddr	Identify the starting address of the GETMAIN common memory pool <i>below</i> the 16M line. This parameter is valid only in BS2000 environments running Sockets versions less than 2.2.	X'10000' - X'E00000'	0(X'10000')
CMLFix	Indicate whether the GETMAIN common memory pool must have a fixed location <i>below</i> the 16 MB line.	YES   NO	NO

Parameter	Use to	Values	Default
	This parameter is valid only in BS2000 environments running Sockets versions less than 2.2.		
CMLScope	Control access to the GETMAIN common memory pool <i>below</i> the 16 MB line. Valid values are "GROUP" (the pool can be accessed only by other tasks using the same user ID) and "GLOBAL" (the pool can be accessed by all user IDs in the system). This parameter is valid only in BS2000 environments running Sockets versions less than 2.2.	GROUP   GLOBAL	GROUP
CMLSize	Specify whether the GETMAIN is performed in a common memory pool <i>below</i> the 16 MB line and, if so, its size in bytes. Valid values are "0" (required application memory is not obtained in a common memory pool, but in the class 6 memory of the system) or the size (in bytes) of user storage to allow in a common memory pool ( <i>below</i> the 16 MB line) of the specified size, rounded up to the next megabyte. This parameter is valid only in BS2000 environments running Sockets versions less than 2.2.	0   <i>bytesize</i>	0
CMScope	Control access to the GETMAIN common memory pool <i>above</i> the 16 MB line. Valid values are "GROUP" (the pool can be accessed only by other tasks using the same user ID) and "GLOBAL" (the pool can be accessed by all user IDs in the system). This parameter is valid only in BS2000 environments.	GROUP   GLOBAL	GROUP
CMSize	Specify whether the GETMAIN is performed in a common memory pool <i>above</i> the 16 MB line and, if so, its size in bytes. Valid values are "0" (required application memory is not obtained in a common memory pool, but in the class 6 memory of the system) or the size (in bytes) of user storage to allow in a common memory pool ( <i>above</i> the 16 MB line) of the specified size, rounded up to the next megabyte. This parameter is valid only in BS2000 environments running Sockets versions less than 2.2.	0   <i>bytesize</i>	0
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 <sub>xxx</sub> and CDATE <sub>xxx</sub> ADARUN parameters. Valid values are "DATASPACE" (dataspace), "EXTENDED"	DATASPACE   EXTENDED   HIPERSPACE   VIRTUAL64	---

Parameter	Use to	Values	Default
	<p>(extended memory), "HIPERSPACE" (hiperspace), or "VIRTUAL64" (virtual 64-bit storage). "EXTENDED" is the only option available on BS2000 RISC machines; "HIPERSPACE" and "VIRTUAL64" are only valid in z/OS environments.</p> <p>This parameter is mutually exclusive with the <code>CFILE</code> and <code>CXFILE</code> parameters.</p> <p>(Adabas Caching Facility)</p>		
CT	Specify the maximum time limit (in seconds) for interregion communication of results from Adabas to the user.	1 - 16777215	60
CWORKStorage	<p>Identify the type of cache space used for Adabas Caching Facility work areas <code>CWORK2FAC</code> and <code>CWORK3FAC</code>. Valid values are "DATASPACE" (dataspace), "EXTENDED" (extended memory), "HIPERSPACE" (hiperspace), or "VIRTUAL64" (virtual 64-bit storage). "EXTENDED" is the only option available on BS2000 RISC machines; "HIPERSPACE" and "VIRTUAL64" are only valid in z/OS environments.</p> <p>(Adabas Caching Facility)</p>	DATASPACE   EXTENDED   HIPERSPACE   VIRTUAL64	---
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	<p>Select a file or range of files to be excluded from caching. Specify this parameter using the following syntax:</p> <pre>CXFILE=({ <i>fnr</i>   <i>fnr1</i>-<i>fnrx</i>}[ , , <i>scope</i>])</pre> <p>where:</p> <ul style="list-style-type: none"> <li>■ <i>fnr</i>, <i>fnr1</i>, <i>fnrx</i>: valid file numbers to be excluded. When a range is specified, the lower file number must be specified first (<i>fnr1</i>).</li> <li>■ <i>scope</i>: 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.</li> </ul>	<p><i>fnr</i>, <i>fnr1</i>, <i>fnrx</i>: 0 - n</p> <p><i>scope</i>: A   D   B</p>	<p><i>fnr</i>, <i>fnr1</i>, <i>fnrx</i>: ---</p> <p><i>scope</i>: B</p>



Parameter	Use to	Values	Default
	This parameter is mutually exclusive with the CASSOxxx, CDATAxxx, and CSTORAGE ADARUN parameters.  (Adabas Caching Facility)		
Datacache	Control whether caching control is enabled (YES) or disabled (NO) for the Data Storage data set. This parameter is valid only in z/OS and z/VM environments.	YES   NO	YES
DBid	Specify the physical database ID of the database.  <b>Note:</b> 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 platform are listed in <i>Adabas Device Types and Blocks Sizes</i> .	<i>devtype</i>	2000 (BS2000)  3390 (z/OS, z/VS, z/VM)
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)	<i>exitname</i>	---
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
DUALCLD	Specify the device type to be used for dual command logging. Valid device types for each platform are listed in <i>Adabas Device Types and Blocks Sizes</i> .	<i>devtype</i>	value of the DEV parameter
DUALCLS	Specify the number of blocks available for each dual command log.	16 - 16777215	---

Parameter	Use to	Values	Default
DUALPLD	Specify the device type to be used for dual protection logging. Valid device types for each platform are listed in <i>Adabas Device Types and Blocks Sizes</i> .	<i>devtype</i>	value of the DEVICE 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
EXcpvr	Indicate whether EXCP or EXCPVR should be used when running APF-authorized. This parameter is valid only in z/OS environments.	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  <ul style="list-style-type: none"> <li>■ 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.</li> <li>■ If ASYTVS=NO, you can use FMXIO to set a limit on the number of I/O operations to be started in parallel overall.</li> </ul>	1 - 16 (ASYTVS=YES)	1 (ASYTVS=YES)
		1 - 100 (ASYTVS=NO)	60 (ASYTVS=NO)
F0rce	Indicate whether the nucleus or Adabas Review hub can overwrite an existing ID table entry.	YES   NO	NO
GRoups	Indicate whether BS2000 interprocess communication is limited to users with the same logon ID. This parameter is available only in BS2000 environments.	YES   NO	NO
HEXnn	Specify the hyperdescriptor exit(s) and associated user routine to be used by the nucleus. Specify this parameter using the following syntax:  <div style="background-color: #e0e0e0; padding: 2px;"><code>HEXnn=user-routine-name</code></div> where:  <ul style="list-style-type: none"> <li>■ <i>nn</i>: the number of the hyperdescriptor exit</li> </ul>	<ul style="list-style-type: none"> <li>■ <i>nn</i>: 01 - 31</li> <li>■ <i>user-routine-name</i>: text (up to eight characters)</li> </ul>	<ul style="list-style-type: none"> <li>■ <i>nn</i>: ---</li> <li>■ <i>user-routine-name</i>: ---</li> </ul>

Parameter	Use to	Values	Default
	<ul style="list-style-type: none"> <li><i>user-routine-name</i>: the name (up to eight characters) of the user routine that gets control for the associated hyperdescriptor exit.</li> </ul>		
IDTName	Define the name for an alternate ID table, resulting in an alternate Adabas environment. This parameter is valid only in BS2000 environments. Valid names must begin with the letters "ADA" and must be eight characters long.	ADAxxxxx	ADABAS5B
IDTPsup	Suppress the IDT name in Adabas output messages. This parameter is valid only in BS2000 environments.	YES   NO	NO
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
INTnas	Specify the time interval (in units of 1.048576 seconds) between SYNS 60 checkpoints.	1 - 16777215	3600 (approximately one hour)
LBP	Specify the maximum number of bytes to be used for the Adabas buffer pool during a session.	80000 - 2147483647	350000
LCP	Specify the maximum number of bytes to be used for the Adabas security pool during a session.	2000 - 16777215	10000
LDEuqp	Specify the number of bytes to be used for the unique (UQ) descriptor pool during a session.	0 - 16777215	5000
LDTp	<p>Define the length of the distributed transaction processing area (Work part 4) or the index of Work part 4:</p> <p>If you have Adabas Transaction Manager Version 7.4 or earlier installed, this parameter defines the length of the distributed transaction processing area (Work part 4) for use with DTP=RM.</p> <p>If you have Adabas Transaction Manager Version 7.5 or later 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).</p> <p>(Adabas Transaction Manager)</p>	<p>Adabas Transaction Manager 7.4 or earlier: 0 - 65535</p> <p>Adabas Transaction Manager 7.5 or later: 8 - one eighth of the WORK4 size</p>	<p>Adabas Transaction Manager 7.4 or earlier: 0</p> <p>Adabas Transaction Manager 7.5 or later: 1</p>
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 asynchronous 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	0   1   80000 - 2147483647	0

Parameter	Use to	Values	Default
	(with a minimum of 80000 allowed) enables asynchronous buffer flush operations and defines the size of the pool directly (in bytes).		
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 - 16777215	10000
LNKGNAME	The name of the link globals table to be employed by an Adabas 8 batch/TSO link routine.  <b>Note:</b> If you specify the LNKGNAME parameter, you must also specify the ADARUN DBID and SVC parameters.	valid module name for an IBM z/OS, z/VSE or z/VM operating system, or for a BS2000 module	LNKGBLS
LOCa1	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 dependencies of the various logging parameters, please review the detailed description.	YES   NO	NO
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 LOGxxxx 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

Parameter	Use to	Values	Default
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 parame
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
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 - 16777215	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	value of the LFI ADARUN parame
LRP1	Specify the size of the Adabas or Event Replicator replication pools, in bytes. This parameter should be specified for only: <ul style="list-style-type: none"> <li>■ an Adabas nucleus that also has REPLICATION=YES set.</li> <li>■ an Event Replicator database</li> </ul> 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. The maximum value for this parameter must be less than or equal to the LWP setting minus 19968 divided by two ( $LS \leq (LWP - 19968) / 2$ ).	19968 - *	49920

Parameter	Use to	Values	Default
LU	Specify the length (in bytes) of the intermediate user buffer area. The range of values varies based on the operating system on which Adabas is installed..  On z/OS and z/VM systems, LU cannot exceed a value greater than that produced by this calculation:  $(NABvalue \times 4096)$  On z/VSE and BS2000 systems, LU cannot exceed a value greater than that produced by this calculation:  $(NABvalue \times 4096) - 256$	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:  $(LS-value + 19968) \times 2$  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
M0de	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 DD/DRUCK data set.	Upper   Mixed	Mixed
MSGPrint	Specify the case of Adabas messages sent to the DD/PRINT data set.	Upper   Mixed	Mixed
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.

Parameter	Use to	Values	Default
MXCANCELWarn	<p>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.</p> <p>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.</p> <p>Adabas Cluster Services, Adabas Parallel Services</p>	0   4 - (MXCANCEL_value - 1)	The value of the MXCANCEL parameter divided by 4.
MXMSG	<p>Specify the timeout threshold, in seconds, in which the processing of an intracluster command must occur.</p> <p>Adabas Cluster Services, Adabas Parallel Services</p>	15 - 32767	300
MXMSGWarn	<p>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.</p> <p>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.</p> <p>Adabas Cluster Services, Adabas Parallel Services</p>	0   4 - (MXMSG_value - 1)	The value of the parameter divided by 4.
MXStatus	<p>Specify the interval, in seconds, at which the member-level XCF status monitoring heartbeat should occur.</p> <p>A value of "0" indicates that no member-level status monitoring should occur.</p> <p>Adabas Cluster Services</p>	0   15 - 21474836	15
MXTNa	<p>Set the maximum non-activity time, in seconds, that can be specified in the OP command for an individual user.</p>	1 - 65535	3600
MXTSx	<p>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.</p>	1 - 65535	3600
MXTT	<p>Set the maximum transaction time limit, in seconds, that can be specified in the OP command for an individual user.</p>	1 - 65535	3600

Parameter	Use to	Values	Default
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
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
NIshq	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 smaller of 65535 or the value of the NH parameter divided by 4. 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 - 65535   NH/4 (whichever is smaller)	20   (NH/NU) * 2 (whichever is larger)
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
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



Parameter	Use to	Values	Default
NT	Specify the number of threads to be used during the Adabas session.	3 - 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
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 on z/OS systems. When this parameter is set to <i>YES</i> , 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 <i>Adabas Device Types and Block Sizes</i>	value of the DEV 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
PLOGSEgds	Indicate whether all portions of a modified spanned storage record should be written to the protection log (PLOG).	YES   NO	NO
PLOGSIze	Specify the number of blocks available for each protection log in multiple protection log data sets. This parameter must be specified with the PLOGDEV parameter.	16 - 16777215	---
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	L1 - L9	---

Parameter	Use to	Values	Default
	<p>parameter. If you want several commands to be prefetched or multifetched for a single file, repeat the same file number in the PREFIFIL parameter.</p> <p>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.</p>		
PREFIFil	<p>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.</p> <p>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.</p>	1 - 5000	---
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
PREFSb1	Specify the size, in bytes, of the buffer for a single prefetch, which determines how much record data is prefetched.	1024 - 65535	3000
PREFTb1	Specify the size, in bytes, of the total prefetch buffer area.	6144 - 16777215	30000
PREFXCmd	<p>Identify the commands that should be <i>excluded</i> in prefetch or multifetch processing.</p> <p>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</p>	L1 - L9	---

Parameter	Use to	Values	Default
	<p>prefetched or multifetched for a single file, repeat the same file number in the PREFXFIL parameter.</p> <p>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.</p>		
PREFXFil	<p>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.</p> <p>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.</p>	1 - 5000	---
PROGram	Identify the program to be run.	ADACOM   ADANUC   ADAREV   NETWRK   RENTUSER   USER   <i>utility-name</i>	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
REPllication	<p>Indicate whether replication should be activated for the Adabas nucleus.</p> <p>(Event Replicator for Adabas)</p>	YES   NO	NO

Parameter	Use to	Values	Default
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   <i>hubid</i>	NO
REVLOGBmax	Specify the maximum allowable number of bytes of a logged buffer for Adabas Review.  (Adabas Review)	0 - 30000	5120
REVLOGMax	Specify the maximum size of all of the logged buffers allowed for an Adabas Review command.  (Adabas Review)	2000 - 32768 (32K)	16384
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 override any duplicates with the Replicator system file), or not at all (NONE).  (Event Replicator for Adabas)	BOTH   FILE   NONE   PARMS	If the Replicator system file resides on the Event Replicator database, definitions are read first from the Replicator system file and then from DDKARTE. If the Replicator system file does not reside on the Event Replicator database, the definitions are read 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

Parameter	Use to	Values	Default
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
SMgt	Indicate whether the error handling and message buffering facility should be enabled for the session.  This parameter is not available on BS2000 systems.	YES   NO	NO
Sortcache	Indicate whether to enable (YES) or disable (NO) controller caching for the Adabas Sort data set. This parameter is valid only in z/OS and z/VM environments.	YES   NO	YES
Spt	Indicate use of Adabas triggers and stored procedures should be enabled for the session.	YES   NO	NO
Submpsz	Specify the common memory pool size, in bytes, for subtask communication in products such as Adabas Parallel Services, Adabas Review, and Event Replicator for Adabas. This parameter is valid only in BS2000 environments.	100000 - *	1024000
Svc	Specify the Adabas SVC number or Adabas Review hub SVC number to be used for the session.	z/OS: 200 - 255  z/VSE: any free SVC value, although 45 is recommended.	z/OS: 249  z/VSE: 45
TAPere1	Specify how end-of-file (EOF) processing is handled for tapes and cartridges. This parameter is valid only in BS2000 environments.	KEEPUNL   KEEP   NO   RELEASE   UNLOAD	KEEPUNL
TARgetid	Specify the unique Entire Net-Work target ID for this node.	1 - 65535	1
TASKctgy	Identify the Adabas task category. This parameter is valid only in BS2000 environments.	BATCH   TP	BATCH

Parameter	Use to	Values	Default
TCPIp	Indicate whether a direct TCP/IP link to the Adabas nucleus should be activated for this session. This parameter is valid only in z/OS environments.	YES   NO	NO
TCPUr1	Identify the universal resource locator (URL) for the direct TCP/IP link to the Adabas nucleus (when TCPIP is set to "YES"). This parameter is valid only in z/OS environments.	<i>url</i>	---
TEmpcache	Indicate whether to enable (YES) or disable (NO) controller caching for the Adabas Temp data set. This parameter is valid only in z/OS and z/VM environments.	YES   NO	YES
TFlush	Specify the time, in seconds, to allow for a synchronous buffer flush.	1 - 16777215	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 MSTSX 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)	<i>prefix</i>	TM
TNA A	Specify the maximum elapsed time, in seconds, that an access-only user may be considered active without issuing an Adabas command.	1 - 16777215	900
TNA E	Specify the maximum elapsed time, in seconds, that an ET logic user may be considered active without issuing an Adabas command.	1 - 16777215	900
TNA X	Specify the maximum elapsed time, in seconds, that an exclusive control user may be considered active without issuing an Adabas command.	1 - 16777215	900
TT	Specify the maximum elapsed time, in seconds, permitted for a logical transaction issued by an ET logic user.	1 - 16777215	900
UExnn	Activate a user exit and specify the name of the user routine for the exit. The value of <i>nn</i> 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.  <b>Note:</b> User exit 2 (UEx2) and user exit 12 (UEx12) are mutually exclusive.	<i>user-routine-name</i>	---
UTonly	Indicate whether the session should be restricted to Adabas utilities only.	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. This parameter is valid only in z/OS and z/VM environments.	YES   NO	YES

## AOSLOG : 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 : 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". Software AG recommends that you restore excluded files from the latest save tape 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 : Activate Automatic Restart Management (ARM) Program

Parameter	Specify . . .	Possible Values	Default
ARMNAME	and activate a particular automatic restart management (ARM) program.	<i>arm-pgm-name</i>	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
```

## ASSOCACHE : Associator Controller Caching

This parameter applies to the operating systems z/OS and z/VM only.

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
```

## ASYTVS : Asynchronous Bufferflush

---

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

 **Note:** When an SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

**Example**

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

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

## CACHE : 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 <code>CACHE=YES</code> .

### Example

```
ADARUN CACHE=YES
```

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

## CACTIVATE : 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 <code>CDEMAND</code> threshold level.
NO	None of the RABN ranges is activated unless: <ul style="list-style-type: none"> <li>■ the <code>CDEMAND</code> threshold level is reached; or</li> <li>■ the <code>CENABLE</code> operator command is accepted by ADACSH.</li> </ul>

### 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 / CASSOL64 / CASSOV64 : Adabas Caching Facility Associator RABN Cache Space Control

 **Note:** CASSOxxx parameters are mutually exclusive with the CFILE and CXFILE parameters.

The CASSOxxx parameters identify one or more Associator RABNs to be cached where xxx indicates the type of cache space:

- CASSODSP (or CASSOD) for data space
- CASSOEXT (or CASSOE) for extended memory
- CASSOHSP (or CASSOH) for hiperspace
- CASSOL64 (or CASSOL) for virtual 64 backed by large pages
- CASSOV64 (or CASSOV) for virtual 64

 **Note:** Virtual 64 storage is only available in z/OS environments.

**{ CASSODSP | CASSOEXT | CASSOHSP | CASSOL64 | 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 CASSOxxx parameters must be specified.

If CASSOxxx 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 CASSOxxx parameter are ignored. The ignored parameters can be added after initialization by using an operator command.
2. The CSTORAGE parameter overrides any CASSOxxx 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 : 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
<i>n</i>	<p>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).</p> <p>For extended memory, data spaces, and hiperspaces:</p> <ul style="list-style-type: none"> <li>■ the minimum unit size is 81920 bytes (or 80K);</li> <li>■ the maximum unit size is 2147475456 bytes (or 2097144K; or 2047M); and</li> <li>■ the default unit size is 1048576 bytes (or 1024K; or 1M).</li> <li>■ In these cases, CASSOMAXS is always rounded up to the nearest 4-kilobyte boundary.</li> </ul> <p>For Virtual 64 memory and Virtual 64 backed by large pages, virtual storage above the 2GB 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:</p> <ul style="list-style-type: none"> <li>■ the minimum unit size and the default unit size is 1M (1048576 bytes or 1024K).</li> </ul>

Value	Description
	<ul style="list-style-type: none"> <li>■ the maximum unit size depends on the limit set for your installation.</li> <li>■ For memory objects, CASSOMAXS is rounded up to the nearest megabyte boundary.</li> </ul>

The total space available is equal to:

Maximum  
ASSO cache space size = CASSOMAXS · 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 14M).

## CBUFNO: 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 3380 Associator, a reduction factor of 18:1 may be achieved.

**CBUFNO = *n***

Value	Description
<i>n</i>	<p>The number of read-ahead buffers available for concurrent I/O processing.</p> <p>The recommended setting is a value that accommodates the expected number of concurrent L2, L3, and L9 commands.</p> <p>The minimum number of read-ahead buffers is 0 (the default). If CBUFNO=0 is specified, <i>no</i> read-ahead caching is performed.</p> <p>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.</p>

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 (nondesorptor search) and L9 (histogram) commands.

## CCTIMEOUT : 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
<i>n</i>	<p>The amount of time (in seconds) a cache space area can remain inactive before it is released back to the system.</p> <p>If demand caching (<code>CDEMAND</code>) is in effect, <code>CCTIMEOUT</code> specifies the number of seconds a RABN range, file, or file range can remain inactive before it is disabled.</p> <p>The minimum number of seconds is 60; the maximum is 2,147,483,647.</p> <p>The default number of seconds is 7200 (2 hours).</p>

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 re-allocated 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 / CDATAL64 / CDATAV64 : Adabas Caching Facility Data Storage RABN Cache Space Control



**Note:** CDATA<sub>xxx</sub> parameters are mutually exclusive with the CFILE and CXFILE parameters.

The CDATA<sub>xxx</sub> parameters identify one or more Data Storage RABNs to be cached where <sub>xxx</sub> indicates the type of cache space:

- CDATADSP (or CDATAD) for data space
- CDATAEXT (or CDATAE) for extended memory
- CDATAHSP (or CDATAH) for hiperspace
- CDATAL64 (or CDATAL) for virtual 64 backed by large pages
- CDATAV64 (or CDATAV) for virtual 64



**Note:** Only extended memory is available for BS2000 RISC machines. Hiperspace is an option only in z/OS environments. Virtual 64 storage is available only in z/OS environments.

**{ CDATADSP | CDATAEXT | CDATAHSP | CDATAL64 | 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 <sub>xxx</sub> parameters must be specified.

If CDATA<sub>xxx</sub> 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<sub>xxx</sub> parameters, subsequent RABN ranges for the CDATA<sub>xxx</sub> parameters will be ignored. The ignored parameters can be added after initialization by using an operator command.

2. The `CSTORAGE` parameter overrides any `CDATAxxx` 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 : 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
<i>n</i>	<p>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).</p> <p>For extended memory, data spaces, and hiperspaces:</p> <ul style="list-style-type: none"> <li>■ the minimum unit size is 81920 bytes (or 80K);</li> <li>■ the maximum unit size is 2147475456 bytes (or 2097144K; or 2047M); and</li> <li>■ the default unit size is 1048576 bytes (or 1024K; or 1M).</li> <li>■ In these cases, <code>CDATAMAXS</code> is always rounded up to the nearest 4-kilobyte boundary.</li> </ul> <p>For Virtual 64 memory and Virtual 64 backed by large pages, virtual storage above the 2GB 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, <code>CDATAMAXS</code> determines the number of 1-megabyte segments to be allocated for the memory object:</p>

Value	Description
	<ul style="list-style-type: none"> <li>■ the minimum unit size and the default unit size is 1M (1048576 bytes or 1024K).</li> <li>■ the maximum unit size depends on the limit set for your installation.</li> <li>■ For memory objects, CDATAMAXS is rounded up to the nearest megabyte boundary.</li> </ul>

The total space available is equal to:

```
Maximum
DATA cache space size = CDATAMAXS · 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 14M).

## CDEMAND : 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.

$$\text{CDEMAND} = \left\{ \begin{array}{c} 0 \\ n \end{array} \right\}$$

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 2,147,483,647 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 : 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 : Collation Descriptor User Exit

Parameter	Specify . . .	Possible Values	Default
CDXnn	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 User Exits documentation 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 : 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 3380 Associator, a reduction factor of 18:1 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 (nondesoriptor 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 `CXCLUDE=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 `CXCLUDE` and `CINCLUDE` operator commands.

### Example

```
ADARUN CBUFNO=5,CXCLUDE=FINN/HIST
```

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

## CFILE : Adabas Caching Facility File Range



**Note:** `CFILE` is mutually exclusive with the `CASSOxxx`, `CDATAxxx`, 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
<i>fnr</i>	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 dataset.
<i>fnr</i> - <i>fnr</i>	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.
<i>cls</i>	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: <ul style="list-style-type: none"> <li>■ 1: 100% (highest priority)</li> <li>■ 2: 75%</li> <li>■ 3: 50%; the default</li> <li>■ 4: 25%</li> <li>■ 5: 10% (lowest priority)</li> </ul>

Value	Description
<i>scope</i>	<p>Caching scope. Valid values are:</p> <ul style="list-style-type: none"> <li>■ ASSO (or A) : Caching is to be used only for Associator for the file</li> <li>■ DATA (or D) : Caching is to be used only for Data Storage for the file</li> <li>■ BOTH (or B) : Caching is to be used for both Associator and Data Storage for the file</li> </ul> <p>The default is BOTH.</p>
<i>cache</i>	<p>Specifies the type of cache storage in which RABNs for the file will be cached. Valid values are:</p> <ul style="list-style-type: none"> <li>■ DATASPACE (or D)</li> <li>■ EXTENDED (or E)</li> <li>■ HIPERSPACE (or H)</li> <li>■ L64 (or L)</li> <li>■ VIRTUAL64 (or V)</li> </ul> <p>The default is DATASPACE.</p> <p><b>Note:</b> Virtual 64 is a cache option only in z/OS environments.</p>

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: Logged Buffer Size Limit

Parameter	Specify . . .	Possible Values	Default
CLOGBMAX	the maximum allowable number of bytes of a logged buffer.	368 - 4294967295	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 both CLOGLAYOUT=5 and 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 : Multiple Command Log Device Type CLOGSIZE : Multiple Command 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 3380 device for command logging. The size of each command log data set is 1350 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3380 device has 15 tracks, and each track can contain 9 blocks of command log data. Thus, 10 cylinders = 10 x 15 x 9 = 1350 blocks).

```
ADARUN PROG=ADANUC, LOGGING=YES, NLOG=6, CLOGDEV=3380, CLOGSIZE=1350
```

## CLOGLAYOUT : Command Logging Format

Parameter	Specify . . .	Possible Values	Default
CLOGLAYOUT	the command log format.	5   8	5

This parameter specifies the format of the Adabas command log.

### Value Meaning

5 The default setting. User programs designed for Adabas 7 or earlier should specify CLOGLAYOUT=5.

**Note:** CLOGLAYOUT=5 CLOG records may be caused by internal Adabas Review V4 commands. These records should be ignored by users.

8 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. Each of the buffers are written out in much the same way as for CLOGLAYOUT=5, except that each buffer is prefixed by its corresponding Adabas buffer description (ABD). 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.

### Notes:

1. Timestamps in an Adabas 8 command log created using CLOGLAYOUT=8 are stored in machine time (GMT), whereas CLOGLAYOUT=5 timestamps are stored, as always, in local time. 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.

2. Because of the different timestamp formats used in CLOGLAYOUT=5 and CLOGLAYOUT=8 command logs, we do not recommend that you mix or merge command logs created using different CLOGLAYOUT settings (and, in fact, Adabas does not allow this in cluster or parallel services environments). If you do, your results will be unpredictable.

### Example

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

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

## CLOGMAX: 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 - 4294967295	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 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 both CLOGLAYOUT=5 and CLOGLAYOUT=8.

### 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 : 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 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.
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.

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 CLOGMRG=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.

## CLUCACHENAME : 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 Adabus 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:** Software AG recommends 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 Adabus 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 : 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 (z/OS systems only).

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 4-kilobate boundary.

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 Adabas Parallel Services Operations* documentation . You can use the Adabas Online System cache structure size calculator described in *Estimating Sizes for the Cache Structure in a Cluster Environment* in *Adabas Parallel Services Operations* documentation 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: Global Cache Area Storage Type Option

Parameter	Specify . . .	Valid Values	Default
CLUCACHETYPE	the virtual storage type for the global cache area	DSP   V64   L64	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" (available on all supported operating systems) and "V64" or "L64" (available only on z/OS systems).

The size of the global cache is specified in the [CLUCACHESIZE parameter](#).

The default value is "DSP", indicating that a shared dataspace will be used for the global cache. In this case, the maximum cache size is 2 GB.

If "V64" is specified (on z/OS systems), a shared storage area in 64-bit virtual storage will be used for the global cache. In this case, the cache can be made much larger than 2 GB, but is still subject to the MEMLIMIT value in effect for the ADACOM job for the Adabas Parallel Services database.

If "L64" is specified (on z/OS 1.9 or later systems running on IBM z10 machines), the global cache space shared by the cluster nuclei will reside in 64-bit virtual storage that is backed by large pages (provided the large page pool has been configured to a sufficient size and is available in the system).



#### Notes:

1. If one ADACOM job is used to hold the shared cache and lock spaces for multiple Adabas Parallel Services clusters, L64 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 L64 cache spaces, but this is not mandatory.

2. Virtual 64-bit storage backed by large pages can only be used on IBM z10 mainframes running z/OS Version 1.9 or above and for which IBM large page support has been enabled via IBM APAR OA20902. You must also allocate the size of the large page pool (use the LFAREA parameter in the IEASYSxx member of SYS1.PARMLIB). The LFAREA parameter allows you to specify the amount of real storage to be used for 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 type of virtual storage used for the global cache area is specified as "V64".

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

## CLUCACHEUNCHANGED : 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 : Cluster Group Name**

Parameter	Specify . . .	Possible Values	Default
<code>CLUGROUPNAME</code>	the name of the XCF messaging group in Adabus Cluster Services.	text	none

The `CLUGROUPNAME` parameter provides:

- the XCF group name to be used by the sysplex cluster nuclei for their communication using XCF.
- the name to be used by the cluster nuclei for their communication.

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

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 : 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:** Software AG recommends 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 : 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 *Adabas Parallel Services Operations* documentation. You can use the Adabas Online System lock structure size calculator described in *Estimating Sizes for the Lock Structure in a Cluster Environment* in *Adabas Parallel Services Operations* documentation to determine appropriate global lock area sizes.

## CLUSTER : 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. This is supported only for z/OS systems. 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 datasets (DD-name/link name 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
```

## CMADDR and CMLADDR : GETMAIN Memory Pool Start Address

These parameters apply only to BS2000 environments running Sockets versions less than 2.2.

Parameter	Specify . . .	Minimum	Maximum	Default
	the start address of the GETMAIN memory pool			
CMADDR	above the 16MB line	0, or X'1000000'	X'FE000000'	0
CMLADDR	below the 16MB line	0, or X'10000'	X'E00000'	0



**Note:** These parameters should only be used if required by an installation site, in which case the site will provide the needed value.

These parameters are involved with setting all required memory into a common memory pool for use by associated tasks on BS2000 above (CMADDR) or below (CMLADDR) the 16 megabyte line. They set the start address of this memory pool.

If the default value ("0") is used for a CMADDR or CMLADDR parameter, the start address depends on the value of the corresponding CMSIZE or CMLSIZE parameter:

- If the CMADDR or CMLADDR parameter value is "0" and the corresponding CMSIZE or CMLSIZE parameter is *not* "0", the operating system will set the start address at the next available megabyte boundary.
- If the CMADDR or CMLADDR parameter value is "0" and the corresponding CMSIZE or CMLSIZE parameter is *also* "0", required memory will not be obtained in a common memory pool but rather in class 6 memory.

## Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the value of the CMADDR parameter to X'01200000'.

### Examples

The following example allows for three (3) megabytes of user storage in the address range X'2000000' to X'4FFFFFF':

```
ADARUN PROG=ADANUC,CMADDR=2000000,CMSIZE=2500000
```

The following example allows for three (3) megabytes of user storage below 16 megabytes in the address range X'200000 to X'4FFFFFF':

```
ADARUN PROG=ADANUC,CMLADDR=200000,CMSIZE=25000000
```

## CMAXCSPS : 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
<i>n</i>	<p>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.</p> <p>The minimum value permitted is 1; the maximum is 16, the default is 8.</p> <p>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.</p> <p><b>Note:</b> 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.</p>

### 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.

## CMDQMODE : Command Queue Mode

This parameter applies to the BS2000 operating system only.

Parameter	Specify . . .	Possible Values	Default
<code>CMDQMODE</code>	whether to allocate the command queue memory pool below or above the 16-MB line.	BELOW   ABOVE	ABOVE (BELOW for Adabas versions prior to Version 8)

`CMDQMODE` specifies whether to allocate the BS2000 memory pool for the Adabas command queue below or above the 16-MB line.

Value	Meaning
BELOW	The default setting. Places the BS2000 memory pool for the Adabas command queue below the 16-MB line in one or more 64-kilobyte segments.
ABOVE	Places the BS2000 memory pool for the Adabas command queue above the 16-MB line in one or more 1-MB segments.

### Example

The following example, places the Adabas command queue memory pool above the 16-MB line in 1-MB segments.

```
ADARUN PROG=ADANUC ,CMDQMODE=ABOVE
```

## CMFIX or CMLFIX : GETMAIN Memory Pool Fixed Location

These parameters apply only to BS2000 environments running Sockets versions less than 2.2.

Parameter	Specify . . .	Possible Values	Default
	whether the GETMAIN memory pool must have a fixed location or not		
<code>CMFIX</code>	above the 16MB line	YES   NO	NO
<code>CMLFIX</code>	below the 16MB line	YES   NO	NO



**Note:** Use these parameters only if required by an installation site, in which case the site will provide the needed value.

The `..FIX` parameter is ignored if the corresponding `..SIZE` parameter value is 0.

These parameters are involved with setting all required memory into a common memory pool for use by associated tasks on BS2000 above (CMFIX) or below (CMLFIX) the 16 megabyte line. This parameter specifies whether the memory pool is at a fixed location or not:

**Value** The application's GETMAIN memory pool . . .

NO need not be located at the same start address in the subtask as in the main task.

YES must be located at the same start address in the subtask as in the main task.



**Note:** A subtask depends on the application. An example of this is Adabas Review in local mode.

### Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the values of both the CMLFIX and CMLFIX parameters to "YES".

### Examples

The following example allows for three (3) megabytes of user storage above 16 megabytes in the address range X'2000000' to X'4FFFFFFF'. This address range will also be used by the subtask:

```
ADARUN PROG=ADANUC,CMADDR=2000000,CMSIZE=2500000,CMFIX=YES
```

The following example allows for three (3) megabytes of user storage below 16 megabytes in the address range X'200000' to X'4FFFFFF'. This address range will also be used by the subtask:

```
ADARUN PROG=ADANUC,CMLADDR=200000,CMSIZE=2500000,CMFIX=YES
```

## CMLxxxx Parameters

The discussion of these parameters is paired with related Adabas parameters as follows:

[CMLADDR](#)

[CMLFIX](#)

[CMLSCOPE](#)

[CMLSIZE](#)

## CMSCOPE or CMLSCOPE : GETMAIN Memory Pool Scope

These parameters apply only to BS2000 environments running Sockets versions less than 2.2.

Parameter	Specify . . .	Possible Values	Default
	access to the GETMAIN memory pool		
CMSCOPE	above the 16MB line	GROUP   GLOBAL	GROUP
CMLSCOPE	below the 16MB line	GROUP   GLOBAL	GROUP



**Note:** These parameters should only be used if required by an installation site, in which case the site will provide the needed value.

The ..SCOPE parameter is ignored if the corresponding ..SIZE parameter value is 0.

These parameters are involved with setting all required memory into a common memory pool for use by associated tasks on BS2000 above (CMSCOPE) or below (CMLSCOPE) the 16 megabyte line. This parameter specifies accessibility to the memory pool:

Value	Meaning
GROUP	The GETMAIN common memory pool can only be accessed by other tasks using the same user ID.
GLOBAL	The GETMAIN common memory pool can only be accessed by all user IDs in the system.

### Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the values of the CMSCOPE and CMLSCOPE parameters to "GROUP".

### Examples

The following example allows for three (3) megabytes of user storage at the next available megabyte boundary above the 16MB line accessible to tasks with the same user ID.

```
ADARUN PROG=ADANUC ,CMSIZE=2500000 ,CMSCOPE=GROUP
```

The following example allows for three (3) megabytes of user storage at the next available megabyte boundary below the 16MB line accessible to tasks with the same user ID.

```
ADARUN PROG=ADANUC ,CMLSIZE=2500000 ,CMLSCOPE=GROUP
```



## CMSIZE or CMLSIZE : GETMAIN Memory Pool Use and Size

These parameters apply only to BS2000 environments running Sockets versions less than 2.2.

Parameter	Specify . . .	Minimum	Maximum	Default
	whether the GETMAIN is performed in a common memory pool and if so, its size in bytes (decimal)			
CMSIZE	above the 16MB line	0	address-limit	0
CMLSIZE	below the 16MB line	0	address-limit	0



**Note:** These parameters should only be used if required by an installation site, in which case the site will provide the needed value.

These parameters are involved with setting all required memory into a common memory pool for use by associated tasks on BS2000 above (CMSIZE) or below (CMLSIZE) the 16 megabyte line. These parameters determine whether a common memory pool is used and if so, its size:

- If the ..SIZE parameter value is 0, the default, the required application memory is obtained in the class 6 memory of the system.
- If the ..SIZE parameter value is *not* 0, the required application memory is obtained in a common memory pool of the specified size rounded up to the next megabyte.

### Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the CMSIZE parameter to "32000000" and the CMLSIZE parameter to "4096000".

### Examples

The following example allows for three (3) megabytes of user storage at the next available megabyte boundary above the 16 megabyte line accessible to tasks with the same user ID.

```
ADARUN PROG=ADANUC,CMSIZE=2500000
```

The following example allows for three (3) megabytes of user storage at the next available megabyte boundary below the 16 megabyte line accessible to tasks with the same user ID.

```
ADARUN PROG=ADANUC,CMLSIZE=2500000
```

## CRETRY : 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
<i>n</i>	<p>The number of seconds between attempts to acquire an area of cache space if the previous <code>GETMAIN</code> or space allocation failed.</p> <p>The minimum number of seconds is 60; the maximum is 2147483647.</p> <p>The default number of seconds is 900 (15 minutes).</p>

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 : 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` and/or `CDATAxxx` 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 | L64 | VIRTUAL64 }**

Value	Description
DATASPACE	Data Space
EXTENDED	Extended Memory
HIPERSPACE	Hiperspace
L64	Virtual 64 storage backed by large pages
VIRTUAL64	Virtual 64 Storage

Extended memory is the only option available on BS2000 RISC machines. Hiperspace is an option only in z/OS environments. Virtual 64 storage is available only in z/OS environments.

### 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 : Command Timeout Limit

---

Parameter	Specify . . .	Minimum	Maximum	Default
CT	the maximum time (seconds) for interregion communication of results from Adabas to the user.	1	16777215	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 cancelled;
- 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 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 : 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 | L64 | VIRTUAL64 }**

Value	Description
DATASPACE	Data Space
EXTENDED	Extended Memory
HIPERSPACE	Hiperspace
L64	Virtual 64 storage backed by large pages
VIRTUAL64	Virtual 64 Storage

Extended memory is the only option available on BS2000 RISC machines. Hiperspace is an option only in z/OS environments. Virtual 64 storage is available only in z/OS environments.

If CWORKSTORAGE is not specified, but the CWORK2FAC and/or CWORK3FAC parameters are specified, then the default CWORKSTORAGE is as follows:

- DATASPACE for BS2000 (except RISC machines), z/OS, z/VSE, z/VM
- EXTENDED for BS2000 RISC machines

### Example

```
ADARUN CWORKSTORAGE=HIPERSPACE
```

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

## CWORK2FAC : Adabas Caching Facility Work Part 2 Percentage

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

Value	Description
<i>n</i>	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.

Each 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 · blksize ) + 4095 ) · 4096 ) + 4096
```

- where

```
#RABNs = int ( WORK part 2 size · CWORK2FAC )
```

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 3380, the cache space size required is

```
cache size = int ( ( ( #RABNs · blksize ) + 4095 ) ÷ 4096 ) · 4096
             = int ( ( ( 500 · 5492 ) + 4095 ) ÷ 4096 ) · 4096
             = 2,750,095
```

## CWORK3FAC : Adabas Caching Facility Work Part 3 Percentage

---

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

Value	Description
<i>n</i>	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.

Each 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 · blksize ) + 4095 ) ÷ 4096 ) · 4096
```

- where

```
#RABNs = int ( WORK part 3 size · CWORK3FAC )
```

If the specified `CWORK3FAC` cannot accommodate the working set of Work part 3 RABNs, the RABNs beyond the computed `CWORK3FAC` are not cached. Normal I/O operations are performed for the non-cached Work part 3 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 CWORK3FAC=50
```

ADACSH caches 50 percent of Work part 3, beginning from the first block 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 3380, the cache space size required is

```
cache size = int ( ( ( #RABNs · blksize ) + 4095 ) ÷ 4096 ) · 4096
            = int ( ( ( 500 5492 ) + 4095 ) ÷ 4096 ) · 4096
            = 2,750,095
```

## CXFILE : Adabas Caching Facility Excluded Files

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



**Note:** `CXFILE` is mutually exclusive with `CASS0xxx`, `CDATAxxx`, and `CSTORAGE` parameters. If any of these parameters is used with `CXFILE`, they will be ignored.

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

Value	Description
<i>fnr</i>	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 dataset.
<i>fnr</i> - <i>fnr</i>	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.
<i>scope</i>	Specifies the scope of exclusion: <ul style="list-style-type: none"> <li>■ ASSO (or A): Only Associator RABNs are to be excluded</li> <li>■ DATA (or D): Only Data Storage RABNs are to be excluded</li> <li>■ BOTH (or B): Both Associator and Data Storage RABNs are to be excluded</li> </ul>

Value	Description
	If the scope term is specified, it must be preceded by two commas to maintain its position within the parentheses. See the <a href="#">CFILE parameter</a> for more information about the positional terms used within the parentheses. The <i>cls</i> and <i>cache</i> terms are not used for the CXFILE parameter but the position of the <i>scope</i> 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 : Data Storage Controller Caching

This parameter applies to the operating systems z/OS and z/VM only.

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 pre-read and maintained as long as possible in the caching facilities of the disk controller.

NO Data Storage blocks are not pre-read.

### 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
```

## DBID : 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 : Device Type

Parameter	Specify . . .	Possible Values	Default
DEVICE	the device type on which the first block of the Associator is stored.	see text	2000 (BS2000) 3390 (other)

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.

Some device type designations have been reserved for special use such as VSAM data set support on z/OS systems. The following is a partial list of reserved device types and their uses:

Device	Use
5555	VSAM DD/xxxxR5
6666	VSAM DD/xxxxR4
7777	VSAM DD/xxxxR3
8888	VSAM DD/xxxxR2
9999	VSAM DD/xxxxR1 - (default for self-defined VSAM block sizes)

For more information about VSAM device usage, see *Device and File Considerations* in the Adabas Installation documentation.

### 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 : 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 *Adabas Parallel Services Operations* or *Adabas Cluster Services Operations* documentation to determine appropriate global cache area sizes and DIRRATIO/ELEMENRATIO 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 : 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 tapes 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 : 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 : 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

**Value Meaning**

If you have Adabas Transaction Manager Version 7.4 or earlier installed and if DTP=RM, the LDTP parameter specifies the size of the Work part 4 area that the nucleus uses to keep information about distributed transactions.

If you have Adabas Transaction Manager Version 7.5 or later 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
```

## DUALCLD : Dual Command Log Device Type DUALCLS : Dual Command 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 3380 device for dual command logging. The size of each command log data set is 1350 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3380 device has 15 tracks, and each track can contain 9 blocks of command log data. Thus, 10 cylinders =  $10 \times 15 \times 9 = 1350$  blocks).

```
ADARUN PROG=ADANUC, LOGGING=YES, DUALCLD=3380, DUALCLS=1350
```

## DUALPLD : Dual Protection Log Device Type DUALPLS : Dual Protection 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 3380 device; the size of each dual log data set is 1200 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3380 device has 15 tracks, and each track contains 8 blocks of protection log data. Thus, 10 cylinders =  $10 * 15 * 8 = 1200$  blocks).

```
ADARUN PROG=ADANUC, DUALPLD=3380, DUALPLS=1200
```

## EXCPVR : Control EXCP or EXCPVR Use

Parameter	Specify . . .	Possible Values	Default
EXCPVR	whether EXCP or EXCPVR should be used when running APF-authorized, in z/OS environments.	YES   NO	YES

The EXCPVR parameter controls whether EXCP or EXCPVR should be used when from an APF-authorized load library in z/OS environments. 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 : 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 *Adabas DBA Tasks*.

### Example

The following example runs Adabas Fastpath with Adabas.

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

## FMXIO : 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.



**Note:** When an SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

### 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.

### 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 : 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.

### 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
```

## GROUPS : User Group Interprocess Communication

This parameter applies to the BS2000 operating system only.

Parameter	Specify . . .	Possible Values	Default
GROUPS	whether BS2000 interprocess communication is limited to users with the same logon ID.	YES   NO	NO

This parameter limits the accessibility of ADARER and its ID table (IDT) to a group of users with the same logon ID.

### Value Meaning

**YES** Allows the user(s) access to the IDT that starts under the same logon ID as the Adabas nucleus. When initialized, the IDT is set to GROUPS=YES; all users (nuclei, utilities, Entire Net-work, and all user tasks) must therefore also specify GROUPS=YES in either the ADARUN statement or using the Adalink parameters.

**NO** The default setting. All users on one machine have access to this IDT, even if they are logged on under a different logon ID.



**Note:** More than one IDT with the same name can be used if one is set to GROUPS=NO (the default) and the rest specify GROUPS=YES.

### Example

The following example allows BS2000 users access to the IDT that starts under the same logon ID as this Adabas nucleus.

```
ADARUN PROG=ADANUC, DBID=11, GROUPS=YES
```

## HEXnn :Hyperdescriptor User Exit

Parameter	Specify . . .	Possible Values	Default
HEXnn	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 User Exits documentation 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
```

## IDTNAME : Define ID Table Name

This parameter applies to the BS2000 operating system only.

Parameter	Specify . . .	Possible Values	Default
IDTNAME	an alternate ID table.	ADACCCC	ADABAS5B

Defines a name for an (alternate) Adabas environment. The name must be eight characters long and must begin with the characters "ADA".



**Note:** All users of the new ID table must include the IDTNAME in either the ADARUN statement or ADALNK parameters.

## Specific Product Recommendations

For Entire Net-Work 6 (mainframe), Software AG recommends setting the value of the IDTNAME parameter to "ADABAS5B".

### Example

The following example defines the name ADAOURDB for the second Adabas environment.

```
ADARUN PROG=ADANUC, IDTN=ADAOURDB
```

---

## IDTPSUP : Suppress IDTNAME in Adabas Output Messages

---

This parameter applies to the BS2000 operating system only.

Parameter	Specify . . .	Possible Values	Default
IDTPSUP	whether the IDTNAME should be suppressed in the Adabas nucleus output messages.	YES   NO	NO

This parameter allows you to suppress the IDTNAME in Adabas messages such as:

```
ADAM83 dbid NEW IDT CREATED, NAME=idtname, GROUPS=NO
```

### Example

The following specification produces output messages with X's replacing the IDT name.

```
ADARUN IDTPSUP=YES
```

For example, the ADAM83 message would print as:

```
ADAM83 dbid NEW IDT CREATED, NAME=XXXXXXXX, GROUPS=NO
```

---

## IGNDIB : 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 im-

portant 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:** Software AG recommends 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.

#### 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 : 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 AG 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
```

---

## INTNAS : Time Interval Between SYNS 60 Checkpoints

---

Parameter	Specify . . .	Minimum	Maximum	Default
INTNAS	the time interval between SYNS 60 checkpoints.	1	16777215	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 documentation.

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
```

---

## LBP : Length of Buffer Pool

---

Parameter	Specify . . .	Minimum	Maximum	Default
LBP	the maximum size of the Adabas buffer pool.	80 000	2147483647	350000

The maximum number of bytes to be used for the Adabas buffer pool during a session.

Adabas allocates the number of bytes specified to the Adabas buffer pool if sufficient memory is available. A minimum of 80000 bytes must be available to successfully initiate a session. For performance reasons, however, Software AG recommends that you specify at least 50000 bytes per thread. Ensure that sufficient memory is available to accommodate the LBP size specified. If asynchronous buffer flush management is used (that is, the LFIOP parameter specifies a nonzero value), the LBP value must be greater than that specified by LFIOP.

In environments running in 31-bit addressing mode, the buffer 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.

The buffer efficiency statistic can be used to determine the optimal size of the parameter.



**Note:** When an SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

### Example

The following example allocates a maximum of 800,000 bytes to the Adabas buffer pool for this session.

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

## LCP : Length of Security Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LCP	the maximum size of the Adabas security pool.	2000	16777215	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 50,000 bytes to the Adabas security pool for this session.

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

## LDEUQP : Length of Unique Descriptor Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LDEUQP	the size of the unique descriptor pool.	0	16777215	5000

The unique (UQ) descriptor pool comprises elements that are the length of a unique descriptor value plus 11 bytes. The UQ descriptor pool should be large enough to hold all unique descriptors that are being changed by *all* transactions at any point in time.

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



**Note:** When an SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

### Example

For example, assuming that the average length of unique descriptor values is about 9 bytes, the following Adabas session can process a maximum of about 50 unique descriptors at one time.

```
ADARUN PROG=ADANUC ,LDEUQP=1000
```

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

Parameter	Specify . . .	Minimum	Maximum	Default
LDTP	the number of blocks to allocate to Work part 4 if you have Adabas Transaction Manager Version 7.4 or earlier installed.	0	65535	0
LDTP	the size (in blocks) of the WORK4 index which is used to administer the data on DDWORKR4 if you have Adabas Transaction Manager Version 7.5 or later installed.	8	one eighth of the WORK4 size	8

If you have Adabas Transaction Manager Version 7.4 or earlier installed, this parameter specifies the length of the distributed transaction processing area, Work part 4:

- If DTP=RM and LDTP is either unspecified or specified as `0', the length of Work part 4 is equivalent to that specified for Work part 1 in the ADARUN LP parameter.
- If specified, the ADARUN LDTP must be at least 200 blocks. A safe size of LP/4 is a good value to start with. If you do not specify an LDTP value, it is set to the value of the LP parameter.



- When Work part 4 contains data about incomplete transactions, it may be increased in size but not decreased. Work part 4 must be empty before it can be made smaller.

If you have Adabas Transaction Manager Version 7.5 or later 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 : Asynchronous Buffer Pool Control

Parameter	Specify . . .	Minimum	Maximum	Default
LFIOP	whether to enable asynchronous buffer flush operation and set the I/O pool size.	80000	2147483647	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	The flush pool is to be sized at 25% of the buffer pool (LBP=) size.
n	The flush pool size is determined by the user.

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.

 **Note:** When an SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

**Example**

The following example enables asynchronous buffer flush operation for this session and allocates an I/O pool of 200,000 bytes (25% of the LBP size).

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

## LFP : 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 documentation.

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.

**Example**

The following example allocates 8,000 bytes for the Adabas internal format buffer pool for this session.

---

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

## LI : Length of Table of ISNs (TBI)

---

Parameter	Specify . . .	Minimum	Maximum	Default
LI	the size of the TBI.	2000	16777215	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 30,000 bytes for the table of ISN list IDs for this session.

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

## LNKNAME: Link Globals Table Name

Parameter	Specify . . .	Possible Values	Default
LNKNAME	the module name of the link globals table to be employed by an Adabas 8 batch/TSO link routine. The link globals table is used to obtain default information and for invoking any linked exits.	valid module name for an IBM z/OS, z/VSE or z/VM operating system, or for a BS2000 module	LNKGBLS

The parameter name LNKNAME may be abbreviated to LGN.



### Notes:

1. If you specify the LNKNAME 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 LNKNAME keyword, either the ADALNK or ADALNKR module will be loaded by ADARUN. This is intended to allow the Adabas 7 batch/TSO link routines to be loaded and executed for the job step. Take care to ensure the proper load library concatenation is provided on the job step so the Adabas 7 ADALNK or ADALNKR module will be loaded. This should allow gradual migration of job steps from Adabas 7 to Adabas 8 as required.

### Example:

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

```
ADARUN PROG=USER, LNKNAME=MYGBLS
```

## LOCAL : Local Nucleus or 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 unknown 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 nucleuse or Adabas Review hub to receive calls from other Entire Net-Work nodes.

**Example**

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

```
ADARUN PROG=ADANUC,LOCAL=YES
```

**LOGGING / LOGxxxx : 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 LOGxxxx 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 LOGGING=NO. 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 LOGxxxx parameters. By default, each type of information is *not* logged.

 **Note:** Some parameters are only valid when CLOGLAYOUT=5; other parameters are only valid when CLOGLAYOUT=8.

Parameter	Specify whether to log the ...	Valid for CLOGLAYOUTs...	Possible Values	Default
LOGABDX	Adabas buffer descriptions (ABDs) <sup>1</sup>	8	YES   NO	NO
LOGCB	extended Adabas control block <sup>2</sup>	8 <sup>2</sup>	YES   NO	NO
LOGCLEX	command log extension (CLEX) <sup>3</sup>	5, 8	YES   NO	NO
LOGFB	format buffer	5, 8	YES   NO	NO
LOGIB	ISN buffer	5, 8	YES   NO	NO
LOGIO	I/O activity	5, 8	YES   NO	NO
LOGMB	multifetch buffer <sup>4</sup>	8	YES   NO	NO
LOGRB	record buffer	5, 8	YES   NO	NO
LOGSB	search buffer	5, 8	YES   NO	NO

Parameter	Specify whether to log the ...	Valid for CLOGLAYOUTs...	Possible Values	Default
LOGUX	user exit B data	5, 8	YES   NO	NO
LOGVB	value buffer	5, 8	YES   NO	NO

<sup>1</sup> LOGABDX=YES can only be used if CLOGLAYOUT=8 is used and if LOGUX=YES.

<sup>2</sup> You can specify the LOGCB parameter with CLOGLAYOUT=5, however its value will be ignored because when CLOGLAYOUT=5 is specified, the Adabas control block is always written to the command log file.

<sup>3</sup> LOGCLEX=YES requires REVIEW=NO although the Adabas Review modules must be available. Command log records with CLEX can be viewed using the batch program REVIEWB.

<sup>4</sup> LOGMB=YES can only be used if CLOGLAYOUT=8 is used and if LOGRB=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 : Maximum Command Log Size

Parameter	Specify . . .	Minimum	Maximum	Default
LOGSIZE	the maximum blocksize for the sequential command log DD/LOG.	100	32760	value of QBLKSIZE

This parameter controls the maximum block size for the sequential command log DD/LOG. 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 blocksize of the command log is set as 2,000 bytes.

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

## LP : 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. Details can be found in [Adabas Restart and Recovery](#), elsewhere in this guide.

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.



### Notes:

1. 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.
2. When an SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

### 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 : Length of Sequential Command Table

Parameter	Specify . . .	Minimum	Maximum	Default
LQ	the size of the table of sequential commands.	2000	16777215	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 30,000 bytes for the table of sequential commands for this session.

```
ADARUN PROG=ADANUC ,LQ=30000
```

---

## LRDP : Length of the Redo Pool

---

Parameter	Specify . . .	Minimum	Maximum	Default
LRDP	the size (in bytes) of the redo pool.	0; or 80000	2147483647	LFIOP value

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. Its default is the size of the flush I/O pool (LFIOP parameter) of the cluster nucleus.

- 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.

The nucleus reports on the usage (high watermark) of the redo pool in a shutdown statistic and in the response to the DRES operator command or the corresponding AOS or ADADBS OPERCOM functions.

## LRPL :Adabas or Event Replicator Replication Pool Size

Parameter	Specify . . .	Possible Values	Default
LRPL	The size of the Adabas or Event Replicator replication pools.	minimum 20,000	100,000 (if REPLICATION=YES) or 0 (if REPLICATION=NO)

### Example:

```
ADARUN PROG=ADANUC,REPLICATION=YES,LRPL=50000
```




### Notes:

1. The LRPL parameter should be specified for an Adabas nucleus running with parameter REPLICATION=YES.
2. The LRPL parameter should be specified for a Event Replicator database.
3. The LRPL parameter values in the Adabas nucleus and the Event Replicator Server may be different.
4. Software AG recommends 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.

## LS : 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) minus the minimum LS value (19968) divided by two ( $LS = LWP/2 - 19968$ ). 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):

 **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 less than 65536:

using 1 descriptor LS/12  
 using 2 descriptors LS/16  
 using 3 descriptors LS/20

If the TOPISN in the file is greater than 65536:

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 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 60,160 bytes.

```
ADARUN PROG=ADANUC,LS=60160
```

## LU : 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 (108 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 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.



### 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. On z/OS and VM systems, LU cannot exceed a value greater than that produced by the following calculation:  $(NABvalue \times 4096)$ ; on z/VSE and BS2000 systems, LU cannot exceed a value greater than that produced by the following calculation:  $(NABvalue \times 4096) - 256$ . For more information about the NAB parameter, read [NAB: Number of Attached Buffers](#), elsewhere in this guide.
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 7.4, the LU parameter must be greater than or equal to 164,000. 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.

- 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 setting must be greater than or equal to 164,000.

**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 : 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 Sx 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.

**Exempl:**

For example, for the following Adabas session, the size of the Adabas Work part 2 is 50,000 blocks.

```
ADARUN PROG=ADANUC ,LWKP2=50000
```

## LWP : Length of Adabas Work Pool

Parameter	Specify . . .	Minimum	Maximum	Default
LWP	the size of the Adabas work pool.	80 000 but not less than NT * 25000	2147483647	150 000



**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.

### Example

The following example runs the Adabas nucleus with a work pool of 900,000 bytes.

```
ADARUN PROG=ADANUC ,LWP=900000
```

## MODE : 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 Adabus 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 : 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. Software AG recommends setting MSGBUF to 10 (kilobytes).

Message buffering can be temporarily disabled and subsequently reactivated using the [SMGT operator command](#).

See the Adabas DBA Tasks documentation 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 : Case of Messages Sent to the Console MSGDRUCK : Case of Messages Sent to DD/DRUCK MSGPRINT : Case of Messages Sent to DD/PRINT

Parameter	Specify the case of Adabas . . .	Possible Values	Default
MSGCONSL	messages sent to the console	UPPER   MIXED	MIXED
MSGDRUCK	messages sent to the DD/DRUCK data set		
MSGPRINT	messages sent to the DD/PRINT 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.



**Note:** The first message line in DDPRINT will not be influenced by the value of parameter MSGPRINT.

### Example

The following example will display messages in the DDDRUCK and DDPRINT data sets in all uppercase:

```
ADARUN MSGDRUCK=U,MSGPRINT=U
```

## MXCANCEL: Timeout Threshold for Canceled Peer Nucleus

Parameter	Specify . . .	Minimum	Maximum	Default
<code>MXCANCEL</code>	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 canceled 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 canceled nucleus has terminated within the `MXCANCEL` timeout period, it does one of the following things:

- It returns response 124, 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 canceled, 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, on z/OS systems, consider specifying a `//SVCDUMP DD DUMMY` statement, which usually produces a very quick dump prior to disconnecting. On other systems, and as a backup method on z/OS, specify a regular system dump statement (e.g., `SYSUDUMP` on z/OS) 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 : Timeout Threshold for Canceled 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 canceled 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 canceled 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 canceled 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: 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 : 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, 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 : 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 : 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 documentation.

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 : Maximum Sx Command Execution Time

Parameter	Specify . . .	Minimum	Maximum	Default
MXTSX	the maximum execution time (in seconds) that can be specified for the <b>ADARUN TLSCMD</b> 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 documentation.

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<sub>x</sub> 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 : 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 documentation.

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).

Software AG recommends 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 : 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 : 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.

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.

Users of the Adabas Review hub should read *Storage Requirements in Adabas Review Concepts and Facilities* documentation for more information about the space requirements of the Command Queue for Adabas Review.

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 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
```

### Example

The following example runs the Adabas Review hub nucleus with 50 attached buffers.

```
ADARUN PROG=ADAREV,NAB=50
```

## NC : 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.

Software AG recommends 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 Adabas Review Concepts and Facilities* documentation.

### 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 : 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 3380 device for command logging. The size of each command log data set is 1350 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3380 device has 15 tracks, and each track can contain 9 blocks of command log data. Thus, 10 cylinders = 10 15 9 = 1350 blocks).

```
ADARUN PROG=ADANUC, LOGGING=YES, NCLOG=6, CLOGDEV=3380, CLOGSIZE=1350
```

## NH : 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 (HQE) is required for each record (ISN) placed in hold status. An HQE 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.

Each HQE requires 28 bytes.

In environments running in 31-bit addressing mode, the hold queue 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.



**Note:** When an SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

### Example

The following example runs the Adabas nucleus with a maximum of 2000 elements in the hold queue.

```
ADARUN PROG=ADANUC,NH=2000
```

## NISNHQ : 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	(NH / 4) or 65535, whichever is smaller	(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 most ISNs allowed on hold for a single user is 1/4 of

the hold queue size, but not more than 65,535 ISNs. 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.

If a user attempts to place more records in hold status than permitted, the user receives a nonzero response code even though there may still be space in the hold queue.



**Note:** When an SLOG file is defined for use on the Event Replicator Server, replication data may be written and read from the SLOG file. With this in mind, consider the values set for the following ADARUN performance and tuning parameters: ASYTVS, FMXIO, LBP, LDEUQP, LFIOP, LP, NH, NISNHQ.

### 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 documentation.

### 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 40 (the NISNHQ default of  $(NH / NU) * 2$ , which is greater than the alternative default of 20).

```
ADARUN PROG=ADANUC,NU=50,NH=2000...
```

## NONDES : Nondesoriptor 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, nondesoriptor searches carry significant overhead.

NONDES=NO prevents searches on nondescriptors. An attempt to search on a nondesoriptor field produces response code 68.



**Note:** The behavior of nondesoriptor searches in Adabas databases differs between mainframe and open systems in regards to null suppression in the fields. In open systems, nondesoriptor 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 : 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 3380 device. The size of each log data set is 1200 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3380 device has 15 tracks, and each track contains 8 blocks of protection log data. Thus, 10 cylinders = 10 \* 15 \* 8 = 1200 blocks).

```
ADARUN PROG=ADANUC ,NPLOG=6 ,PLOGDEV=3380 ,PLOGSIZE=1200
```

## NQCID : 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.

### 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 documentation.

### 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 : 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).

Although supported, the BS2000 device type 2007 is not recommended for use with Adabas. Support for the 2007 will be removed in a later Adabas release.

### IBM and Compatible Devices

Device Type	Max. Value	Device Type	Max. Value
3310	1022	8345	5728
3330	1061	8350	2358
3340	877	8380	2267
3350	1155	8381	2867
3359	1155	8385	5865
3370	1278	8390	2674
3375	1022	8391	3418
3379	1022	8392	4611
3380	1371	8393	6995
3389	1371	9332	1278
3390	1429	9335	1278
		9345	2785

### BS2000 Devices

Device Type	Max. Value	Device Type	Max. Value
2000	1022	2010	4096
2001	1022	2200	4096
2002	2046	2201	3072
2003	1534	2202	4096
2004	2558		
2005	2046		
2006	2558		
2007	7678		
2008	8188		
2009	8188		



## 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 documentation.

### 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 : Number of Threads

Parameter	Specify . . .	Minimum	Maximum	Default
NT	the number of user threads to be used during the Adabas session.	3	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.

### Example

The following example runs the Adabas nucleus with six (6) user threads.

```
ADARUN PROG=ADANUC, NT=6
```

## NU : 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.

### Example

The following example runs the Adabas nucleus with a maximum of 500 elements in the user queue.

```
ADARUN PROG=ADANUC,NU=500
```

## NUCID : Adabas Cluster Nucleus ID

---



**Note:** The NUCID parameter replaces the earlier parameters PLXID and SMPID.

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 65,001-65,479). 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.
- Software AG strongly recommends 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, . . .
```

## OPENRQ : 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: 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 on z/OS systems.	YES   NO	NO

This parameter is only valid for z/OS 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: Multiple Protection Log Device Type PLOGSIZE: Multiple Protection Log 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 be 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 3380 device. The size of each log data set is 1200 blocks, based on an allocation of 10 cylinders. (Each cylinder of a 3380 device has 15 tracks, and each track contains 8 blocks of protection log data. Thus, 10 cylinders = 10 \* 15 \* 8 = 1200 blocks).

```
ADARUN PROG=ADANUC,NPLOG=6,PLOGDEV=3380,PLOGSIZE=1200
```

## PLOGSEGSDS: Spanned Record PLOG Control

Parameter	Specify . . .	Possible Values	Default
PLOGSEGSDS	whether all portions of a modified spanned storage record are written to the protection log (PLOG).	YES   NO	NO

When this value is set to NO (the default), only some portions of a modified spanned data record are written to the PLOG; all portions are not written. This optimizes Adabas I/O performance, however it prevents the ADACDC and ADASEL utilities from parsing the records. Set this value to YES if you want the ADACDC or ADASEL utilities to parse modified spanned data records, bearing in mind that I/O performance may be slower. When set to YES, all portions of a modified spanned data record are written to the PLOG.

## PLOGRQ : 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: <ul style="list-style-type: none"> <li>- multiple PLOG data sets are used if a PLOGSIZE parameter is provided;</li> <li>- dual PLOG data sets are used if a DUALPLS parameter is provided;</li> <li>- a SIBA data set is used if JCL for a SIBA data set is provided;</li> <li>- otherwise, no protection log data set is used.</li> </ul>
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 : 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 documentation.

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 : Include Commands in Prefetch/Multifetch Processing PREFIFIL : Include Files in Prefetch/Multifetch 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 20,000 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 3,000 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 : 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.

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 : Prefetch Buffer Length per Call

Parameter	Specify . . .	Minimum	Maximum	Default
PREFSBL	the prefetch buffer length for one prefetch.	1024	65535	3000

The PREFSBL parameter specifies the size of the buffer for a single prefetch, which determines how much record data 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. If a required record is already in a prefetch buffer, the record is taken directly from the buffer. PREFSBL specifies the size of this 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 initial size of PREFSBL:

$$300 + FBL + 2 * (16 + RBL) + 2 * ((16 + RBL) * recs) = PREFSBL$$

In this calculation, *FBL* is the length of the format buffer, *RBL* is the length of the record buffer, and *recs* is the number of records per call.

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 15,000-byte prefetch buffer.

```
ADARUN PROG=USER, PREFETCH=YES, PREFSBL=15000
```

## PREFTB L : Total Size of Prefetch Buffer

Parameter	Specify . . .	Minimum	Maximum	Default
PREFTB L	the size of the total prefetch buffer area.	6144	16777215	30000

PREFTB L 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 PREFTB L value must be equal to or greater than the PREFSBL value. The PREFTB L-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 30,000 bytes for the total prefetch buffer area; this size provides buffer space for up to 10 prefetch sequences.

```
ADARUN PROG=USER, PREFETCH=YES, PREFTB L=30000
```

## PREFXCMD (PREFXFIL) : 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 PREFIFIL/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 20,000 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 3,000 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 : 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 <i>Adabas Session Execution</i> , in <i>Adabas Operations</i> documentation.
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 your Adabas TP monitor installation documentation.
USER	a user program to be run using a non-reentrant Adabas batch/TSO link routine.  For more information, read <i>Linking Applications to Adabas</i> , in <i>Adabas Operations</i> documentation.
<i>utility-name</i>	an Adabas utility  Specify an Adabas utility for <i>utility-name</i> . For more information, refer to <i>Adabas Utilities</i> documentation.

### 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 : 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 Adabas Utilities*.



**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
CKD-Disk	track-size / k, where k = smallest value such that track-size / k <= 32760
FBA-Disk	32760

Under all operating systems except z/VSE, this block size may be provided in the JCL or data set label.

Under BS2000, disk block sizes are always rounded up to the next integer multiple of 2048 bytes.

### Examples

For example, a block size of 65,536 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 32,760 bytes for the Adabas sequential data sets is used by the ADAORD utility:

```
ADARUN PROG=ADAORD,QBLKSIZE=32760
```

## READONLY : 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
```

## REPLICATION Parameter

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.

### Example

The following example activates replication for an Adabas session.

```
ADARUN PROG=ADANUC,REPLICATION=YES
```



## REVFILTER : 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 : 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.
<u>LOCAL</u>	Adabas Review 4.2 and above is started in local mode running as an extension to ADALOG. This is the only active Adabas Review setting available under z/VM.
dbid	Adabas Review is started in hub mode. The physical database ID that you specify for the hub identifies <ul style="list-style-type: none"> <li>■ the hub (server) itself (with PROGRAM=ADAREV) that is being started; or</li> <li>■ from an Adabas nucleus (client), the hub that is the target for Adabas Review processing for that nucleus (with PROGRAM=ADANUC).</li> </ul>

### 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.

### 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: 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: 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 32768 (32K).	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**

The following example runs the Adabas nucleus using a command logged buffer size limit of 32000 bytes. The sum of all logged buffers for an Adabas Review command to this nucleus cannot exceed 10000 bytes.

```
ADARUN PROG=ADAREV, REVLOGMAX=10000
```

**RPLCONNECTCOUNT Parameter**

Parameter	Specify . . .	Possible Values	Default
RPLCONNECTCOUNT	the number of connection attempts made for the Adabas or Event Replicator Server nucleus after an attempt fails (response 148 is issued).	0 - 2,147,483,647	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

---

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 (response 148 is issued).	0 - 2,147,483,647	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

---

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

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 Event Replicator for Adabas Concepts*. 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.

### 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

---

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 <a href="#">RPWARNPERCENT parameter</a> .	1-99	10

This ADARUN parameter applies to both the Adabas and Event Replicator Server replication pools.

## RPWARNINTERVAL Parameter

---

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

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

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).

## SMGT : 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



**Note:** This parameter is not available on BS2000 systems.

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 documentation.

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 : Sort Area Controller Caching

This parameter is valid for the operating environments z/OS and z/VM only.



Parameter	Specify . . .	Possible Values	Default
<code>SortCache</code>	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 pre-read and maintained as long as possible in the caching facilities of the disk controller.

**NO** Sort blocks are not pre-read 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 (`AssocCache`), 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 : Activate Adabas Triggers and Stored Procedures

Parameter	Specify . . .	Possible Values	Default
<code>SPT</code>	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 documentation.

#### Example

The following example sets Adabas to use the triggers and stored procedures functions.

```
ADARUN PROGRAM=ADANUC ,MODE=MULTI ,SPT=YES
```

## SUBMPSZ : GETMAIN Memory Pool for Subtasks

Parameter	Specify . . .	Possible Values	Default
SUBMPSZ	the common memory pool size, in bytes, for subtask communication in products such as Adabas Review, Adabas Parallel Services, and Event Replicator for Adabas.	100000 - address-limit	1,024,000

For Adabas Review on BS2000 systems, the recommended value is 140,000,000 bytes.



### Notes:

1. Setting this parameter for Adabas Review replaces an optional zap for increasing the subtask common memory.
2. This parameter must be set to the recommended value for the Adabas Review hub and the Adabas Review nuclei, irrespective of the value of the REVIEW parameter.
3. For Adabas triggers and stored procedures in BS2000 environments, make sure that the nucleus is started with the ADARUN parameter SUBMPSZ set to "2,000,000" (or larger). Otherwise, the Natural subtask will deliver a "ADAI2S - 04000004 no mother task common memory" error.

### Example

The following example allows for four (4) megabytes of common memory pool storage for use in the communication between the Adabas nucleus and the subtasks.

```
ADARUN PROG=ADANUC ,SUBMPSZ=4096000
```

## SVC : SVC Number

This parameter applies to the operating environments z/OS and z/VSE only.

Parameter	Specify . . .	Possible Values	Default
SVC	the Adabas SVC number or Adabas Review hub SVC number to be used for the session.	see text	45 (z/VSE) 249 (z/OS)

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 and z/VSE.

Valid SVC values are as follows:

z/OS 200-255

z/VSE 45 is recommended; any free SVC value can be used. See the Adabas Installation documentation for information about finding free values for z/VSE.

### Example

The following example runs an Adabas session under z/OS using SVC 202 for the Adabas SVC.

```
ADARUN PROG=ADANUC ,SVC=202
```

The following example runs an Adabas Review hub session under z/OS using SVC 202 for the Adabas Review hub SVC.

```
ADARUN PROG=ADAREV ,SVC=202
```

## TAPEREL : Tape Handling Control (BS2000 Only)

Parameter	Specify . . .	Possible Values	Default
TAPEREL	end-of-file (EOF) handling for tapes and cartridges.	KEEPUNL   KEEP   NO   RELEASE   UNLOAD	KEEPUNL

Value	Meaning
-------	---------

KEEPUNL (default) Retain the I/O unit assignment but unload the tape or cartridge.

KEEP Retain the I/O unit assignment without unloading.

NO Do not perform the RELEASE macro function.

RELEASE Do not keep the I/O unit assignment; do not unload the tape or cartridge.

UNLOAD Do not keep the I/O assignment; unload the tape/cartridge.

### Example

In the following example, the I/O assignment remains and the tape/cartridge remains loaded after the ADAULD utility is executed.

```
ADARUN PROG=ADAULD ,TAP=KEEP
```

## TARGETID : 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), Adabas TPF, Natural global bufferpools, 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
```

## TASKCTGY : Adabas Batch/TP Task Category Control

This parameter applies in BS2000 operating environments only.

Parameter	Specify . . .	Possible Values	Default
TASKCTGY	the Adabas task category.	BATCH   TP	BATCH

BS2000 TP (interactive processing) environments favor TP-processing tasks over batch tasks. By default, Adabas has batch status. You can use the TASKCTGY parameter to assign TP-processing priority to the Adabas nucleus.

### Value Meaning

BATCH (default) The nucleus retains batch status.

TP Issues a TINF macro to force the Adabas nucleus to TP status.

### Example

The following example forces the nucleus to TP-processing status.

```
ADARUN PROG=ADANUC , TASKCTGY=TP
```

## TCPIP : TCP/IP Access Control (z/OS Only)

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 version 7 supports a direct TCP/IP link to the z/OS 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://STACKNAME:12345
```

## TCPURL : 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
```

where

*api-name* is a 1-3 character value identifying the application programming interface (API) to use. Both APIs for the IBM TCP/IP stack (HPS, OES) and the API for the Interlink stack (ILK) are currently supported.

*stackid* is a 1-8 character value identifying the stack to use. For the HPS API, this is the name of the TCP/IP started task. For the OES API, no value is needed. For the ILK API, this is the subsystem identifier.

*port-number* is a 1-5 character number in decimal notation.

Optionally, you can specify this and additional URLs using the operator command [TCPIP](#).

### Example

The following example runs Adabas with a direct TCP/IP link to the nucleus.

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=HPS://STACKNAME:12213
```

## TEMPCACHE : Temp Area Controller Caching

---

This parameter is valid for the operating environments z/OS and z/VM only.

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 : Synchronous Buffer Flush Duration

Parameter	Specify . . .	Minimum	Maximum	Default
TFLUSH	the time allowed for a synchronous buffer flush.	1	16777215	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 : 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 <a href="#">ADARUN MSTSX</a> 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 MSTSX](#) parameter, the maximum MSTSX 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 documentation.

### 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 : 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 are likely to occur.

### Example

ATM will allow 250 entries in its internal request queue.

```
ADARUN PROG=ADANUC, DTP=TM, TMDRQ=250
```



## TMETDATA : 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	<p>ET data is always stored in and read from the ATM transaction manager's database, without regard to the database ID indicated by the Adabas command.</p> <ul style="list-style-type: none"> <li>■ treats ET data as belonging to a global transaction rather than to a database;</li> <li>■ there is just one copy of a client's ET data in a system;</li> <li>■ eliminates confusion resulting from the existence of different ET data in different databases under the same ETID;</li> <li>■ overrides the Natural <code>ETDB</code> parameter;</li> <li>■ should not be used if the <code>EmergencySerialETs</code> runtime control will be set to <code>YES</code> or <code>FORCE</code> for any client environment in which ET data is stored or read.</li> </ul>
TARGETS	<p>ET data is stored during the commit process in all changed databases that are running with <code>DTP=RM</code>.</p> <p>ET data that is stored by an <code>ET</code> or <code>CL</code> command is always stored in the database that is the target of the command, whether or not that database runs with <code>DTP=RM</code>. If the database runs with <code>DTP=NO</code>, the ET data is written to the database only after any open global transaction has been committed.</p> <p>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.</p> <ul style="list-style-type: none"> <li>■ treats ET data as belonging to a database or a local database transaction rather than to a global transaction;</li> <li>■ a client's ET data might exist in several databases in a system;</li> <li>■ a client can have several different ET data values in different databases at the same time;</li> <li>■ means that 3GL application programs using ET data must have knowledge of the database that holds ET data for a given transaction;</li> <li>■ honors the Natural <code>ETDB</code> parameter.</li> </ul>

**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.

**Example**

ET data is to be stored in the transaction manager's database only.

```
ADARUN PROG=ADANUC ,DTP=TM ,TMETDATA=ATM
```

**TMGTT : 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 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 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.

**Example**

ATM will back out global transactions that are not prepared within about 600 seconds (10 minutes) of the first transactional command.

ADARUN PROG=ADANUC ,DTP=TM ,TMGTT=600

## TMLOG : 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 datasets.

Possible values:

Value	Description
YES	The ATM manager writes every record to an ATMLOG dataset. 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 datasets can be deactivated at any time using the operator command <code>TM NOLOG</code> .
NO	The ATM manager will not write log records to the ATMLOG datasets. Logging to the ATMLOG datasets can be activated at any time using the operator command <code>TM LOG</code> . 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.

On z/OS and VSE systems, if the ATM manager writes log records to the ATMLOG (TMLOG=YES or `TM LOG` issued) and the dataset currently in use becomes full, the manager closes that dataset and switches to the other dataset. Automatic log switching currently does not take place on BS2000 systems. A system message might appear in the console log, indicating that the first log dataset has become full. You may also switch between the two ATMLOG datasets by using the operator command `TM FE0FLOG`. Whenever the ATMLOG datasets are switched, a message is displayed on the console indicating the dataset that is currently in use.

### Example

The ATM manager is to maintain its internal log buffer areas and write all log records to the ATMLOG datasets.

ADARUN PROG=ADANUC ,DTP=TM ,TMLOG=YES

## TMMMSGSEV : 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 and to the DDPRINT dataset has a severity level. Message severity levels are explained in the Adabas Transaction Manager *Messages and Codes*.

Using the TMMMSGSEV 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.

All messages created by the ATM manager are written to DDPRINT regardless of the TMMMSGSEV specification.

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.

### Example

The ATM manager should not send message of severity less than 4 to the console.

```
ADARUN PROG=ADANUC ,DTP=TM ,TMMMSGSEV=4
```

## TMRESTART : 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.
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.

### Example

After recording them in the STJ, the ATM manager is to discard details of any global transactions that it owns, and that remain incomplete at the end of its restart processing.

```
ADARUN PROG=ADANUC ,DTP=TM ,TMRESTART=FORCE
```

## TMSYNCMGR : 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.

### Example

ATM registers with RRMS so that it can take part in transactions involving other RRMS-enabled resource managers.

```
ADARUN  PROG=ADANUC ,DTP=TM ,TMSYNCMGR=RRMS
```

## TMTCIDPREF : 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 ATM database ID.

*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.

### Example

ATM will assign dynamic TCIDs beginning with the string "DY".

```
ADARUN  PROG=ADANUC ,DTP=TM ,TMTCIDPREF=DY
```

## TNAA : Nonactivity Time Limit for Access-Only Users

Parameter	Specify . . .	Minimum	Maximum	Default
TNAA	the nonactivity time limit for access-only users.	1	16777215	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 SMPID 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 Referencedocumentation.
- 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 documentation).

### 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 : Nonactivity Time Limit for ET Logic Users

Parameter	Specify . . .	Minimum	Maximum	Default
TNAE	the nonactivity time limit for ET logic users.	1	16777215	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 SMPID 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 documentation.

- 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 documentation).

### 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 : Nonactivity Time Limit for Exclusive Control Users

Parameter	Specify . . .	Minimum	Maximum	Default
TNAX	the nonactivity time limit for exclusive control users.	1	16777215	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 SMPID 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 documentation.
- 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 documentation).

### 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 : Transaction Time Limit

Parameter	Specify . . .	Minimum	Maximum	Default
TT	the transaction time limit for ET logic users.	1	16777215	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).

Software AG recommends 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 documentation.
- 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 documentation).

### 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 : 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.

The specified user exit routine(s) must be loadable at execution time.

The Adabas user exits documentation in *Adabas User Exits* 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
```

## UTIONLY : 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
```

## VISTA : 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 *Adabas DBA Tasks*.

### Example

The following example runs Adabas with Adabas Vista.

```
ADARUN PROG=ADANUC,VISTA=YES
```

## WORKCACHE : Work Area Controller Caching

This parameter is valid for the operating environments z/OS and z/VM only.

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 pre-read and maintained as long as possible in the caching facilities of the disk controller.

NO Work blocks are not pre-read 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
```

# 6 Setting Adabas Session Parameters

---

- General Guidelines ..... 196
- Using Session Statistics to Determine Parameter Settings ..... 200
- Parameters for Operating Systems ..... 200

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 Version 7.4 or earlier installed, this parameter defines the length of the distributed transaction processing area (Work part 4) for use with DTP=RM.  If you have Adabas Transaction Manager Version 7.5 or later 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

Parameter	Description
NC	Number of elements in command queue
NH	Number of elements in hold queue
NISNHQ	Maximum number of records in hold queue per user
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 Sx command for individual user
MXTT	Maximum transaction time limit using OP command for individual user
TLSCMD	Sx 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 (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.

### BS2000

Parameter	Description
CMADDR	Start address for common memory pool
CMDQMODE	Command queue memory pool location
CMFIX	Fix location for common memory pool
CMLADDR	Below 16MB; see CMADDR
CMLFIX	Below 16MB; see CMFIX
CMLSCOPE	Below 16MB; see CMSCOPE
CMLSIZE	Below 16MB; see CMSIZE
CMSCOPE	Access to common memory pool
CMSIZE	Use and size of common memory pool
GROUPS	Resource naming restrictions
IDTNAME	ID table name
TAPEREL	Tape handling
TASKCTGY	Adabas task category control

### z/OS and VSE or Compatible Systems

Parameter	Description
SVC	z/OS or VSE SVC assignment for Adabas

**z/OS and z/VM Systems Only**

Parameter	Description
XXXXCACHE	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 .....	204
▪ Adabas Fastpath ADARUN Parameters .....	205
▪ Adabas Cluster Services ADARUN Parameters .....	205
▪ Adabas Delta Save ADARUN Parameters .....	206
▪ Adabas Parallel Services ADARUN Parameters .....	206
▪ Adabas Review ADARUN Parameters .....	207
▪ Adabas Transaction Manager ADARUN Parameters .....	207
▪ Adabas Vista ADARUN Parameters .....	208
▪ Event Replicator for Adabas ADARUN Parameters .....	208

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
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
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
<code>CLOGMRG</code>	Merge cluster command logs automatically
<code>CLUCACHENAME</code>	Adabas cluster cache structure/area name
<code>CLUGROUPNAME</code>	Adabas cluster group name
<code>CLULOCKNAME</code>	Adabas cluster lock structure/area name
<code>CLUSTER</code>	Adabas cluster nucleus session control
<code>DIRRATIO</code>	Together with <code>ELEMENTRATIO</code> defines the ratio of directory entries to data elements in Adabas cluster environments
<code>ELEMENTRATIO</code>	Together with <code>DIRRATIO</code> defines the ratio of directory entries to data elements in Adabas cluster environments
<code>LRDP</code>	Length of the cluster redo pool
<code>MXCANCEL</code>	Time limit for a canceled peer nucleus to terminate
<code>MXCANCELWARN</code>	Cluster nucleus timeout threshold for canceled peer nucleus warning
<code>MXMSG</code>	Time limit for an inter-nucleus command to be processed
<code>MXMSGWARN</code>	Cluster nucleus timeout threshold for internucleus command
<code>MXSTATUS</code>	Member-level XCF status monitoring heartbeat
<code>MXWTOR</code>	Cluster nucleus self-termination operator query interval
<code>NUCID</code>	Adabas cluster nucleus ID
<code>PROGRAM</code>	Program to run

## Adabas Delta Save ADARUN Parameters

---

The following table lists the ADARUN parameters pertinent to Adabas Delta Save:

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
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
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
LRDP	Length of the cluster redo pool
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



## Adabas Review ADARUN Parameters

The following table lists the ADARUN parameters pertinent to Adabas Review:

Parameter	Usage
CMDQMODE	Command queue memory pool location (BS2000 only)
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
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
SUBMPSZ	GETMAIN memory pool for subtasks (BS2000 only)
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 Version 7.4 or earlier installed, this parameter defines the length of the distributed transaction processing area (Work part 4) for use with DTP=RM.  If you have Adabas Transaction Manager Version 7.5 or later 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

Parameter	Usage
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
LP	Length of data protection area (Work part 1)
LRPL	Adabas or Event Replicator replication pool size
LU	Length of intermediate user buffer
NAB	Number of attached buffers
NH	Number of hold queue elements
NISNHQ	Number of ISNs in hold queue for user
NT	Number of threads
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
RPWARNINCREMENT	Replication pool usage threshold problem warning message interval
RPWARNINTERVAL	Replication pool usage warning messages suppression interval

Parameter	Usage
RPWARNMESSAGELIMIT	Replication pool usage warning message count before message suppression begins
RPWARNPERCENT	Replication pool usage threshold



# III

---

■ 8 Operator Commands .....	213
■ 9 Adabas Session Execution .....	251
■ 10 Adabas Utility Execution .....	267
■ 11 Linking Applications to Adabas .....	275
■ 12 Adabas Restart and Recovery .....	303
■ 13 Local Time Zone Change Support .....	315
■ 14 Adabas Timeout Feature .....	319



# 8 Operator Commands

---

- Entering Operator Commands ..... 214
- Operator Commands ..... 216
- SMGT Display Output Samples ..... 248

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.

## Entering Operator Commands

---

This section describes how to enter operator commands in different environments. It is divided into the following topics:

- [Entering Commands on BS2000 Systems](#)
- [Entering Commands on z/OS Systems](#)
- [Entering Commands on z/VM Systems](#)
- [Entering Commands on VSE Systems](#)

### Entering Commands on BS2000 Systems


In BS2000 environments, enter each command at the operator console by addressing the Adabas nucleus with its task sequence number (TSN) in the following form:

```
/INTR TSN, command
```

For testing purposes, the nucleus may be run as a dialogue process. The nucleus may be interrupted by pressing the K2 key, after which the prompt "/" appears. Now an operator command can be sent to the nucleus in the following form:



```
/INTR command
```

 **Note:** In the dialogue mode, the nucleus stops as long as the INTR message is not sent back. The resume statement /RESUME causes the nucleus to resume where it was interrupted when no operator command is issued.

Operator commands are processed by a STXIT routine.

### Entering 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
```

where *jobname* is the name specified by the EXEC job control statement (usually ADARUN).

### Entering Commands on z/VM Systems

In z/VM environments, use the CP SEND command to enter operator commands from either the console of the virtual machine or from a secondary console of a disconnected virtual machine.

#### Example:

```
CP SEND USER04 ADAEND
```

- issued at the DBA virtual machine console sends the ADAEND command to the USER04 virtual machine, providing that the CP directory for the USER04 virtual machine contains a CONSOLE statement enabling the DBA console as a secondary console.

### Entering Commands on VSE Systems

To enter operator commands in VSE environments, use the following steps:

1. Enter an MSG command for the VSE partition in which Adabas is executing.

#### Example:

```
MSG Fn
```

When ready for communication, Adabas will respond with Adabas message ADAI29. In cases where the commands ADAEND, CANCEL, and HALT cause Adabas to end the nucleus session, no outstanding reply is presented so that an orderly shutdown can occur without the need for operator intervention.

2. Enter the desired Adabas operator command or commands.

More than one command can be entered in a session. You can also enter a command followed directly by a slash (/) to end any further operator communications until the next MSG command by VSE.

3. When all desired commands have been entered, close the operator communication session by entering a null command (EOB).
4. To enter any more commands after the session is closed, repeat this entire procedure.

## Operator Commands

---

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
- ASYTVS Command
- CANCEL Command
- CLOGMRG Command
- CLUFREEUSER Command
- CT Command
- DAUQ Command
- DCQ Command
- DDIB Command
- DDSF Command
- DELUF Command
- DELUI Command
- DFILES Command
- DFILUSE Command

- DHQ Command
- DHQA Command
- DLOCKF Command
- DNC Command
- DNFV Command
- DNH Command
- DNU Command
- DONLSTAT Command
- DPARM Command
- DPPT Command
- DRES Command
- DSTAT Command
- DTH Command
- DUQ Command
- DUQA Command
- DUQE Command
- DUUQE Command
- FEOFCL Command
- FEOFPL Command
- FMXIO Command
- HALT 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
- NOLOGGING Command
- NOLOGCB Command
- NOLOGFB Command
- NOLOGIB Command
- NOLOGIO Command
- NOLOGRB Command
- NOLOGSB Command
- NOLOGUX Command
- NOLOGVB Command
- ONLRESUME Command
- ONLSTOP Command
- ONLSUSPEND Command>

- RALOCKF Command
- RALOCKFA Command
- RDUMPST Command
- READONLY Command
- REVIEW Command
- STOPF Command
- STOPI Command
- STOPU Command
- SYNCC Command
- TCPIP Command
- TNAA Command
- TNAE Command
- TNAX Command
- TT Command
- UNLOCKF Command
- UNLOCKU Command
- UNLOCKX Command
- UTIONLY 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 have 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 has the lock. Until then, the exclusive-control user must wait.

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.

### **AOSLOG Command**

```
AOSLOG={YES | NO}
```

Use the `AOSLOG` command to activate and deactivate logging of certain Adabas calls that modify the nucleus to DD/PRINT. These calls are issued either by ADADBS OPERCOM or Adabas Online System. Read and display calls are not logged.

### **ASYTVS Command**

```
ASYTVS={YES | NO}
```

Use the `ASYTVS` command to activate or deactivate asynchronous flushing of buffers based on volume serial number.

**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 | UIDPREFIX} = 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:

<code>cccccccc</code>	where the argument is 1-8 letters, digits, or embedded '-' signs without surrounding apostrophes.
<code>'cccccccc'</code>	where the argument is 1-8 characters with surrounding apostrophes.
<code>X'xxxxxxxxxxxxxxxxxx'</code>	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, 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.

### DAUQ Command



**DAUQ**

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.

### 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`.

## 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.

### 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.

**DHQA Command**A rectangular button with a thin blue border containing the text "DHQA" in a bold, blue, sans-serif font.

Use the `DHQA` command to display up to 1000 hold queue elements.

**DLOCKF Command**A rectangular button with a thin blue border containing the text "DLOCKF" in a bold, blue, sans-serif font.

Use the `DLOCKF` command to display the locked files.

**DNC Command**A rectangular button with a thin blue border containing the text "DNC" in a bold, blue, sans-serif font.

Use the `DNC` command to display the number of posted command queue elements waiting to be selected.

**DNFV Command**A rectangular button with a thin blue border containing the text "DNFV" in a bold, blue, sans-serif font.

Use the `DNFV` command to display nucleus file variables; that is, 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, subcode 15.

### Sample Output

```
ADAI29 OPER CMD: DNFV
FNR=00008 A=Y U=Y ID= CA=00000 CU=00001
```

where;

FNR= <i>nnnnn</i>	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= <i>nucid</i>	is the ID of the nucleus that owns the file lock, if the file is locked.
CA= <i>nnnnn</i>	is the number of users on this nucleus who are currently accessing this file.
CU= <i>nnnnn</i>	is the number of users on this nucleus who are currently updating this file.

### DNH Command



Use the `DNH` command to display the number of ISNs currently in the hold queue.

### DNU Command



Use the `DNU` command to display the number of current users.

### DONLSTAT Command



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**

 A rectangular box with a thin blue border containing the word "DPARM" in a bold, blue, sans-serif font.

Use the **DPARM** command to display the Adabas session parameters currently in effect.

**Sample Output**

```
ADAI29 Oper cmd: DPARM
ADAN16 00199 2009-01-27 22:48:43 READONLY=NO,UTIONLY=NO
ADAN16 00199 2009-01-27 22:48:43 ASYTVS=YES,AOSLOG=NO
ADAN16 00199 2009-01-27 22:48:43 NC=100,NH=500,NT=8,NU=100
ADAN16 00199 2009-01-27 22:48:43 LBP=5026304,LFP=20000,LWP=320000
ADAN16 00199 2009-01-27 22:48:43 LI=10000,LP=500,LQ=10000,LS=79872
ADAN16 00199 2009-01-27 22:48:43 LFIOP=80000,FMXIO=1,LU=65535
ADAN16 00199 2009-01-27 22:48:43 TT=6000,TNAA=6000,TNAE=6000,CT=60
ADAN16 00199 2009-01-27 22:48:43 TNAX=6000,MXTT=3600,MXTNA=3600
ADAN16 00199 2009-01-27 22:48:43 TLSCMD=300,MXTSX=3600
ADAN16 00199 2009-01-27 22:48:43 NOLOG
ADAN41 00199 2009-01-27 22:48:43 Function completed
```

**DPPT Command**

 A rectangular box with a thin blue border containing the word "DPPT" in a bold, blue, sans-serif font.

Use the **DPPT** command to display the parallel participant table (PPT); that is, to display the nucleus's own PPT block.

This command produces internal information for use by Software AG 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



Use the `DRES` command to display the allocated pool space and the highest use level (high water mark) reached so far during the current session by record count and by percent for the following resources:

- Attached buffers (AB) - current allocation not supported
- 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
- Replication pool
- User queue file list pool
- Work pool (WP)
- Pool for global transaction IDs (XIDs; nonzero only with Adabas Transaction Manager)
- Redo pool (nonzero only with Adabas Cluster Services)

The actual values are displayed in nucleus message ADAN28 described in the *Adabas Messages and Codes* documentation.

#### **DSTAT Command**

A rectangular button with a thin blue border containing the text "DSTAT" in a bold, blue, sans-serif font.

Use the DSTAT command to display the current Adabas nucleus status.

#### **DTH Command**

A rectangular button with a thin blue border containing the text "DTH" in a bold, blue, sans-serif font.

Use the DTH command to display thread status.

#### **DUQ Command**

A rectangular button with a thin blue border containing the text "DUQ" in a bold, blue, sans-serif font.

Use the DUQ command to display up to five active and inactive user queue elements.

#### **DUQA Command**

A rectangular button with a thin blue border containing the text "DUQA" in a bold, blue, sans-serif font.

Use the DUQA command to display up to 100 user queue elements.

### 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.

### DUUQE Command

**DUUQE**

Use the `DUUQE` command to display utility user queue elements.

### FEOFCL Command

**FEOFCL [,GLOBAL]**

Use the `FEOFCL` command to close the current dual or multiple command log and switch to the 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.

#### 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. This command is only valid when CLOGLAYOUT=5.

### **LOGVB Command**

**LOGVB**

Use the LOGVB command to start logging of the Adabas value buffer for each command logged.

### **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**A rectangular button with a thin blue border containing the text **NOLOGFB** in blue, bold, uppercase letters.

Use the `NOLOGFB` command to stop or prevent logging of the Adabas format buffer.

**NOLOGIB Command**A rectangular button with a thin blue border containing the text **NOLOGIB** in blue, bold, uppercase letters.

Use the `NOLOGIB` command to stop or prevent logging of the Adabas ISN buffer.

**NOLOGIO Command**A rectangular button with a thin blue border containing the text **NOLOGIO** in blue, bold, uppercase letters.

Use the `NOLOGIO` command to stop or prevent logging of Adabas I/O activity.

**NOLOGRB Command**A rectangular button with a thin blue border containing the text **NOLOGRB** in blue, bold, uppercase letters.

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. This command is only valid when `CLOGLAYOUT=5`.

### **NOLOGVB Command**

**NOLOGVB**

Use the `NOLOGVB` command to stop or prevent logging of the Adabas value buffer.

### **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 remove the advance lock on the specified file (see `ALOCKF` command) without running the utility.

**RALOCKFA Command**

```
RALOCKFA
```

Use the `RALOCKFA` command to remove the advance lock on all files for which it has been set (see `ALOCKF` command) without running the utility.

### 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.

### 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.

### 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.

This command does not stop EXF or UTI users.



## STOPI Command

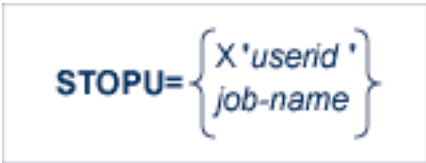


**STOPI=** *time*

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.



This command does not stop EXF or UTI users.

## STOPU Command



**STOPU=**  $\left\{ \begin{array}{l} X'userid' \\ job-name \end{array} \right\}$

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.

- 
**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.
- 
**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'
```

## SYNCC Command

A rectangular box with a thin border containing the text "SYNCC" in a bold, blue, sans-serif font.

Use the `SYNCC` command to force synchronization for all ET users. The nucleus waits for all ET users to reach ET status.

## TCPIP Command

A rectangular box with a thin border containing the text "TCPIP={ OPEN=url | CLOSE=url | CLOSE }" in a bold, blue, sans-serif font.

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.

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` `TCPUURL` 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:

*api-name://stackid:port-number*

where:

*api-name* is a 1-3 character value identifying the application programming interface (API) to use. Both APIs for the IBM TCP/IP stack (HPS, OES) and the API for the Interlink stack (ILK) are currently supported.

*stackid* is a 1-8 character value identifying the stack to use:

- for the HPS API, this is the name of the TCP/IP started task.
- for the OES API, no value is needed.
- for the ILK API, this is the subsystem identifier.

*port-number* is a 1-5 character number in decimal notation.

## Examples

```
TCPIP=OPEN=ILK://ILZ5:1234
```

```
TCPIP=CLOSE=ILK://ILZ5:1234
```

To close all open URLs:

```
TCPIP=CLOSE
```

### TNAA Command

A rectangular box with a thin border containing the text **TNAA= n** in blue font.

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

A rectangular box with a thin border containing the text **TNAE= n** in blue font.

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

A rectangular box with a thin border containing the text **TNAX= n** in blue font.

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

A rectangular box with a thin blue border containing the text `TT=n` in a bold, blue, sans-serif font.

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

A rectangular box with a thin blue border containing the text `UNLOCKF=n` in a bold, blue, sans-serif font.

Use the `UNLOCKF` command to unlock the specified file. File usage is restored to its pre-locked status.

### UNLOCKU Command

A rectangular box with a thin blue border containing the text `UNLOCKU=n` in a bold, blue, sans-serif font.

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

A rectangular box with a thin blue border containing the text `UNLOCKX=n` in a bold, blue, sans-serif font.

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`.

**Adabas Utility Operator Command DSTAT**

```
DSTAT
```

Use the `DSTAT` command to display the current `ADALOD`, `ADAORD`, `ADARES`, `ADASAV`, `ADAULD`, or `ADAVAL` operating status. The following are examples of the output that results when `DSTAT` is entered during specific Adabas utility operations:



**Note:** `ADAORD`, `ADARES`, `ADAULD`, and `ADAVAL` do not accept operator commands if they run in `MODE=SINGLE`.

**Example 1: 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 2: 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 3: 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 4: 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 5: 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, ACTIPIN, 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
<i>exit-code</i>	The code that identifies an exit routine to an SMGT operator command is one of the following: <ul style="list-style-type: none"> <li>■ UEX<math>n</math>, where <math>n</math> is a single-digit user exit number</li> <li>■ UX<math>nn</math>, where <math>nn</math> is a double-digit user exit number</li> <li>■ HX<math>nn</math>, where <math>nn</math> is a hyperexit number (single-digit numbers are preceded by a zero; e.g., HX02)</li> <li>■ CX00 (the Adabas Caching Facility user exit)</li> <li>■ SX00 (the PIN routine user exit)</li> </ul>
<i>start, end</i>	The range of addresses for the SNAP operand where <i>start</i> is the hexadecimal address where the SNAP dump begins and <i>end</i> is the hexadecimal address where it ends.
<i>module-name</i>	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.

Variable Type	Description
<i>pin-number</i>	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, hyperexit, 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`.

## 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`:

<code>ALL</code>	(the default) displays all reports
<code>EXITS</code>	displays current user exit, hyperexit, and other exit status
<code>MSGBUF</code>	displays message buffering status
<code>PINS</code>	displays PIN routine status
<code>SUMMARY</code>	displays only the summary information from <code>ALL</code>
<code>LAST</code>	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
ADAN02 00127 SMGT COMMAND PROCESSED
```

## DISPLAY=MSGBUF

```
ADANA1 00127 SMGT DISPLAY ACTIVE
ADANA2 00127 SMGT ACTIVE
ADANAE 00127 MESSAGE BUFFERING IS ACTIVE
ADAN02 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
ADAN02 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
ADAN02 00127 SMGT COMMAND PROCESSED
```

**DISPLAY=LAST**

No error:

```
ADANA1 00127 SMGT DISPLAY ACTIVE
ADANA2 00127 SMGT ACTIVE
ADANA5 00127 NO ERROR CONDITIONS HANDLED
ADAN02 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*
ADAN02 00127 SMGT COMMAND PROCESSED
```

# 9 Adabas Session Execution

---

▪ Program Synchronization Using Operating System Services .....	252
▪ BS2000 Session .....	253
▪ z/OS Session .....	255
▪ z/VM Session .....	259
▪ VSE Session .....	260
▪ Notes .....	262
▪ Single-User Mode .....	264
▪ User Profile .....	265

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.

## Program Synchronization Using Operating System Services

---

Adabas version 7 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.

The operating system services used are as follows:


Operating System	Service
z/OS	systems-wide ENQ/DEQ macros (SCOPE=SYSTEMS) with major name (QNAME) 'ADABAS'
BS2000	system-wide ENQAR/DEQAR macros (SCOPE=GLOBAL)
VSE	system-wide LOCK/UNLOCK macros
z/VM	system-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.

## BS2000 Session

The following table contains all data sets which are used when executing an Adabas session under BS2000.

Data Set	Link Name	Logical Unit	Storage Medium	Additional Information
ADARUN parameters	DDCARD	SYSDTA	disk	<a href="#">note 1</a>
ADARUN / Adabas messages	DDPRINT	SYSOUT	disk	<a href="#">note 2</a>
Associator	DDASSOR $n$ or DDASSO $nn$		disk	<a href="#">note 3</a>
Data Storage	DDDATAR $n$ or DDDATA $nn$		disk	<a href="#">note 3</a>
Work	DDWORKR1 DDWORKR4		disk	<a href="#">note 4</a>
Recovery Aid log	DDRLOGR1		disk	<a href="#">note 5</a>
Protection log multiple log 1 multiple log 2	DDSIBA DDPLOGR1 DDPLOGR2		tape disk disk disk	<a href="#">note 6</a> <a href="#">note 7</a> <a href="#">note 7</a>
Command log multiple log 1 multiple log 2	DDLOG DDCLOGR1 DDCLOGR2		tape disk disk disk	<a href="#">note 8</a> <a href="#">note 9</a> <a href="#">note 9</a>
ECS encoding objects	DDECSOJ		tape/disk	<a href="#">note 10</a>

### Example of Adabas Session Job Control (BS2000)

This job includes multiple protection logging, multiple command logging, Recovery Aid logging, and universal encoding support (UES):

In SDF Format:

```
/.ADANUC LOGON
/MODIFY-TEST-OPTIONS DUMP=YES
/REMARK *
/REMARK * ADABAS NUCLEUS
/REMARK *
/ASS-SYSLST L.NUC
/ASS-SYSDTA *SYSCMD
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK DDASSOR1,ADAYyyyy.ASSO,SHARE-UPD=YES
/SET-FILE-LINK DDDATAR1,ADAYyyyy.DATA,SHARE-UPD=YES
```

```
/SET-FILE-LINK DDWORKR1,ADAyyyyy.WORK
/SET-FILE-LINK DDPLOGR1,ADAyyyyy.PLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDPLOGR2,ADAyyyyy.PLOGR2,SHARE-UPD=YES
/SET-FILE-LINK DDCLOGR1,ADAyyyyy.CLOGR1,SHARE-UPD=YES
/SET-FILE-LINK DDCLOGR2,ADAyyyyy.CLOGR2,SHARE-UPD=YES
/SET-FILE-LINK DDRLOGR1,ADAyyyyy.RLOGR1,SHARE-UPD=YES
/START-PROGRAM *M(ADA.MOD,ADARUN),PR-MO=ANY
ADARUN PROG=ADANUC,DB=yyyyy
ADARUN LBP=600000
ADARUN LWP=320000
ADARUN LS=80000
ADARUN LP=400
ADARUN NAB=24
ADARUN NC=50
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
/REMARK
/LOGOFF SYS-OUTPUT=DEL
```

### In ISP Format:

```
/.ADANUC LOGON
/OPTION MSG=FB,DUMP=YES
/REMARK *
/REMARK * ADABAS NUCLEUS
/REMARK *
/SYSFILE SYSLST=L.NUC

/FILE ADAvrs.MOD,LINK=DDLIB
/FILE EXAMPLE.ADAyyyyy.ASSOR1,LINK=DDASSOR1,SHARUPD=YES
/FILE EXAMPLE.ADAyyyyy.DATAR1,LINK=DDDATAR1,SHARUPD=YES
/FILE EXAMPLE.ADAyyyyy.WORKR1,LINK=DDWORKR1
/FILE EXAMPLE.ADAyyyyy.PLOGR1,LINK=DDPLOGR1,SHARUPD=YES
/FILE EXAMPLE.ADAyyyyy.PLOGR2,LINK=DDPLOGR2,SHARUPD=YES
/FILE EXAMPLE.ADAyyyyy.CLOGR1,LINK=DDCLOGR1,SHARUPD=YES
/FILE EXAMPLE.ADAyyyyy.CLOGR2,LINK=DDCLOGR2,SHARUPD=YES
/FILE EXAMPLE.ADAyyyyy.RLOGR1,LINK=DDRLOGR1,SHARUPD=YES
/EXEC (ADARUN,ADAvrs.MOD)
ADARUN PROG=ADANUC,DB=yyyyy
ADARUN LBP=600000
ADARUN LWP=320000
ADARUN LS=80000
ADARUN LP=400
ADARUN NAB=24
ADARUN NC=50
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
/REMARK
/LOGOFF NOSPOOL
```

### JCL Required for UES Support (BS2000)

If you are using universal encoding support (UES), the following additional JCL is required for BS2000 environments:

In SDF Format:

```
/SET-FILE-LINK DDECSOJ,ADAvrs.ALLECSO
```

In ISP Format:

```
/FILE ADAvrs.ALLECSO,LINK=DDECSOJ
```

## z/OS Session

The following data sets are required when executing an Adabas session under z/OS.

Data Set	DD Name	Storage Medium	Additional Information
ADARUN parameters	DDCARD	card image	<a href="#">note 1</a>
ADARUN / Adabas messages	DDPRINT	printer	<a href="#">note 2</a>
Associator	DDASSOR $n$ or DDASSO $nn$	disk	<a href="#">note 3</a>
Data Storage	DDDATAR $n$ or DDDATA $nn$	disk	<a href="#">note 3</a>
Work	DDWORKR1 DDWORKR4	disk	<a href="#">note 4</a>
Recovery Aid log	DDRLOGR1	disk	<a href="#">note 5</a>
Protection log	DDSIBA	tape/disk	<a href="#">note 6</a>
multiple log 1	DDPLOGR1	disk	<a href="#">note 7</a>
multiple log 2	DDPLOGR2	disk	<a href="#">note 7</a>
Command log	DDLOG	tape/disk	<a href="#">note 8</a>
multiple log 1	DDCLOGR1	disk	<a href="#">note 9</a>
multiple log 2	DDCLOGR2	disk	<a href="#">note 9</a>
ECS encoding objects	DDECSOJ	tape/disk	<a href="#">note 10</a>
Abnormal termination	MPMDUMP	printer	<a href="#">note 11</a>

Data Set	DD Name	Storage Medium	Additional Information
Abnormal termination (if large buffer pools are used)	SVCDUMP	printer	<a href="#">note 12</a>
SMGT dump and snap dump	ADASNAP	printer	<a href="#">note 13</a>

### 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
//DDPRINT DD SYSOUT=X
//SYSUDUMP DD SYSOUT=X
//MPMDUMP DD SYSOUT=X
//ADASNAP DD SYSOUT=X
//DDCARD 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)

A sample startup job for a UES-enabled database is provided in member ADANUCU in the ADA*vrs*.JOBS data set. If you are using universal encoding support (UES), you must

- include the following additional libraries for internal products in the STEPLIB:

```
//STEPLIB DD DISP=SHR,DSN=ADABAS.APSvrs.LDnn
//          DD DISP=SHR,DSN=ADABAS.APSvrs.LD00
```

where *nn* is the load library level. If the library with a higher level number is not a full replacement for the lower level load library(s), the library with the higher level must precede those with lower numbers in the STEPLIB concatenation.



**Note:** If you are using an Adabas load library prior to version 7.2.2, it contains internal product libraries with an earlier version number and must be ordered below the current internal product libraries in the STEPLIB concatenation.

- add the following additional JCL related to internal product libraries:

```
//DDECSOJ DD DISP=SHR,DSN=ADABAS.ADAvrs.EC00
//SYSPARM DD *
SYSTEM_ID=ADAAPS
ABEND_RECOVERY=NO
THREAD_ABEND_RECOVERY=NO
```

### 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=WATvrs.LOAD
```



**Note:** This library is distributed with Entire Net-Work.

- identify the TCP/IP stack you intend to use with the `CDI_DRIVER` parameter of the `SYSPARM` statement:

```
//DDECSOJ DD DISP=SHR,DSN=ADABAS.ADAvrs.ECOn
//SYSPARM DD *
SYSTEM_ID=ADAAPS
ABEND_RECOVERY=NO
THREAD_ABEND_RECOVERY=NO
* User must choose one of the following depending on the TCP/IP stack used:
*CDI_DRIVER=(`tcpip,PAALSOCK,SUBSYS=ACSS')      <--Interlink TCP/IP stack
*CDI_DRIVER=(`tcpip,PAAISOCK,ADDRSPCE=STACKNAME') <--IBM TCP/IP stack for HPS
*CDI_DRIVER=(`tcpip,PAAOSOCK,ADDRSPCE=STACKNAME') <--IBM TCP/IP stack for OE
```

## 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.DUMPxx`. The dump title is "ADABAS System Dump". 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 your Adabas messages and codes documentation.

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. For APF-authorized jobs, the dump title will be "ADABAS System Dump"; for non-authorized jobs, the title will be "ADABAS Transaction Dump". 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.

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/VM Session

The following table contains all data sets which are used when executing an Adabas session under z/VM.

Data Set	DD Name	Storage Medium	Additional Information
ADARUN parameters	DDCARD	disk	<a href="#">note 1</a>
ADARUN/Adabas messages	DDPRINT	disk/virtual printer	<a href="#">note 2</a>
Associator	DDASSOR $n$ or DDASSO $nn$	disk	<a href="#">note 3</a>
Data Storage	DDDATAR $n$ or DDDATA $nn$	disk	<a href="#">note 3</a>
Work	DDWORKR1 DDWORKR4	disk	<a href="#">note 4</a>
Protection log	DDSIBA	tape/disk	<a href="#">note 6</a>
multiple log 1	DDPLOGR1	disk	<a href="#">note 7</a>
multiple log 2	DDPLOGR2	disk	<a href="#">note 7</a>
Command log	DDLOG	tape/disk	<a href="#">note 8</a>
multiple log 1	DDCLOGR1	disk	<a href="#">note 9</a>
multiple log 2	DDCLOGR2	disk	<a href="#">note 9</a>

Data Set	DD Name	Storage Medium	Additional Information
Abnormal termination	-	disk/virtual printer	<a href="#">note 11</a>
UES configuration parameters	SYSPARM	disk	<a href="#">note 10</a>
UES configuration parameters	CONFIG	disk	<a href="#">note 10</a>
UES trace output	STDOUT	disk	<a href="#">note 10</a>
UES error output	STDERR	disk	<a href="#">note 10</a>

For more information on managing UES support in z/VM environments, read *Managing UES Support of VM Databases*, in *Adabas z/VM Installation Guide*.

## VSE Session

The following table contains all data sets used when executing an Adabas session under VSE. SYSnnn means that any programmer logical unit may be used.

Data Set	File Name	Logical Unit	Storage Medium	Additional Information
ADARUN parameters	none CARD CARD	SYSRDR SYS000 SYSnnn	reader tape disk	<a href="#">note 1</a>
ADARUN/Adabas messages	none	SYSLST	printer	<a href="#">note 2</a>
Associator	ASSOR $n$ or ASSO $nn$	SYSnnn	disk	<a href="#">note 3</a>
Data Storage	DATAR $n$ or DATA $nn$	SYSnnn	disk	<a href="#">note 3</a>
Work	WORKR1	SYSnnn	disk	<a href="#">note 4</a>
Recovery Aid log	RLOGR1	SYSnnn	disk	<a href="#">note 5</a>
Protection log multiple log 1 multiple log 2	SIBA PLOGR1 PLOGR2	SYS014 SYSnnn SYSnnn SYSnnn	tape disk disk disk	<a href="#">note 6</a> <a href="#">note 7</a> <a href="#">note 7</a>
Command log multiple log 1 multiple log 2	LOG CLOGR1 CLOGR2	SYS012 SYSnnn SYSnnn SYSnnn	tape disk disk disk	<a href="#">note 8</a> <a href="#">note 9</a> <a href="#">note 9</a>
ECS encoding objects	DDECSOJ	SYS020 *	tape disk	<a href="#">note 10</a>

## Example of Adabas Session Job Control (VSE)

This job includes multiple protection logging, multiple command logging, and Recovery Aid logging:

```
// ASSGN SYS031, dddd, VOL=ADA001, SHR
// ASSGN SYS032, dddd, VOL=ADA002, SHR
// ASSGN SYS033, dddd, DISK, VOL=ADA003, SHR
// ASSGN SYS034, dddd, VOL=ADA004, SHR
// DLBL ASSOR1, 'EXAMPLE.ADAyyyyy.ASSOR1', 2099/365, DA
// EXTENT SYS031, ADA001, , , 15, 1500
// DLBL DATAR1, 'EXAMPLE.ADAyyyyy.DATAR1', 2099/365, DA
// EXTENT SYS032, ADA002, , , 15, 3000
// DLBL WORKR1, 'EXAMPLE.ADAyyyyy.WORKR1', 2099/365, DA
// EXTENT SYS033, ADA003, , , 15, 600
// DLBL PLOGR1, 'EXAMPLE.ADAyyyyy.PLOGR1', 2099/365, DA
// EXTENT SYS034, ADA004, , , 15, 600
// DLBL PLOGR2, 'EXAMPLE.ADAyyyyy.PLOGR2', 2099/365, DA
// EXTENT SYS034, ADA004, , , 615, 600
// DLBL CLOGR1, 'EXAMPLE.ADAyyyyy.CLOGR1', 2099/365, DA
// EXTENT SYS034, ADA004, , , 1215, 600
// DLBL CLOGR2, 'EXAMPLE.ADAyyyyy.CLOGR2', 2099/365, DA
// EXTENT SYS034, ADA004, , , 1815, 600
// DLBL RLOGR1, 'EXAMPLE.ADAyyyyy.RLOGR1', 2099/365, DA
// EXTENT SYS034, ADA004, , , 1300, 600
// DLBL ADAvCL, 'ADABAS.ADAvrs.LOADLIB', 2099/365
// EXTENT , ADADSK
// EXEC ADARUN, SIZE=ADARUN
ADARUN PROG=ADANUC, SVC=xxx, DEVICE=ddd, 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=ddd
ADARUN NCLOG=2, CLOGSIZE=1800, CLOGDEV=ddd
/*
/*
```

## JCL Required for UES Support (VSE)

The following additional JCL is required for universal encoding support (UES):

```
// ASSGN SYS020,disk,VOL=volume,SHR
// DLBL DDECSOJ,'ADABAS.ADAvrs.ECSLIB'
// EXTENT SYS020
```

## 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 and/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 in a VSE environment, DD statements for ASSOR1 and ASSOR2 are required.

If less than 10 data sets exist for each, you must use the ASSOR\* and DATAR\* DD names in the JCL. For example, if, in a z/OS environment, 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 in a z/OS environment would be 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 (ASSOR1 in VSE) to DDASSOR9 (ASSOR9 in VSE) and DDASSO10 (ASSO10 in VSE) to DDASSO99 (ASSO99 in VSE). And the range of Data Storage DD names that can be used in JCL is DDDATAR1 (DATAR1 in VSE) to DDDATAR9 (DATAR9 in VSE) and DDDATA10 (DATA10 in VSE) to DDDATA99 (DATA99 in VSE).



4. The Adabas Work data sets. The WORKR1 data set is mandatory. If you have Adabas Transaction Manager version 7.5 or later installed, an additional work data set, WORKR4 is also mandatory.

Software AG recommends running the nucleus with DISP=OLD (under z/OS; share not specified for BS2000 and VSE) 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 Version 7.5 or later 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.

- 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.

- The Entire Conversion Services (ECS) objects data set is required for universal encoding support (UES) in all operating environments except for z/VM environments. In z/VM environments, the SYSPARM, CONFIG, STDOUT, and STDERR DD statements are required for UES support. Under z/VM, the installer/user only requires access to the ECS objects via an mdisk/sfs link.
- This data set is used to take an Adabas dump including SVC (except in z/VM environments), 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.

- 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.
- 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.

## 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 multi-client fields. One record per user is stored. The profile table is maintained using Adabas Online System.

### Profile Table Values

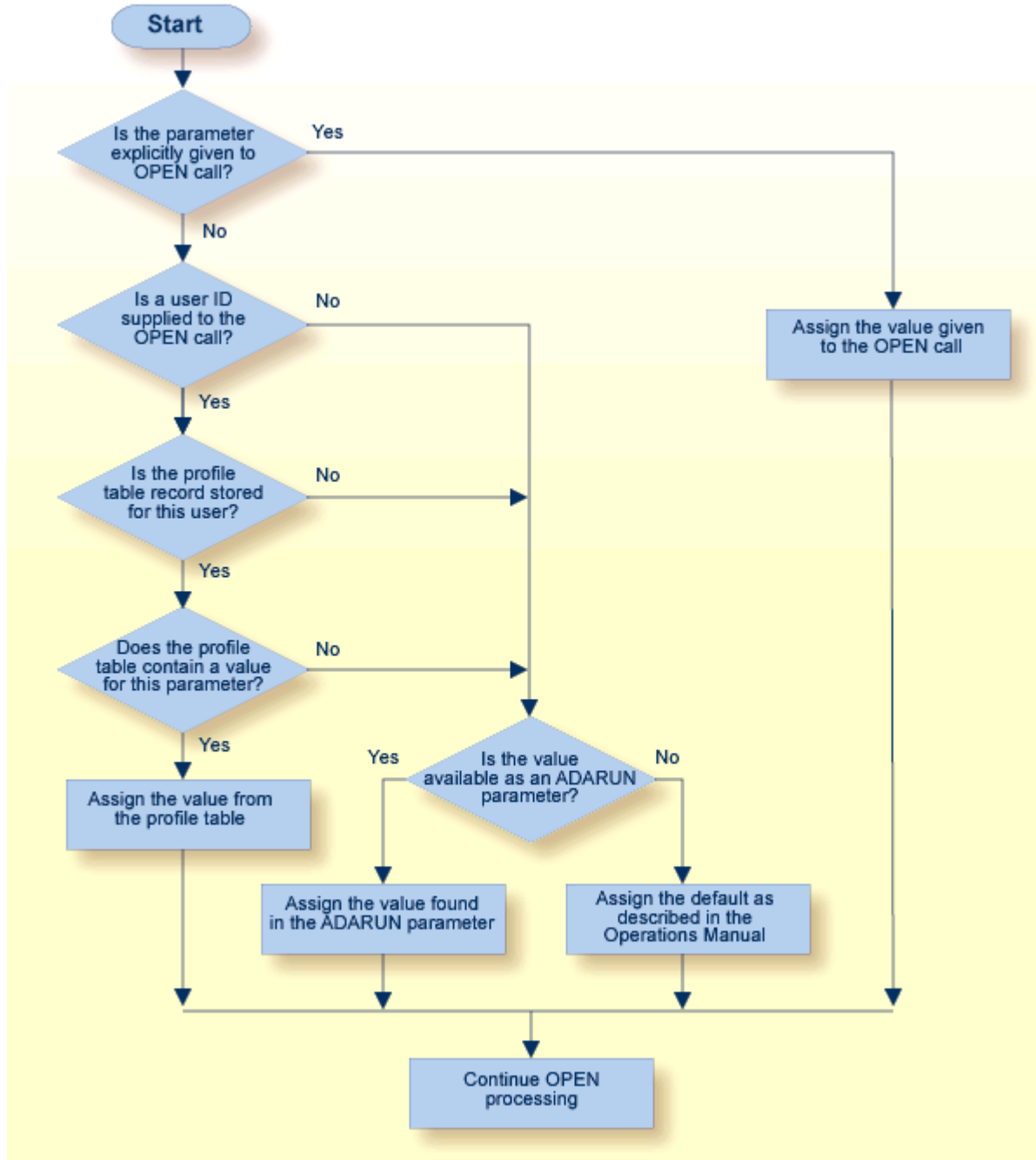
The user-related values shown below are currently stored in the profile table.

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*	Sx 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
OWNERID	Owner ID for multi-client 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.



Profile Table Decision Flow

# 10 Adabas Utility Execution

---

▪ Condition and Termination Codes .....	268
▪ File and Database Usage .....	268
▪ Disk Accesses by Utility and Function .....	271
▪ Possible Resource Conflicts .....	272
▪ Repetitive Utility Execution Under z/VM .....	274

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.
8	encountered an error condition but then continued.
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.

## File and Database Usage

---

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	Adabas Nucleus	DIB Entry Created
ADAACK	All	EXU	P	C	NO
ADACDC	All	n/a	P	C	NO
ADACMP	Compress FDT=...	ACC	P	A	NO
	Compress data	n/a	P	C	NO
	Decompress data	n/a		C	NO
	Decompress INFILE=...	EXU(EXF)		A	NO
ADACNV	All	(1)	NP	N	YES
	With TEST parameter	ACC	P	C	NO
ADADBS	All, except:	ACC or UTI	P <sup>(4)</sup>	A	NO
	RECOVER	UTI	(5)	A	NO
	RESETDIB	n/a	P	C	NO
	OPERCOM DDIB	n/a	P	C	NO
ADADCK	All	EXU	P	C	NO
ADADEF	All	(1)	NP	N	NO
ADAFRM	Formatting other than TEMP/SORT data set	n/a	NP	N	NO
	Formatting TEMP/SORT data set	n/a	P	C	NO
	Formatting single blocks	n/a	P	C	NO
ADAICK	ICHECK	EXU	P	C	NO
ADAINV	All	EXU or UTI	P	A	YES
ADALOD	All	UTI	P	C	YES
ADAMER	ADAM estimation	n/a	P	C	NO
ADAORD	REORDER A/D/DB	(1)	NP	A	YES
	REORDER FILE (A)	EXU or UTI <sup>(2)</sup>	P	A	YES
	REORDER FILE (D)	EXU or UTI <sup>(2)</sup>	P	A	YES
	REORDER FILE	EXU or UTI <sup>(2)</sup>	P	A	YES
	RESTRUCTURE DB	(1)	NP	A	YES
	RESTRUCTURE FILE	EXU <sup>(2)</sup>	P	A	YES
	STORE	UTI	P	A	YES
ADAPLP	All	n/a	P	C	NO
ADAPRI	Maint print	n/a	P	C	NO
ADARAI	DISABLE,	n/a	P	N	NO
	PREPARE, REMOVE CHKDB, LIST RECOVER	n/a	P	C	NO
ADAREP	Database reporting	n/a	P	C	NO
	CPLIST/CPEXLIST	ACC	P	A	NO

Utility	Function	File/DB Usage	Database Activity	Adabas Nucleus	DIB Entry Created
ADARES	Regenerate file,	UTI <sup>(3)</sup>	P	A	NO
	Backout file	(1)	NP	A	NO
	Regenerate,	n/a	P	A/C	NO
	Backout	n/a	P	C	NO
	Copy	UTI	P	A	NO
	Copy dual/multi log Repair				
ADASAV	Save file	UTS	P	C	YES
	Save database	UTS	P <sup>(6)</sup>	C	YES
	Restore file	UTI	P	C	YES
	Restore database	(1)	NP	N	YES
	Restore protection log	(1)	NP	N	YES
ADASEL	Select protection log data	n/a	P	C	NO
ADAULD	Unload file	EXU or EXF	P	A	NO
	Unload file from save tape	n/a	P	C	NO
ADAVAl	Validate	EXU	P	A	NO
ADAZAP	All	(1)	NP	N	NO

The codes used in the File/Database Usage, Database Activity, and Adabas Nucleus columns of this 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.
(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.



## Database Activity Status Codes and Notes

Code	Meaning
NP	Other activity against the database is not permitted.
P	Activity against other database files is permitted.
(4)	When completed, ADADBS ADD/INCREASE end the nucleus session to permit reallocation and formatting.
(5)	ADADBS RECOVER cannot run when other utilities are running.
(6)	When running ADASAV SAVE with an active nucleus (online), no other utilities can be run which update the database.

## Adabas Nucleus Status Codes

Code	Meaning
A	Adabas nucleus must be active.
C	Adabas nucleus may be active.
N	Adabas nucleus must not be active.

## Disk Accesses by Utility and Function

The following table lists the data sets that each utility function accesses on disk:

Utility	Function	Disk Accessed Directly
ADAACK	all	ASSO, DATA
ADACDC	all	ASSO
ADACMP	all	n/a
ADACNV	all	ASSO, DATA, WORK, dual PLOG if specified
ADADBS	OPERCOM DDIB, RESETDIB others	ASSO, if nucleus is down n/a
ADADCK	all	ASSO, DATA
ADADEF	DEFINE NEWWORK	ASSO, DATA, WORK ASSO, WORK
ADAFRM	all	the data set to be formatted
ADAICK	DATAPRINT, DSCHECK others	ASSO, DATA ASSO
ADAINV	all	ASSO
ADALOD	all	ASSO, DATA, WORK
ADAMER	all	n/a

Utility	Function	Disk Accessed Directly
ADAORD	REORDB, REORFILE, REORDATA, REORFDATA, STORE  REORASSO, REORFASSO, RESTRUCTUREDB, RESTRUCTUREF	ASSO, DATA  ASSO
ADAPLP	WORKPRI PLOGPRI others	WORK Dual PLOG n/a
ADAPRI	all	ASSO + data set to be printed
ADARAI	PREPARE, REMOVE all others	ASSO n/a
ADAREP	all  report from SAVETAPE*	ASSO; DATA is not required but is recommended because ADAREP tries to print the volume numbers.  n/a
ADARES	CLCOPY, PLCOPY  all others	Dual CLOG, PLOG; ASSO is recommended but not required  ASSO
ADASAV	SAVE DB, SAVE FILE, RESTORE FILE	ASSO, DATA; also WORK if nucleus is down
ADASEL	all	ASSO
ADAULD	all	n/a
ADAVAL	all	n/a
ADAZAP	all ADAZAP is to DATA ADAZAP is to WORK	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 7 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.

## Repetitive Utility Execution Under z/VM

---

To run the same utility (ADAREP, for example) more than once from an EXEC, the use of EXECOS is strongly recommended; otherwise, utility-dependent program checks can occur. The following example shows how the ADAREP utility could be run under EXECOS:

```
'EXECOS EXEC ADAREP REPCPLST 1'  
'NUCXDROP ADARUN'  
'EXECOS EXEC ADAREP REPCPLST 1'
```

# 11 Linking Applications to Adabas

---

▪ How the Adabas API Works .....	276
▪ Available Link Routines .....	278
▪ Required Work Area .....	279
▪ Required Application Reentrancy Properties .....	281
▪ Adabas Control Block (ACB) Options .....	282
▪ Extended Adabas Control Block (ACBX) Options .....	284
▪ Programming Conventions for Issuing Direct Calls .....	286
▪ Using the CICS COMMAREA or TWA with COBOL Programs .....	291
▪ Using the Adabas API in Batch Mode .....	295
▪ Support for OpenEdition z/OS Adabas Clients .....	299

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.

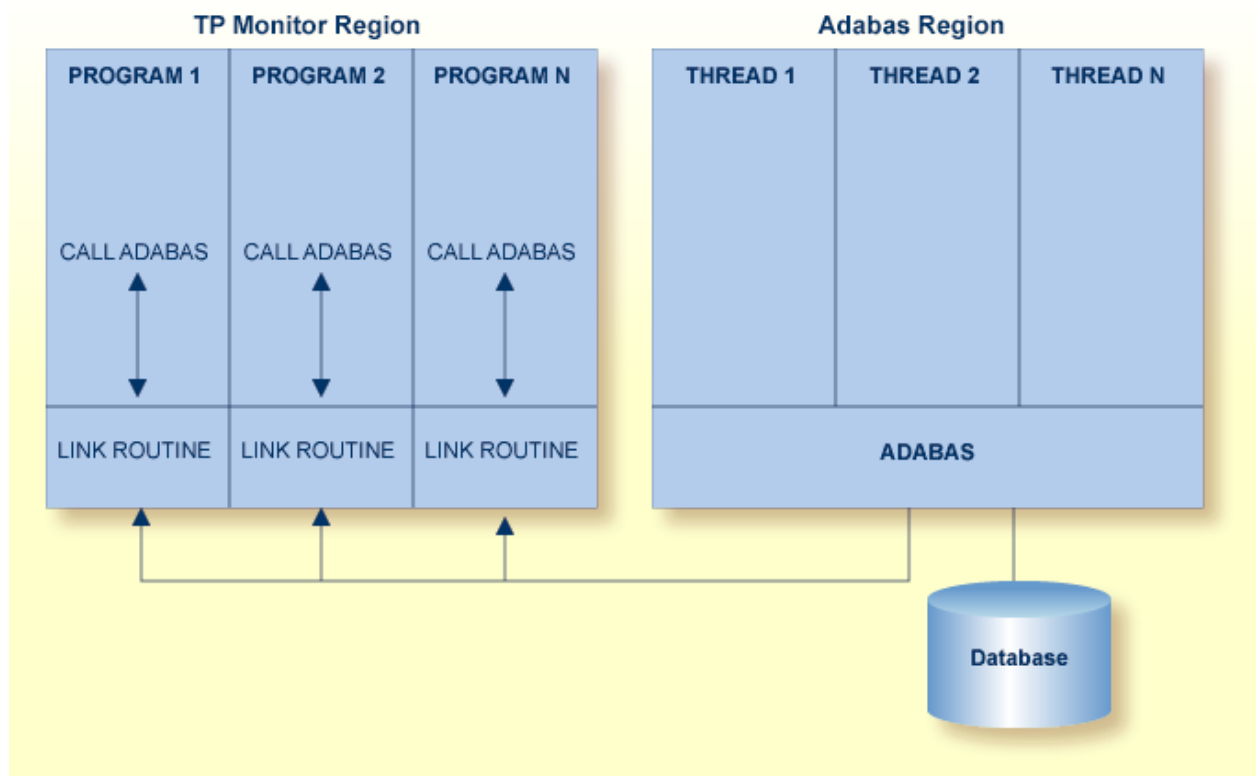
The TP monitor's equivalent of a WAIT is posted when the Adabas function is complete.

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:



### Adabas/TP Monitor Communication

#### 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 or partition, 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.

### For IBM Operating Environments

The following table lists the API versions and the corresponding supported TP monitors available for IBM operating environments:

Link Routine Member	TP Monitor
ADACICS	CICS command-level under Adabas 7 and 8
ADACICT	Adabas 7 and 8 CICS task-related user exit (TRUE)
ADALCO	Executable module for Com-plete under Adabas 8.
ADALCO8	Base module for Com-plete under Adabas 8. 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 8 calls when Com-plete is initialized.
ADALNI	Executable module for IMS TM under Adabas 8 on z/OS systems.
ADALNI8	Base module for IMS TM under Adabas 8 on z/OS systems.
ADALNK	Executable module for batch/TSO under Adabas 7 and 8 on z/OS systems.
ADALNK8	Base module for batch/TSO under Adabas 7 and 8 on z/OS systems.
ADALNKR	Executable reentrant batch/TSO module under Adabas 7 and 8 on z/OS systems
ADALNKR8	Base reentrant batch/TSO module under Adabas 7 and 8 on z/OS systems
LNCSTUB	High-performance CICS stub for Adabas 7 and 8



## For BS2000 Operating Environments

The BS2000 version 7 ADALNK module contains the combined functionality of all the ADALNx modules of previous Adabas versions and contains the following entry points:

Entry Point	TP Monitor
ADALNK	Batch / TIAM
ADALNR	Batch / TIAM (reentrant)
ADALNB	UTM running Natural
ADALNU	UTM with Assembler or a 3GL language
ADAUTM	UTM with SNI BS2000 system Adabas database operations. For more information, read <i>ADAUTM (Universal Transaction Monitor Support)</i> , in the <i>Adabas BS2000 Installation Guide</i> .

## 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.

The Adabas 7 link routines, which are no longer provided with Adabas 8, required that this special work area be obtained by the application program from the TP monitor system. For complete information on using Adabas 7 link routines, refer to your Adabas 7 documentation.



**Note:** If your application program is written in Natural, the necessary API work area is handled without change to the program code.

Adabas 8 link routines do not require that the application program obtain a work area, except if your applications make ACB interface calls to the Adabas 8 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 8 Batch/TSO Reentrant Link Routine \(ADALNKR\) Calling Requirements](#)

- [Mixing ACB and ACBX Interface Direct Calls to ADALNKR](#)

## Adabas 8 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 AG.

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 DDCARD input data (ADARUN cards) or the default database ID value assembled into the link routine at offset X'80'. Applications written in Software AG'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 DDCARD 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.

Software AG recommends 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 AG 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 AG'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 DDCARD 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. Software AG strongly recommends 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 (release 1.4 and above).

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 AG 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**

Different COMMAREA formats are provided, one for Adabas 7 releases and one for Adabas 8.

The format of the Adabas 7 CICS COMMAREA, which must be at least 32 bytes long, is:

COMMID	DC	CL8'ADABAS52'	Adabas COMMAREA ID
CACBA	DC	A(ADACB)	Address of Adabas 7 CB
CAFBA	DC	A(ADAFB)	Address of Adabas 7 FB
CARBA	DC	A(ADARB)	Address of Adabas 7 RB
CASBA	DC	A(ADASB)	Address of Adabas 7 SB
CAVBA	DC	A(ADAVB)	Address of Adabas 7 VB
CAIBA	DC	A(ADAIB)	Address of Adabas 7 IB
		↵	

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 Software AG-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 sections illustrate the JCL/JCS required to link the batch application module with ADAUSER.

### Link Example (BS2000)

```
/ EXEC $TSOSLNK
PROGRAM USERPROG
INCLUDE USERPGM,           ... User Library
INCLUDE ADAUSER,         ... Adabas Library
END
```

### Link Example (z/OS)

```
// 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.

### Link Example (z/VM)

```
FILEDEF ADALIB DISK ADAVnnn LOADLIB fm
FILEDEF SYSLIN DISK LINKEDIT CARDS A
LKED userprog ( NCAL LET XREF MAP LIBE USERLIB LIST
```

The file LINKEDIT CARDS A contains the following linkage editor control statements. The user program exists as a TEXT file.

```
INCLUDE ADALIB(ADABAS)
ENTRY userprog
NAME userprog(R)
```



**Note:** Link with ADAUSER is not applicable when using CMS since this system dynamically loads all necessary modules.

**Link Example (VSE)**

```

* Appropriate assignments must be made for private
libraries, where necessary.
*
// OPTION CATAL
  PHASE USERPROG,*
  INCLUDE USERPGM
  INCLUDE ADAUSER
  ENTRY USREP           (see note)
// EXEC LNKEDT

```



**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 or partition.
- 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 (BS2000)**

In SDF Format:

```

/ASS-SYSDTA *SYSCMD      (ADARUN PARAMETERS)
/SET-FILE-LINK DDLIB,ADAvrs.MOD
/SET-FILE-LINK BLSLIB00,user modlib
/START-PROGRAM USERPROG,PR-MO=ANY,RUN-MODE=ADV(ALT-LIB=YES)
ADARUN MODE=MULTI...

```

In ISP Format:

```
/FILE ADABAS MODLIB, LINK=DDLIB
/SYSFILE TASKLIB=user modlib
/SYSFILE SYSDTA=(SYSCMD) (ADARUN PARAMETERS)
/EXEC USERPROG
ADARUN MODE=MULTI...
```



### Notes:

1. As an alternative to using SYSDTA as the input stream, the user program can assign a sequential file containing the ADARUN parameters to the link name DDCARD using /SET-FILE-LINK (in ISP format, /FILE).
2. Software AG recommends that you link the ADAUSER module to user programs in a TP environment; for example, COBOL.

### Multiuser Mode Example (z/OS)

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
//DDCARD DD *
ADARUN MODE=MULTI
//DDPRINT DD SYSOUT=*
//... user DD statements ...
```

### Multiuser Mode Example (z/VM)

User programs running under z/VM in multiuser mode cannot use ADARUN to control other programs.

### Multiuser Mode Example (VSE)

The following example assumes that the user program USERPROG has been linked with the module ADAUSER and is to be executed in multiuser mode.

```
//...user program assignments...
// LIBDEF PHASE,SEARCH=(user-library, ADABAS-library)
// EXEC USERPROG
ADARUN MODE=MULTI
/*
```

If the user program reads statement input, one of the following applies:

- If all user statements are read before the first Adabas call, they must immediately follow the EXEC statement and be followed by /\*. The user file must be opened, read, and closed before the first Adabas call.
- If the first Adabas call is made before the first user statement is read, the user statements must follow the ADARUN parameter statements and start with a /\* statement.
- Otherwise, the ADARUN parameter statements must be read from file CARD on tape or disk.



#### Notes:

1. ADARUN and/or the Adalink, ADAIOR, ADAMOD, and other Adabas modules must be available for dynamic loading during execution.
2. The Adabas load library must be available during execution so that required modules can be dynamically loaded.

### 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](#).



**Note:** User programs that use z/VM facilities cannot access an Adabas nucleus in single-user mode.

## Support for OpenEdition z/OS Adabas Clients

A client running under OpenEdition z/OS can access Adabas. An OpenEdition application containing calls to Adabas can be linked with ADALNK (option 1) or ADAUSER (option 2).

Software AG recommends that you link your OpenEdition application with ADAUSER (option 2) for the following reasons:

- the application is not tied to a specific database ID and SVC number, or Adabas release;
- the DDPRINT output provides information about the database ID and SVC number used, as well as diagnostic information in case of error (DDPRINT output is lost when using option 1); and
- the program occupies less space in the hierarchical file system (HFS).

### Option 1 : Link OpenEdition Application with ADALNK

An OpenEdition application that contains calls to Adabas can be linked with the module ADALNK. The database ID and SVC number must be zapped into the Adabas CSECT of the linked module at the offsets described in section *Writing User Exits for an Adalink* of the Adabas Installation documentation.

The following sample ZAP and link job has the following steps:

- COPYLNK: copy module ADALNK to another library
- ZAPLNK: zap the copied ADALNK module
- BINDAPP1: link (bind) the application with the zapped ADALNK into OpenEdition

```

/*
/* COPY AND RENAME ADALNK
/*
//
COPYLNK

  EXEC PGM=IEBCOPY
//INLIB DD DSN=ADABAS.load.library,DISP=SHR
//OTLIB DD DSN=ADABAS.lnk.library,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY INDD=INLIB,OUTDD=OTLIB
SELECT MEMBER=((ADALNK,ADALNKOE,R))
/*
/*
/* ZAP DBID AND SVC INTO COPIED ADALNK
/*
//
ZAPLNK

  EXEC PGM=IMASPZAP
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DSN=ADABAS.lnk.library,DISP=SHR
//SYSIN DD *
NAME ADALNKOE ADABAS
VER 0080 0001          DEFAULT DBID 1
VER 0084 0AF9          DEFAULT SVC 249
REP 0080 00D3          <===== CHANGE TO USER DBID (HERE DBID 211)
REP 0084 0AE8          <===== CHANGE TO USER SVC (HERE SVC 232)
/*
/*
/* BIND APPLICATION
/*
//
BINDAPP1

```

```

EXEC PGM=IEWBLINK,
//          PARM='LIST,LET,XREF,MAP,CASE=MIXED'
//SYSPRINT DD  SYSOUT=*
//SYSLMOD  DD  PATH='/u/group/user',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//APPLIB   DD  DSN=your.appl.library,DISP=SHR
//LNKLOAD  DD  DSN=ADABAS.lnk.library,DISP=SHR
//SYSLIN   DD  *
    INCLUDE APPLIB(applname)
    INCLUDE LNKLOAD(ADALNKOE)
    ENTRY applent
    NAME oeappl(R)
/*

```

## Option 2 : Link OpenEdition Application with ADAUSER

An OpenEdition application that contains calls to Adabas can be linked with the module ADAUSER.

Additionally, a member ddcard must be set up in the OpenEdition hierarchical file system (HFS) to contain the ADARUN parameters required by the client; for example:

```
ADARUN  PROG=USER,DBID=211,SVC=232,MODE=MULTI
```

Prior to the first call to Adabas, the application must set the current working directory (using the `chdir()` function, for example) to the directory where file ddcard is located. As the application runs, Adabas searches the current working directory for member ddcard, and extracts the parameters. Additionally, Adabas directs the DDPRINT output to member ddprint of the current working directory.



**Note:** Member names ddcard and ddprint are case-sensitive. Member name DDCARD is not valid and will be ignored.

The following sample link job has one step:

- BINDAPP2: link (bind) the application with ADAUSER into OpenEdition

```

/*
/* BIND APPLICATION
/*
//
BINDAPP2

EXEC PGM=IEWBLINK,
//          PARM='LIST,LET,XREF,MAP,CASE=MIXED'
//SYSPRINT DD  SYSOUT=*
//SYSLMOD  DD  PATH='/u/group/user',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),

```

```
//      PATHMODE=(SIRWXU,SIRWXG,SIRWXO)
//APPLIB DD DSN=your.appl.library,DISP=SHR
//ADALOAD DD DSN=ADABAS.load.library,DISP=SHR
//SYSLIN DD *
        INCLUDE APPLIB(applname)
        INCLUDE ADALOAD(ADAUSER)
        ENTRY applent
        NAME oeappl(R)
/*
```

### Setting the OpenEdition Shell Variable STEPLIB

For both options, the OpenEdition shell variable STEPLIB must be set to ensure access to the Adabas load library. The following sample job sets the variable from OpenEdition running in batch mode:

```
/*
//OEATCH EXEC PGM=BPXBATCH,
//          PARM='PGM /u/group/user/oeappl '
//STDIN   DD
PATH='/u/group/user/oeappl.in',PATHOPTS=(ORDONLY)
//STDOUT  DD  PATH='/u/group/user/oeappl.out',
//          PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
//STERR   DD  PATH='/u/group/user/oeappl.err',
//          PATHOPTS=(OWRONLY,OCREAT),PATHMODE=SIRWXU
//STDENV  DD  *
STEPLIB=ADABAS.load.library
/*
//
```

### Limitations for OpenEdition Support

Support is *not* available for running the following under OpenEdition:

- the Adabas nucleus or utilities
- clients running in single-user mode (MODE=SINGLE)
- clients running in 24-bit addressing mode (AMODE 24)



# 12 Adabas Restart and Recovery

---

▪ Work Data Set .....	304
▪ Protection Log .....	305
▪ Restart Operations .....	307
▪ Database Recovery .....	309

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 (DD/SIBA) data set; or
- to multiple protection log (DD/PLOGR1, DD/PLOGR2, ... DD/PLOGR8) 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 AG 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 (DD/PLOGR $n$  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 DD/PLOGR1 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 (DD/SIBA).

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 (or PLOGRn)
```

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, Software AG strongly recommends 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:

**ADAN58**                    **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 under the following headings:

- [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, Software AG recommends 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

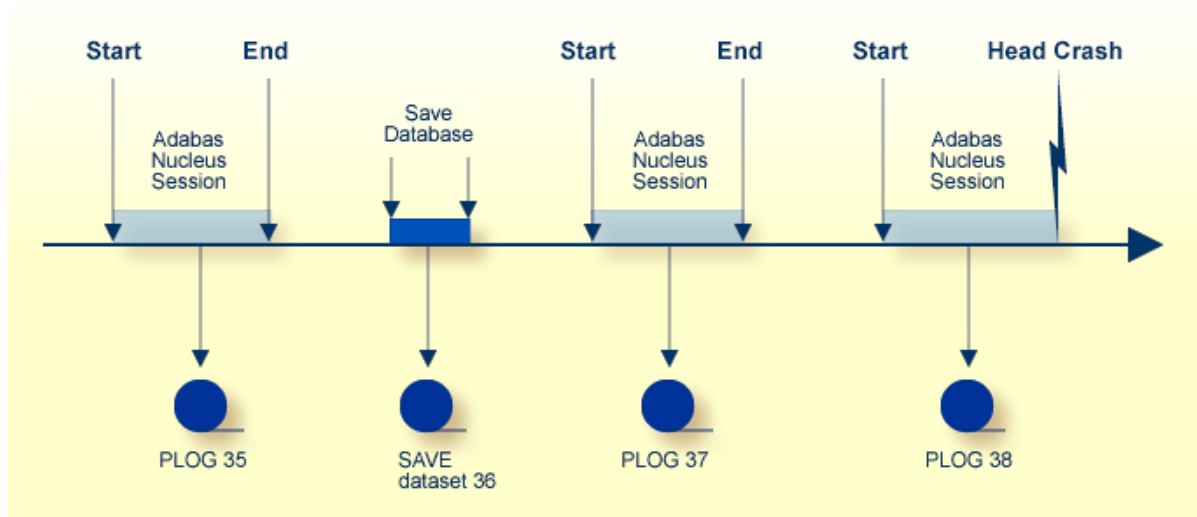
#### ► To recreate the database to the status of the failure

- 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.

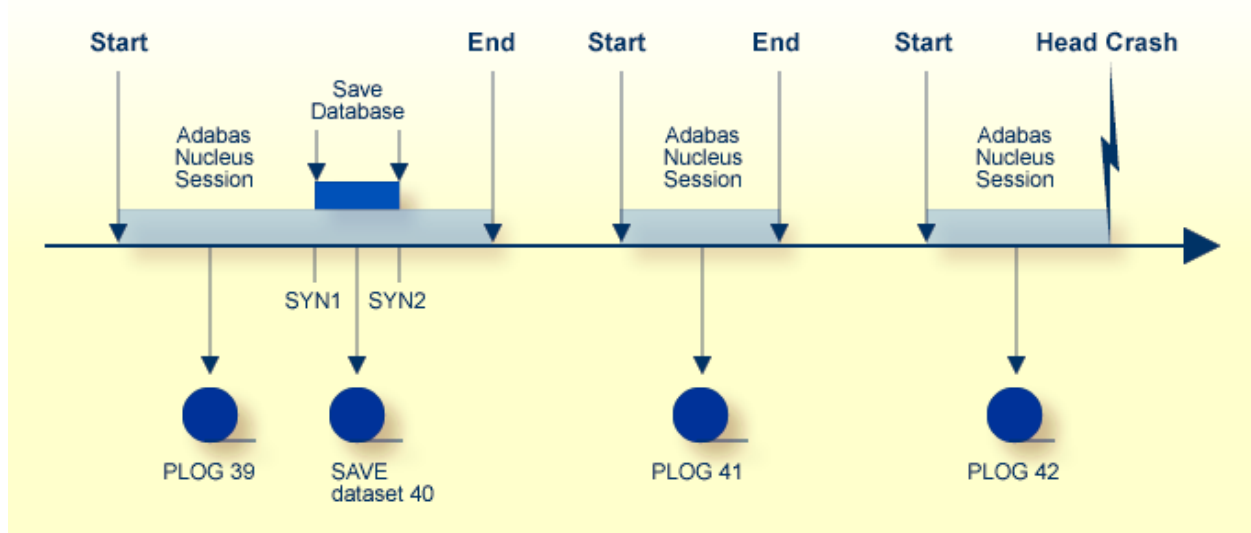
The nucleus writes a protection log number 39.

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/JCS), 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.



**Note:** The job stream generation function is not yet available under VSE or z/VM.

# 13 Local Time Zone Change Support

---

- Time Stamp Overview ..... 316
- Activating the Local Time Zone Change ..... 316

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.

Although this applies to all mainframe platforms, a local time zone change on z/VM or VSE is still reflected only after a shut down and restart of the Adabas nucleus.

## 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 AG 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 and BS2000 systems, 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, Software AG recommends 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.

- 3 On z/VM or VSE systems, shut down and restart the Adabas nucleus.





# 14 Adabas Timeout Feature

---

- Timeout Parameters ..... 320
- Action Taken If Time Limit Exceeded ..... 321

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<sub>x</sub> 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 TNA<sub>x</sub> 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.

This action is *not* taken for TNA<sub>x</sub> 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 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.

# Index

---

## A

- ABDs
  - enable/disable logging of, 123
- ACBX
  - enable/disable logging of, 123
- Adabas
  - overview of operating structure, 5
- Adabas buffer descriptions (ABDs)
  - enable/disable logging of, 123
- Adabas command logged buffer size limit
  - ADARUN parameter, 81
- Adabas control block
  - logging
    - command to start, 232
    - command to stop, 234
- Adabas Delta Save Facility
  - display status
    - using operator command, 223
  - parameter to enable/disable, 105
  - parameter to specify user exit routine, 106
- Adabas error handling and message buffering facility
  - commands, 244
- Adabas Fastpath
  - parameter to enable/disable, 109
- Adabas Parallel Services
  - ADARUN parameter to enable, 89
- Adabas Recovery Aid
  - function of, 313
- Adabas Review
  - command to deactivate, 238
  - hub ID
    - command to set/modify, 238
  - local mode
    - command to switch to, 238
  - parameter to set, 167
- Adabas sysplex cluster
  - nucleus ID
    - parameter to specify, 152
- Adabas Transaction Manager
  - ATM manager logging option, 185
  - external transaction coordinator support, 187
  - global transaction time limit, 184
  - identifying databases for ET data, 183
  - restart action control, 186
  - severity level for suppression of warning messages, 186
  - size of the ATM request queue, 182
  - TCID client ID prefix, 188
- Adabas Triggers and Stored Procedures
  - parameter to activate, 175
- Adabas Vista
  - parameter to enable/disable, 193
- Adabus Cluster Services
  - ADARUN parameter to enable, 89
- ADACICS
  - CICS command-level link routine for Adabas 7 and 8, 278
- ADACICT
  - link routine Adabas 7 and 8 CICS task-related user exit (TRUE), 278
- ADAEND
  - operator command, 218
- ADAIOR
  - blocksize of sequential data set
    - parameter to specify, 164
- ADALCO
  - link routine for Com-plete, 278
- ADALCO8
  - link routine for Com-plete, 278
- Adalink
  - definition of, 276
- ADALNI link routine
  - for IMS TM, 278
- ADALN18 link routine
  - for IMS TM, 278
- ADALNK link routine
  - for batch and TSO, 278
- ADALNK8 link routine
  - for batch and TSO, 278
- ADALNKR link routine
  - for batch and TSO, 278
- ADALNKR8 link routine
  - for batch and TSO, 278
- ADALOD utility
  - example of DSTAT command usage, 243
- ADAORD utility
  - example of DSTAT command usage for, 243
- ADARES utility
  - example of DSTAT command usage for, 243
- ADARUN
  - directory of parameters, 35, 203
  - guidelines for setting parameters, 195
  - parameter syntax, 33
  - parameters specific to an operating system, 200
  - using session statistics to determine settings, 200
- ADARUN parameters
  - Logged Buffer Size Limit for Review, 168
  - LRPL, 127
  - REVLOGBMAX parameter, 168
  - REVLOGMAX parameter, 169

RPLCONNECTCOUNT, 169  
 RPLCONNECTINTERVAL, 170  
 RPLPARMS, 170  
 RPLSORT, 171  
 RPWARNINCREMENT, 172  
 RPWARNINTERVAL, 172  
 RPWARNMESSAGELIMIT, 173  
 RPWARNPERCENT, 173  
 Total Logged Buffer Size Limit for a Review Command, 169

ADASAV utility  
 abnormal termination  
 command to check status, 238  
 example of DSTAT command usage for, 244

ADASNAP data set, 246-247, 264

ADAULD utility  
 example of DSTAT command usage for, 244

ADAUSER  
 Adabas `stub` program, 277  
 link with Adabas API, 295

ALOCKF  
 operator command, 219

AOSLOG  
 ADARUN parameter, 62  
 operator command  
 using console, 219

API  
 batch operation, 295  
 build the ACBX, 285  
 build the control block, 283  
 code reentrancy requirements, 281  
 control block options, 282, 284  
 database IDs, 285  
 file numbers, 285  
 issue direct calls to Adabas, 286  
 link applications to Adabas, 276  
 one-byte file numbers, 283  
 passing parameters to, 279  
 pointer to the Adabas control block (ACB), 282  
 pointer to the extended Adabas control block (ACBX), 284  
 required work area, 279  
 support for OpenEdition, 299  
 limitations, 302  
 two-byte file numbers, 283  
 versions of, 278  
 work area defined by DSECTs, 279

AREXCLUDE  
 ADARUN parameter, 62

ARM  
 parameter to activate, 63

ARMNAME  
 ADARUN parameter, 63

ASSOCACHE  
 ADARUN parameter, 63

Associator  
 parameter to  
 specify device type for storing first block, 103

Associator data set  
 parameter to enable/disable controller caching for, 63

ASYTVS  
 ADARUN parameter, 64  
 operator command  
 using console, 219

attached buffer  
 parameter to set time limit for hold, 98

attached buffers  
 command to display usage, 228  
 number of  
 parameter to specify, 143

autobackout  
 definition of, 308

Automatic Restart Management (ARM)  
 explanation, 309

autorestart  
 definition of, 308  
 parameter to activate ARM, 63  
 parameter to exclude files from processing, 62

## B

batch  
 ADALNK link routine, 278  
 ADALNK8 link routine, 278  
 ADALNKR link routine, 278  
 ADALNKR8 link routine, 278  
 API work area for, 279  
 process a request to Adabas, 277  
 using the Adabas API in, 295

bold, 2

braces ({}), 3

brackets ([]), 3

BS2000  
 Adabas data sets required for, 253  
 ADARUN CMADDR parameter, 90  
 ADARUN CMFIX parameter, 92  
 ADARUN CMLADDR parameter, 90  
 ADARUN CMLFIX parameter, 92  
 ADARUN CMLSCOPE parameter, 94  
 ADARUN CMLSIZE parameter, 95  
 ADARUN CMSCOPE parameter, 94  
 ADARUN CMSIZE parameter, 95  
 batch or TP task priority  
 parameter to specify, 178  
 entering operator commands, 214  
 interprocess communication  
 parameter to limit to users with same logon ID, 112  
 parameter for subtask GETMAIN memory pool, 176

buffer flush  
 asynchronous  
 based on volume serial number, parameter to disable, 64  
 operator command to activate using console, 219  
 parameter to enable and set I/O pool size, 119  
 parameter to limit number of parallel I/O operations, 110  
 modify number of I/Os, 231  
 synchronous  
 parameter to specify duration of, 181

buffer pool  
 Adabas  
 parameter to set maximum size of, 116  
 attached  
 space allocation, 143  
 internal format  
 parameter to set size of, 120

buffers  
 Adabas command logged buffer size limit, 81  
 logged buffer size limit, 79

## C

## CACHE

- ADARUN parameter, 64, 75

## cache

- shared cache, 86

## cache structure

- name, 83

- ratio

- directory entries to elements, 104

## caching

## controller

- parameter to enable/disable controller caching for Associator data set, 63

- parameter to enable/disable controller caching for Data Storage data set, 102

- parameter to enable/disable controller caching for Sort data set, 174

- parameter to enable/disable controller caching for Temp data set, 180

- parameter to enable/disable controller caching for Work data set, 194

## caching facility

- Associator cache space unit size, 67

- Data Storage cache space unit size, 72

- excluding files from caching, 101

- lowest Adabas buffer efficiency before activation, 74

- number of storage areas allocated, 91

- parameter to activate RABN ranges, 65

- parameter to control data space RABN caching, 71

- parameter to control extended memory RABN caching, 71

- parameter to control hiperspace RABN caching, 71

- 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 virtual 64 storage RABN caching, 71

- parameter to enable, 64, 75, 77

- parameter to exclude command types, 76

- 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

## CANCEL

- operator command, 220

## CASSODSP

- ADARUN parameter, 66

## CASSOEXT

- ADARUN parameter, 66

## CASSOHSP

- ADARUN parameter, 66

## CASSOMAXS

- ADARUN parameter, 67

## CASSOV64

- ADARUN parameter, 66

## CBUFNO

- ADARUN parameter, 69

## CCTIMEOUT

- ADARUN parameter, 70

## CDATADSP

- ADARUN parameter, 71

## CDATAEXT

- ADARUN parameter, 71

## CDATAHSP

- ADARUN parameter, 71

## CDATAMAXS

- ADARUN parameter, 72

## CDATAV64

- ADARUN parameter, 71

## CDEMAND

- ADARUN parameter, 74

## CDXnn

- ADARUN parameter, 75

## CEXCLUDE

- ADARUN parameter, 76

## CFILE

- ADARUN parameter, 77

## checkpoints

- SYNS 60

- parameter to set time interval between, 116

## choices in syntax, 3

## CICS

- Adabas call using, 287

- code reentrancy requirements, 282

- issuing direct calls to Adabas through, 287

- LNCSUB link routine for command level, 278

## CLOGBMAX

- ADARUN parameter, 79

## CLOGDEV

- ADARUN parameter, 79

## CLOGLAYOUT

- ADARUN parameter, 80

## CLOGMAX

- ADARUN parameter, 81

## CLOGMRG

- ADARUN parameter, 82

- operator command, 220

## CLOGSIZE

- ADARUN parameter, 79

## CLUCACHENAME

- ADARUN parameter, 83

## CLUCACHESIZE

- ADARUN parameter, 84

## CLUCACHETYPE

- ADARUN parameter, 85

## CLUCACHEUNCHANGED

- ADARUN parameter, 86

## CLUGROUPNAME

- ADARUN parameter, 87

## CLULOCKNAME

- ADARUN parameter, 88

## CLULOCKSIZE

- ADARUN parameter, 89

## CLUSTER

- ADARUN parameter, 89

## cluster

- group name, 87

- redo updates

- parameter to set size of pool, 126

- CMADDR
    - ADARUN parameter, 90
  - CMAXCSPS
    - ADARUN parameter, 91
  - CMDQMODE
    - ADARUN parameter, 92
  - CMFIX
    - ADARUN parameter, 92
  - CMLADDR
    - ADARUN parameter, 90
  - CMLFIX
    - ADARUN parameter, 92
  - CMLSCOPE
    - ADARUN parameter, 94
  - CMLSIZE
    - ADARUN parameter, 95
  - CMSCOPE
    - ADARUN parameter, 94
  - CMSIZE
    - ADARUN parameter, 95
  - COBOL
    - set the control byte, 284
  - collation descriptor
    - user exit
      - parameter to specify ID and name, 75
  - Com-plete
    - ADALCO link routine, 278
    - ADALCO8 link routine, 278
    - API work area, 279
    - issuing direct calls to Adabas through, 286
    - reentrancy requirements, 282
  - command ID
    - count per user
      - parameter to set maximum, 149
      - requirement for OPEN with active user profile, 265
  - Command log
    - command to start, 232
    - command to stop, 234
    - dual
      - parameter to specify device type, 107
      - parameter to specify size, 107
    - multiple
      - parameter to specify device type, 79
      - parameter to specify size, 79
      - parameter to enable merge, 82
      - parameter to enable/disable, 123
      - parameter to specify format of, 80
    - sequential
      - parameter to set blocksize, 124
  - command log
    - dual
      - switch to other, 230
    - extension
      - enable/disable logging of, 123
  - Command log data sets
    - maximum number of
      - parameter to specify, 145
  - Command log merge
    - command to switch on or off, 220
  - command queue
    - command to display usage, 228
    - parameter to specify location of memory pool, 92
  - command queue element
    - maximum number of
      - parameter to specify, 144
      - parameter to set time limit for hold, 98
  - command queue element (CQE)
    - display posted, 225
  - command queue element (DCQ)
    - display all posted
      - using operator command, 222
  - commands
    - display total processed for a file, 224
    - exclude from prefetch/multifetch
      - using parameter, 161
    - include in prefetch/multifetch
      - using parameter, 157
    - log update to DD/PRINT
      - using console, 219
    - operator, 213
      - entering under BS2000, 214
      - entering under VSE, 215
      - entering under z/OS, 215
      - under z/VM, 215
    - parameter to
      - set time limit for completion, 98
  - communication
    - cluster nuclei
      - timeout threshold for canceled peer nucleus, 134
      - timeout threshold for command, 136
  - conventions
    - syntax, 2
  - CRETRY
    - ADARUN parameter, 96
  - CSTORAGE
    - ADARUN parameter, 97
  - CT
    - ADARUN parameter, 98
    - operator command, 222
  - CWORK2FAC
    - ADARUN parameter, 99
  - CWORK3FAC
    - ADARUN parameter, 100
  - CWORKSTORAGE
    - ADARUN parameter, 98
  - CXFILE
    - ADARUN parameter, 101
- ## D
- Data Integrity Block (DIB)
    - parameter to
      - ignore active nucleus entry in, 114
  - data integrity block (DIB)
    - operator command to display, 222
  - data protection area
    - parameter to allocate blocks for, 125
  - data sets
    - required fo z/VM, 259
    - required for BS2000, 253
    - required for VSE
      - VSE, 260
    - required for z/OS, 255
  - Data Storage data set
    - parameter to enable/disable controller caching for, 102
  - database
    - re-creating after failure, 309
    - repair after failure, 308



- restarting after failure, 309
  - DATACACHE
    - ADARUN parameter, 102
  - DAUQ
    - operator command, 222
  - DBID
    - ADARUN parameter, 103
  - DCQ
    - operator command, 222
  - DDIB
    - operator command, 222
  - DDSF
    - operator command, 223
  - default parameter values, 2
  - DELUF
    - operator command, 223
  - DELUI
    - operator command, 223
  - descriptor
    - collation
      - parameter to specify user exit for, 75
      - unique
        - parameter to set size of pool, 118
  - DEVICE
    - ADARUN parameter, 103
  - device types
    - reserved for VSAM, 103
  - DFILES
    - operator command, 224
  - DFILUSE
    - operator command, 224
  - DHQ
    - operator command, 224
  - DHQA
    - operator command, 225
  - direct calls
    - issuing through IMS TM, 291
    - using API, 286
      - CICS, 287
      - CICS command level examples, 288-289
      - CICS macro level example, 288
      - Com-plete examples, 286
      - IMS TM, 291
      - IMS TM nonreentrant examples, 291
  - DIRRATIO / ELEMENTRATIO
    - ADARUN parameter, 104
  - distributed transaction processing
    - parameter to
      - ignore data in Work part 4, 115
      - parameter to enable/disable, 106
  - DLOCKF
    - operator command, 225
  - DNC
    - operator command, 225
  - DNFV command
    - display nucleus file variables, 225
  - DNH
    - operator command, 226
  - DNU
    - operator command, 226
  - DONLSTAT
    - operator command, 226
  - DPARM
    - operator command, 227
  - DPPT command
    - display nucleus's own PPT block, 227
  - DRES
    - operator command, 228
  - DSF
    - ADARUN parameter, 105
  - DSFEX1
    - ADARUN parameter, 106
  - DSTAT
    - operator command, 229
    - utilities, 243
  - DTH
    - operator command, 229
  - DTP
    - ADARUN parameter, 106
  - DUALCLD
    - ADARUN parameter, 107
  - DUALCLS
    - ADARUN parameter, 107
  - DUALPLD
    - ADARUN parameter, 108
  - DUALPLS
    - ADARUN parameter, 108
  - dump
    - command to check status, 238
  - DUQ
    - operator command, 229
  - DUQA
    - operator command, 229
  - DUQE
    - operator command, 230
  - DUUQE
    - operator command, 230
- ## E
- ellipsis (...), 3
  - error handling and message buffering facility
    - parameter to activate, 174
  - ETID
    - use with user profile table, 265
  - EXCPVR
    - controlling page fixing, 154
  - EXECOS
    - recommended use
      - under z/VM, 274
  - extended Adabas control block (ACBX)
    - enable/disable logging of, 123
- ## F
- failure
    - types of, 307
  - FASTPATH
    - ADARUN parameter, 109
  - FEOFCL
    - operator command, 230
  - FEOFPL
    - operator command, 230
  - file
    - advance lock
      - command to remove, 237
      - command to set, 219
  - files

- exclude from prefetch/multifetch
  - using parameter, 161
- include in prefetch/multifetch
  - using parameter, 157
- lock
  - using operator command, 231
- locked
  - display using operator command, 225
- parameter to exclude from
  - autorestart processing, 62
  - re-creating after failure, 310
- stop users of
  - using operator command, 238
- unlock
  - using operator command, 242
- FMXIO
  - ADARUN parameter, 110
- FMXIO=nn
  - operator command, 231
- FORCE
  - ADARUN parameter, 111
- format buffer
  - enable/disable logging of, 123
  - logging
    - command to start, 232
    - command to stop, 235
- format pool
  - command to display usage, 228

## G

- global cache area
  - construct type, 85
  - name, 83
  - ratio
    - directory entries to elements, 104
    - size, 84
- global lock area
  - name, 88
  - size, 89
- GROUPS
  - ADARUN parameter, 112

## H

- HALT
  - operator command, 231
- HEXnn
  - ADARUN parameter, 113
- hold queue
  - command to display usage, 228
  - count of ISNs in
    - parameter to set maximum, 146
- hold queue element
  - display, 224
  - maximum number of
    - parameter to specify, 146
- hyperdescriptor
  - parameter to
    - specify user exit ID and name, 113

## I

- I/O activity

- enable/disable logging of, 123
- logging
  - command to start, 233
  - command to stop, 235
- ID Table
  - parameter to
    - allow nucleus to overwrite existing entry, 111
    - specify an alternate, 113
- IDT name
  - suppressing in messages, 114
- IDTNAME
  - ADARUN parameter, 113
- IDTPSUP
  - ADARUN parameter, 114
- IGNDIB
  - ADARUN parameter, 114
- IGNDTP
  - ADARUN parameter, 115
- IMS
  - Adabas call structure for use with, 291
  - API work area, 279
- IMS TM
  - ADALNI link routine, 278
  - ADALNI8 link routine, 278
- indentation, 3
- INTNAS
  - ADARUN parameter, 116
- ISN
  - count in hold queue
    - operator command to display, 226
    - parameter to set maximum, 146
  - count per TBI element
    - parameter to set maximum, 149
  - list processing area
    - space allocation using parameter, 130
- ISN buffer
  - enable/disable logging of, 123
  - logging
    - command to start, 233
    - command to stop, 235
- italic, 2

## L

- LBP
  - ADARUN parameter, 84, 116
- LCP
  - ADARUN parameter, 117
- LDEUQP
  - ADARUN parameter, 118
- LDTP
  - ADARUN parameter, 118
- LFIOP
  - ADARUN parameter, 119
- LFP
  - ADARUN parameter, 120
- LI
  - ADARUN parameter, 121
- link globals table
  - name, 122
- link routines
  - corresponding to TP monitors, 278-279
- LNCSTUB
  - link routine for CICS (command-level), 278

LNKGNAME  
 ADARUN parameter, 122

LOCAL  
 ADARUN parameter, 122

local buffer pool  
 size parameter, 84

lock  
 file in advance  
 command to remove, 237  
 command to set, 219

lock structure  
 name, 88

LOCKF  
 operator command, 231

LOCKU  
 operator command, 231

LOCKX  
 operator command, 232

LOGABDX  
 ADARUN parameter, 123

LOGCB  
 ADARUN parameter, 123  
 operator command, 232

LOGCLEX  
 ADARUN parameter, 123

LOGFB  
 ADARUN parameter, 123  
 operator command, 232

logged buffer size limit  
 ADARUN parameter, 79

LOGGING  
 ADARUN parameter, 123  
 operator command, 232

LOGIB  
 ADARUN parameter, 123  
 operator command, 233

LOGIO  
 ADARUN parameter, 123  
 operator command, 233

LOGMB  
 ADARUN parameter, 123

LOGRB  
 ADARUN parameter, 123  
 operator command, 233

LOGSB  
 ADARUN parameter, 123  
 operator command, 233

LOGSIZE  
 ADARUN parameter, 124

LOGUX  
 ADARUN parameter, 124  
 operator command, 234

LOGVB  
 ADARUN parameter, 124  
 operator command, 234

lowercase, 2

LP  
 ADARUN parameter, 125

LQ  
 ADARUN parameter, 125

LRDP  
 ADARUN parameter, 126

LRPL parameter, 127

LS

ADARUN parameter, 128

LU  
 ADARUN parameter, 129

LWKP2  
 ADARUN parameter, 130

LWP  
 ADARUN parameter, 131

## M

message buffer area  
 parameter to set size of, 132

message case  
 parameter to set, 133

minimum keywords, 2

mixed case, 2

MODE  
 ADARUN parameter, 131

modes of operation  
 parameter to specify, 131

MSGBUF  
 ADARUN parameter, 132

MSGCONSL  
 ADARUN parameter, 133

MSGDRUCK  
 ADARUN parameter, 133

MSGPRINT  
 ADARUN parameter, 133

multifetch  
 automatic activation  
 parameter to specify, 156  
 buffer length per call  
 parameter to set, 159  
 count of records to obtain  
 parameter to set maximum, 159  
 exclude commands from  
 using parameter, 161  
 exclude files from  
 using parameter, 161  
 include commands in  
 using parameter, 157  
 include files in  
 using parameter, 157  
 total buffer length  
 parameter to set, 160

multifetch buffer  
 enable/disable logging of, 123

multiprocessing  
 nucleus ID for  
 parameter to specify, 152

multiuser  
 operating mode  
 parameter to specify, 131

MXCANCEL  
 ADARUN parameter, 134

MXCANCELWARN  
 ADARUN parameter, 135

MXMSG  
 ADARUN parameter, 136

MXMSGWARN  
 ADARUN parameter, 137

MXSTATUS  
 ADARUN parameter, 138

MXTNA

- ADARUN parameter, 139
- MXTSX
  - ADARUN parameter, 140
- MXTT
  - ADARUN parameter, 141
- MXWTOR
  - ADARUN parameter, 142

## N

- NAB
  - ADARUN parameter, 143
- Natural
  - applications written in, 279
- NC
  - ADARUN parameter, 144
- NCLOG
  - ADARUN parameter, 145
- NH
  - ADARUN parameter, 146
- NISNHQ
  - ADARUN parameter, 146
- NOLOGCB
  - operator command, 234
- NOLOGFB
  - operator command, 235
- NOLOGGING
  - operator command, 234
- NOLOGIB
  - operator command, 235
- NOLOGIO
  - operator command, 235
- NOLOGRB
  - operator command, 235
- NOLOGSB
  - operator command, 236
- NOLOGUX
  - operator command, 236
- NOLOGVB
  - operator command, 236
- NONDES
  - ADARUN parameter, 147
- normal font, 2
- NPLOG
  - ADARUN parameter, 148
- NQCID
  - ADARUN parameter, 149
- NSISN
  - ADARUN parameter, 149
- NT
  - ADARUN parameter, 151
- NU
  - ADARUN parameter, 152
- NUCID
  - ADARUN parameter, 152
- Nucleus
  - TCP/IP URL
    - parameter to set, 179
- nucleus
  - BS2000 batch or TP task priority
    - parameter to specify, 178
  - command to display current status, 229
  - control TCP/IP access using TCPIP parameter
    - using parameter, 179

- ID for multiprocessing
  - parameter to specify, 152
- isolated
  - parameter to define as a local nucleus, 122
- number of threads per
  - parameter to set maximum, 151
- operator commands, 216
- overview in Adabas operation, 5
- SVC for
  - parameter to specify, 176
- TCP/IP link to
  - command to open/close, 240
- nucleus file variables
  - display, 225

## O

- online operation
  - process a request to Adabas, 276
- online process
  - display status of, 226
  - resume a suspended process, 236
  - stop cleanly, 237
  - suspend, 237
- ONLRESUME
  - operator command, 236
- ONLSTOP
  - operator command, 237
- ONLSUSPEND
  - operator command, 237
- open command
  - requirement for
    - parameter to specify, 153
- OpenEdition
  - Adabas API support for, 299
  - limitations, 302
- OPENRQ
  - ADARUN parameter, 153
- operations
  - overview of Adabas, 5
- operator commands
  - DNFV
    - display nucleus file variables, 225
  - DPPT
    - display nucleus's own PPT block, 227
- optional syntax elements, 3

## P

- page fixing, controlling
  - EXCPVR runs, 154
- PGFIX
  - ADARUN parameter, 154
- PL/I
  - compiler code reentrancy, 281
- PLOGDEV
  - ADARUN parameter, 154
- PLOGRQ
  - ADARUN parameter, 155
- PLOGSEGSDS
  - ADARUN parameter, 155
- PLOGSIZE
  - ADARUN parameter, 154
- PPT

- display nucleus's own block, 227
- PREFETCH
  - ADARUN parameter, 156
- prefetch
  - automatic activation
    - parameter to set, 156
  - buffer length per call
    - parameter to set, 159
  - count of records to obtain
    - parameter to set maximum, 159
  - exclude commands from
    - using parameter, 161
  - exclude files from
    - using parameter, 161
  - include commands in
    - using parameter, 157
  - include files in
    - using parameter, 157
  - total buffer length
    - parameter to set, 160
- PREFICMD
  - ADARUN parameter, 157
- PREFIFIL
  - ADARUN parameter, 157
- PREFNREC
  - ADARUN parameter, 159
- PREFSBL
  - ADARUN parameter, 159
- PREFTBL
  - ADARUN parameter, 160
- PREFXCMD
  - ADARUN parameter, 161
- PREFXFIL
  - ADARUN parameter, 161
- profile table
  - for user-related information, 265
  - maintenance, 266
  - values, 265
- PROGRAM
  - ADARUN parameter, 162
- Protection log
  - description of, 305
  - dual
    - description of, 306
    - parameter to specify device type, 108
    - parameter to specify size, 108
    - switch to other, 306
  - multiple
    - parameter to specify device type, 154
    - parameter to specify size, 154
  - requirement for
    - parameter to specify, 155
  - sequential
    - description of, 305
- protection log
  - dual
    - switch to other, 230
- Protection log data sets
  - maximum number of
    - parameter to specify, 148
- protection log data sets
  - spanned data records in, 155
- punctuation and symbols in syntax, 4

## Q

- QBLKSIZE
  - ADARUN parameter, 164
- quasi-reentrant code
  - definition, 281

## R

- RALOCKF
  - operator command, 237
- RALOCKFA
  - operator command, 237
- RDUMPST
  - operator command, 238
- read-only status
  - command to switch on or off, 238
  - parameter to set for session, 165
- READONLY
  - ADARUN parameter, 165
  - operator command, 238
- record buffer
  - enable/disable logging of, 123
  - logging
    - command to start, 233
    - command to stop, 235
- recovery, 303
- redo pool
  - command to display usage, 228
  - parameter to set size of, 126
- reentrant code
  - definition, 281
- region
  - address space as a, 6
- repeated syntax elements, 3
- REPLICATION
  - REPLICATION parameter, 166
- replication pool
  - command to display usage, 228
- required syntax elements, 3
- resources
  - statistics
    - command to display, 228
- restart, 303
- REVFILTER
  - ADARUN parameter, 167
- REVIEW
  - ADARUN parameter, 167
  - operator command, 238
- REVLGBMAX parameter, 168-169
- RPLCONNECTCOUNT parameter, 169
- RPLCONNECTINTERVAL parameter, 170
- RPLPARMS parameter, 170
- RPLSORT parameter, 171
- RPWARNINCREMENT parameter, 172
- RPWARNINTERVAL parameter, 172
- RPWARNMESSAGELIMIT parameter, 173
- RPWARNPERCENT parameter, 173

## S

- search buffer
  - enable/disable logging of, 123
  - logging

- command to start, 233
  - command to stop, 236
  - searches
    - complex S1/S2/S4
      - parameter to set time limit for, 181
    - non-descriptor fields
      - parameter to allow/disallow, 147
  - security pool
    - Adabas
      - parameter to set maximum size, 117
    - command to display usage, 228
  - sequential data sets
    - blocksize
      - parameter to specify, 164
  - session
    - Adabas
      - definition of, 303
    - BS2000 sample job control to execute, 255
    - current parameters
      - operator command to display, 227
    - read-only status
      - parameter to specify, 165
    - sample BS2000 job control to execute, 253
    - sample VSE job control to execute, 261-262
    - sample z/OS job control for TCP/IP, 257
    - sample z/OS job control for UES, 257
    - sample z/OS job control to execute, 256
    - SVC for
      - parameter to specify, 176
    - unique identifier, 311
    - user
      - definition of, 303
    - utilities execution only
      - command to specify, 243
      - parameter to specify, 193
  - shared cache
    - unchanged block option, 86
  - single user
    - operating mode
      - parameter to specify, 131
    - requirements and restrictions, 264
  - SMGT
    - ABNORMALTERM operand, 245
    - ACTPIN operand, 245
    - ADARUN parameter, 174
    - ADDPIN operand, 245
    - DEACTPIN operand, 245
    - DELPIN operand, 245
    - DISPLAY operand, 246
      - sample output, 248
    - DUMP operand, 246
    - MSGBUF operand, 245
    - operator command, 244, 246
    - SNAP operand, 247
    - XACTIVATE operand, 247
    - XCRITICAL operand, 247
    - XDEACTIVATE operand, 247
    - XLOAD operand, 248
    - XNOTCRITICAL operand, 247
  - Sort
    - set size of internal processing area, 128
  - Sort data set
    - enable/disable controller caching for data set, 174
  - SORTCACHE
    - ADARUN parameter, 174
  - spanned records
    - controlling logging in PLOG, 155
  - SPT
    - ADARUN parameter, 175
  - STOPF
    - operator command, 238
  - STOPI
    - operator command, 239
  - STOPU
    - operator command, 239
  - STXIT
    - operator command processing, 215
  - SUBMPSZ
    - ADARUN parameter, 176
  - subparameter syntax, 3
  - SVC
    - ADARUN parameter, 176
  - SVC dump facility, 264
  - SYNCC
    - operator command, 240
  - syntax conventions, 2
    - bold, 2
    - braces ({}), 3
    - brackets ([]), 3
    - defaults, 2
    - ellipsis (...), 3
    - indentation, 3
    - italic, 2
    - lowercase, 2
    - minimum keywords, 2
    - mixed case, 2
    - mutually exclusive choices, 3
    - normal font, 2
    - optional elements, 3
    - punctuation and symbols, 4
    - repeated elements, 3
    - required elements, 3
    - subparameters, 3
    - underlining, 2
    - uppercase, 2
    - vertical bars (|), 3
- ## T
- table of ISNs
    - number of ISNs per element
      - parameter to set maximum, 149
    - parameter to set size of, 121
  - table of ISNs pool
    - command to display usage, 228
  - table of sequential commands
    - parameter to set size of, 125
  - table of sequential commands pool
    - command to display usage, 228
  - tape
    - BS2000 handling
      - parameter to specify, 177
  - TAPEREL
    - ADARUN parameter, 177
  - TARGETID
    - ADARUN parameter, 178
  - TASKCTGY
    - ADARUN parameter, 178

- TCP/IP
    - link to Adabas nucleus
      - command to open/close, 240
      - session execution JCL, 257
  - TCPIP
    - ADARUN parameter, 179
    - operator command, 240
  - TCPURL
    - ADARUN parameter, 179
  - Temp data set
    - controller caching for data set
      - parameter to enable/disable, 180
  - TEMPCACHE
    - ADARUN parameter, 180
  - TFLUSH
    - ADARUN parameter, 181
  - threads
    - number allowed per nucleus
      - parameter to set maximum, 151
    - status
      - command to display, 229
  - timeout control
    - interregion communication limit
      - parameter to set, 98
    - limit for complex searches
      - parameter to set, 181
    - non-activity limit
      - command to set, 239
      - for access-only users parameter to set, 188
      - for ET logic users parameter to set, 189
      - for exclusive control users parameter to set, 190
      - parameter to set maximum time, 139
    - Sx command execution
      - parameter to set maximum time, 140
    - transaction limit
      - parameter to set for ET logic users, 191
      - parameter to set maximum time, 141
  - timeout facility, 319
    - action taken when time limit exceeded, 321
    - parameters, 320
  - TLSCMD
    - ADARUN parameter, 181
  - TMDRQ
    - ADARUN parameter, 182
  - TMETDATA
    - ADARUN parameter, 183
  - TMGTT
    - ADARUN parameter, 184
  - TMLOG
    - ADARUN parameter, 185
  - TMMMSGSEV
    - ADARUN parameter, 186
  - TMRESTART
    - ADARUN parameter, 186
  - TMSYNCMGR
    - ADARUN parameter, 187
  - TMTCIDPREF
    - ADARUN parameter, 188
  - TNAA
    - ADARUN parameter, 188
    - operator command, 241
  - TNAE
    - ADARUN parameter, 189
    - operator command, 241
  - TNAX
    - ADARUN parameter, 190
    - operator command, 241
  - TP monitor
    - link routine corresponding to, 278-279
  - transaction
    - time limit for ET logic users
      - parameter to set, 191
  - transaction ID (XID) pool
    - command to display usage, 228
  - TSO
    - ADALNK link routine, 278
    - ADALNK8 link routine, 278
    - ADALNKR link routine, 278
    - ADALNKR8 link routine, 278
    - API work area, 279
  - TT
    - ADARUN parameter, 191
    - operator command, 242
- ## U
- UES
    - session execution JCL, 257
  - UEXn
    - ADARUN parameter, 192
  - underlining, 2
  - unique descriptor pool
    - command to display usage, 228
    - parameter to set size of, 118
  - UNLOCKF
    - operator command, 242
  - UNLOCKU
    - operator command, 242
  - UNLOCKX
    - operator command, 242
  - uppercase, 2
  - user
    - access-only
      - non-activity time limit parameter to set, 188
      - set non-activity time limit using operator command, 241
    - batch
      - how Adabas identifies, 304
    - command to clear, 239
    - definition of, 303
    - ET logic
      - non-activity time limit parameter to set, 189
      - set non-activity time limit using operator command, 241
      - set transaction time limit using operator command, 242
      - synchronize all using operator command, 240
      - transaction time limit parameter to set, 191
    - exclusive control
      - non-activity time limit parameter to set, 190
      - set non-activity time limit using operator command, 241
    - non-exclusive control
      - lock file for, using operator command, 232
      - unlock file for using operator command, 242
    - non-utility
      - lock file for, using operator command, 231
      - unlock file for using operator command, 242
    - number of command IDs per
      - parameter to set maximum, 149
    - number of ISNs in hold queue per
      - parameter to specify, 146

- profile, 265
  - management of, 265
- program
  - relationship to Adabas operation, 6
- session
  - definition of, 303
- User buffer
  - intermediate
    - parameter to set size of, 129
- user exits
  - B
    - enable/disable logging of, 124
    - logging command to start, 234
    - logging command to stop, 236
  - collation descriptor
    - parameter to specify ID and name, 75
  - hyperdescriptor
    - parameter to specify ID and name, 113
  - link applications to Adabas, 277
  - parameter to specify ID and name, 192
- user queue
  - command to display usage, 228
- user queue element
  - display for recent user
    - using operator command, 222
  - display for specified user
    - using operator command, 230
  - display up to 100
    - using operator command, 229
  - display up to 5
    - using operator command, 229
  - display utility
    - using operator command, 230
  - maximum number of
    - parameter to set, 152
- user queue element (UQE)
  - delete for a file
    - using operator command, 223
  - delete inactive
    - using operator command, 223
- user queue file list pool
  - command to display usage, 228
- users
  - count of current
    - operator command to display, 226
- utilities
  - condition and termination codes, 268
  - data sets accessed on disk, 271
  - display status, 243
  - exclusive user of files required
    - conditions for termination, 272
  - executing, 267
  - file and database usage, 268
  - overview in Adabas operation, 6
  - recommended use of EXECOS when running, 274
  - running under
    - z/VM, 274
  - session for utility execution only
    - parameter to specify, 193
- utility control statement
  - parameter values
    - default, 2
- UTIONLY
  - ADARUN parameter, 193

- operator command, 243

## V

- value buffer
  - enable/disable logging of, 124
  - logging
    - command to start, 234
    - command to stop, 236
- vertical bars (|), 3
- virtual machines
  - term used to define storage space, 6
- VISTA
  - ADARUN parameter, 193
- VSAM
  - reserved device type designations for, 103
- VSE
  - Adabas data sets required for, 260
  - entering operator commands, 215

## W

- Work
  - part 1
    - space allocation using parameter, 125
  - part 2
    - space allocation using parameter, 130
  - part 4
    - parameter to ignore data in, 115
- work
  - part 1
    - function of, 304
- Work data set
  - controller caching
    - parameter to enable/disable, 194
- Work pool
  - Adabas
    - parameter to set size of, 131
    - intermediate user buffer area
      - parameter to set size of, 129
    - sort processing area
      - parameter to set size of, 128
- work pool
  - command to display usage, 228
- WORKCACHE
  - ADARUN parameter, 194

## X

- XCF group
  - name, 87
- XID pool
  - command to display usage, 228

## Z

- z/OS
  - Adabas data sets required for, 255
  - control TCP/IP access using TCPIP parameter, 179
  - entering operator commands, 215
  - TCP/IP URL
    - parameter to set, 179
- z/VM
  - Adabas data sets required for, 259



entering operator commands, 215  
running utilities under, 274

