Adabas Auditing on Mainframe

Adabas Auditing Reference

Version 221

October 2021

ADABAS & NATURAL

Table of Contents

Adabas Auditing Reference

This section provides reference information relating to the use of Adabas Auditing.

Messages and Codes	Describes the error messages that can result when using Adabas Auditing.
Audit Server Initialization Parameters	Provides basic information about server initialization parameters related to Adabas Auditing.
Adabas Nucleus Auditing Parameters (ADAANP)	Explores the two main groups of ADAANP parameters: global and file parameters.
Pertinent ADARUN Parameters for Auditing	Describes the ADARUN settings pertinent to the Adabas Audit Server and Adabas nuclei.
Pertinent ADARUN Parameters for Auditing with Subscription Logging	Explores the ADARUN settings pertinent to the Adabas Audit Server with Subscription Logging (SLOG) active.
Pertinent Operator Commands for Auditing	Describes only those operator commands pertinent to the operation of the Adabas Audit Server and to Adabas Auditing activated in an Adabas nucleus.
Utilities Used with Adabas Auditing	Presents lists of the utilities and utility functions which apply to the use of Adabas Auditing.

1 About this Documentation

Document Conventions

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

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at **https://documenta-tion.softwareag.com**.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at **https://empower.softwareag.com/**.

You can find product information on the Software AG Empower Product Support website at **https://empower.softwareag.com**.

To submit feature/enhancement requests, get information about product availability, and download products, go to **Products**.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the **Knowledge Center**.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at **https://empower.softwareag.com/public_directory.aspx** and give us a call.

Software AG Tech Community

You can find documentation and other technical information on the Software AG Tech Community website at **https://techcommunity.softwareag.com**. You can:

- Access product documentation, if you have Tech Community credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.

- Access articles, code samples, demos, and tutorials.

- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.

- Link to external websites that discuss open standards and web technology.

Data Protection

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

2 Messages and Codes

This section describes the error messages that can result when using Adabas Auditing.

For all other error messages refer to the *Adabas for Mainframes* documentation > *Messages and Codes*.

Adabas Nucleus Auditing Start-up and Run-time Messages

The following messages may be issued by an Adabas nucleus running with `ADARUN AUDITING=YES`.

PARM-ERROR 130	**Auditing initialization error - detected during system open**
Explanation	Once AUDITING is turned on, the only option allowed for the AUDITING parameter is YES.
Action	Set `AUDITING = YES`.

ADAAE6	**{yyyy.mm.dd hh.mm.ss} No connection attempts will be made as a {yyyy.mm.dd hh.mm.ss} result of the AUDCONNECT request**
Explanation	This message is printed as a result of the AUDCONNECT operator command (issued to an Adabas nucleus). The message is printed when no connection attempts will be made as a result of the AUDCONNECT operator command.
Action	Analyze the AUDCONNECT operator command you issued for invalid Auditing Server IDs that might have been specified.

ADAAEG	**{yyyy.mm.dd hh.mm.ss} A connection will be attempted {yyyy.mm.dd hh.mm.ss} with server id**
Explanation	This message is printed as a result of the AUDCONNECT operator command (issued to an Adabas nucleus). The message is printed for each Auditing Server ID for which the nucleus attempts a connection.
Action	No action is required for this informational message.

ADAAEW	**Collection of auditing data for {xxxxxx} commands for file {nnnnn} will not be controlled by field names**
Explanation	The value of *xxxxx* will be either access or update.
	If the value is "access", then one of the ADAANP parameters FACCACBX, FACCDS, or FACCINFO is set to FIELDS and there was no field list provided in FACCFIELDS. In this case, FACCACBX, FACCDS, and FACCINFO are changed to NO.
	If the value is "update", then one of the ADAANP parameters FUPDACBX, FUPDAI, FUPDBI or FUPDINFO is set to FIELDS and there was no field list provided in FUPDFIELDS. In this case, FUPDACBX, FUPDAI, FUPDBI, and FUPDINFO are changed to NO.

Action	Provide a field list (FACCFIELDS or FUPDFIELDS) or set the data collection parameters to a value other than FIELDS (YES or NO).

ADAAE4	**Warning during FACCFIELDS validation: Field {xx} is not defined in file {nnnn}**
Explanation	The field name *xx* in the list of fields in ADAANP parameter FACCFIELDS is not a valid field for file *nnnn*.
Action	Ensure that the field *xx* is correct for this file. For the FACCFIELDS ADAANP parameter, each field should be in the FDT or SDT. This is a warning only. The field *xx* for file *nnnn* will be ignored.

ADAAE5	**Warning during FUPDFIELDS validation: Field {xx} is not defined in file {nnnnn} or is a special descriptor**
Explanation	The field name *xx* in the list of fields in ADAANP parameter FUPDFIELDS is not a valid field for file *nnnn*.
Action	Ensure that the field *xx* is correct for this file. For the FUPDFIELDS ADAANP parameter, each field should be in the FDT. Fields defined in the SDT (special descriptors) are invalid. This is a warning only. The field *xx* for file *nnnn* will be ignored.

Adabas Audit Server Messages

The following messages may be issued by the Adabas Audit Server.

Most messages are written to the console using WTO. Some are noted as being written to file DDAUDERR instead. Messages on DDAUDERR are formatted similarly to a multi-line WTO: the message number and DBID appear on only the first line of the message. Each line is prefixed with the Julian date and local time.

ADAA01	**{yyyy.mm.dd hh.mm.ss} No connection attempts will be made as a {yyyy.mm.dd hh.mm.ss} result of the AUDCONNECT request**
Explanation	This message is printed as a result of the AUDCONNECT operator command (issued to an Auditing Server). The message is printed when no connection attempts will be made as a result of the AUDCONNECT operator command.
Action	Analyze the AUDCONNECT operator command you issued for invalid Adabas database IDs that might have been specified.

ADAA0G	**{yyyy.mm.dd hh.mm.ss} A connection will be attempted** **{yyyy.mm.dd hh.mm.ss} with database id**
Explanation	This message is printed as a result of the AUDCONNECT operator command (issued to an Auditing Server). The message is printed for each Adabas database for which the Auditing Server attempts a connection.
Action	No action is required for this informational message.

ADAAJ2	**The value 'x' is invalid in subscription {subs-id}** **DBID/FNR = nnnnn/nnnnn for {phrase1}** **Valid values are {phrase2}**
Explanation	A value read from the subscription *subs-id* defined in the ALA system file is incorrect. The value is listed as 'x' above, the DBID/FNR is identified, and the name of the incorrect value is listed, i.e. *phrase1* might be "*send insert data image*". An expected valid value is listed, *phrase2*, an example being 'Y' or 'N'.
Action	Using the AMN or SYSALA editor for the ALA system file, verify the subscription, dbid/fnr, and the name of the incorrect value (*phrase1*). Correct if necessary (*phrase2* values). Also contact your Software AG support representative with the contents of the message.

ADAAK2	**Filter {xxxxxxxx} source field name {xx}. Invalid value '{/X'xxxxxxxx}' for keyword**
Explanation	The field name *xx* in filter *xxxxxxxx* has an invalid keyword value as reported in the message.
Action	Investigate the cause of the error, correct it, and restart the Audit Server.

ADAAK3	**Misuse of A( ) or X( ) notation**
Explanation	The notation was specified incorrectly.
Action	Investigate the cause of the error, correct it, and restart the Audit Server.

ADAAK4	**Misuse of wildcharacter in value**
Explanation	The wildcharacter notation was specified incorrectly.
Action	Investigate the cause of the error, correct it, and restart the Audit Server.

ADAAK8	**Global Format {xxxxxxxx} has an invalid length**
Explanation	The Global Format *xxxxxxxx* length is invalid. Maximum size is 32767.
Action	Investigate the cause of the error, correct it, and restart the Audit Server.

ADAALA	**For Subscription xxxxxxx. Format xxxxxxx not found in format definitions**
Explanation	The format *xxxxxxxx* defined for Subscription *xxxxxxxx* was not found in the format definitions.
Action	Investigate the cause of the error, correct it, and restart the Audit Server.

ADAALB	**Subscription {subs-id} database {dbid} file {nn}. {before|after|key image} uses global format buffer {gfb}. This global format buffer contains field table information. It may not be used by more than one combination of database and file.**
Explanation	A global format buffer (GFB) definition contains a field table and is used for more than one combination of database ID and file number. This is not allowed.
Action	Correct the subscription so that the GFB field table is used for a single combination of database ID and file number.

ADAALC	**Subscription {subs-id} database {dbid} file {nn}. {before|after|key} image uses global format buffer {gfb}. This global format buffer is also used by subscription {subs-id} database {dbid} file {nn}.**
Explanation	A global format buffer (GFB) definition is used by a subscription for the same file number on different databases. This is not allowed.
Action	Correct the subscription so that the GFB is used properly.

ADAALD	**GFORMAT xxxxxxx is not unique**
Explanation	Duplicate names for global format definitions have been found.
Action	Check the parameter input, correct the error, and restart the Audit Server.

ADAALE	**Subscription {xxxxxxx} Global and Local format buffers are mutually exclusive**
Explanation	A global or a local format may be specified, but not both.
Action	Check the parameter input, correct the error, and restart the Audit Server.

ADAALJ	**Invalid hexadecimal input for X( ) notation**
Explanation	The hexadecimal value was invalid.
Action	Investigate the cause of the error, correct it, and restart the Audit Server.

ADAALK	Filter {xxxxxxxx}, Group {xxxx}, Condition {xxxx} inconsistent with value list
Explanation	A value list cannot be used with the specified condition.
Action	Investigate the cause of the error, correct it, and restart the Audit Server.

ADAALL	Global filter format {xxxxxxxx} not found in Global format definitions
Explanation	The filter format ID *xxxxxxxx* does not match a global format.
Action	Investigate the cause of the error, correct it, and restart the Audit Server.

ADAALM	Subscription {xxxxxxxx} has been deactivated
Explanation	The subscription *xxxxxxxx* was deactivated because of an error.
Action	Ensure the Audit Server is running with an ALOG defined if you wish to use destinations of type AUDIT.

ADAALP	Filter {xxxxxxxx}; target field not allowed
Explanation	For a request or client filter, the target can only be one or more values. The target cannot be a field. The subscription is deactivated.
Action	Correct the filter to use a value as the target of the compare.

ADAA18	Unable to acquire a UQE for subscription Response {rrrrr}, subcode {sssss}
Explanation	The subscription is deactivated.
	This message appears in file DDAUDERR and is not displayed on the console.
Action	Increase the value specified for the ADARUN NU parameter, and restart the Audit Server.

ADAA2D	Format buffer error Subscription {subscription-id}, {error-description} Response code {rrrrr}, subcode {sssss} {field name ff|Additions2 x'aaaa'}
Explanation	An error occurred processing a format buffer of a command for the subscription named in the message. The response code and subcode are shown in the message. When a field name is available it is shown, otherwise the contents of either ACBAdd2 bytes 0:1 or ACBXErrB are shown in hex.
	Possible error descriptions are:
	■ Request format buffer
	■ Client-Info format buffer
	This message appears in file DDAUDERR and is not displayed on the console.
Action	Correct the error in the format buffer indicated by the error description.

ADAA2E   **Open of File destination {name} failed as ALOG is not active**

Explanation   A command was issued to open an Audit destination, but a command log was not defined in the Audit Server.

Action   Ensure the Audit Server is running with an ALOG defined if you wish to use Adabas Auditing.

ADAA2F   **One or more Audit destinations are defined but ALOG is not active**

Explanation   One or more Audit destination definitions are being used, but the ALOG is not defined in the Audit Server start-up JCL. The AUDIT destinations are set to "Unavailable" and processing continues.

Action   Identify the ALOG in the Audit Server start-up JCL and try again.

ADAA2G   **Failed to set conversion attributes to subscription flist encoding for subscription {subscription-id} DBID {dbid} file {fnr}. Failed to set conversion attributes to user encoding**

Explanation   An error occurred processing a command for the subscription named in the message. A description of the error, the response code, and subcode are also listed in the message.

This message appears in file DDAUDERR and is not displayed on the console.

Action   Correct the conversion attribute problem indicated by the error description.

ADAA2H   **Subscription {subscription-id} DBID {dbid} file {fnr}. filter {filter-id} group {group-name} condition {nnnnn} field {xx} error: {error-description}**

Explanation   An error occurred with the field, filter, and subscription named in the message. A description of the error, the database ID, file number, and condition are also listed in the message. Possible error descriptions are:

- Begin byte or length not supported for field format
- Begin byte plus length greater than field length
- Field, PE or MU instance not found in Format buffer
- Field specified cannot be used in field level filter
- Field type unrecognized
- Invalid begin byte specified
- Invalid length specified
- Invalid or out of range FLIST data: *data*

  hex: *hex*
- Source field type incompatible with target

This message appears in file DDAUDERR and is not displayed on the console.

Action	Correct the problem indicated by the error description.

ADAA2Q	**Error during subscription processing: {error-description}.Subscription{ subscription-id}. Response code {rrrrr}, subcode {sssss}**
Explanation	An error occurred processing a command for the subscription named in the message. A description of the error, the response code, and subcode are also listed in the message. Possible error descriptions are:

- no FCB

- no FDT

- record too big

- decompress error

This message appears in file DDAUDERR and is not displayed on the console.

Action	Investigate the cause of the error. The error may be an internal error. If you are unable to determine the cause, contact Software AG technical support.

ADAA1Q	**SLOG turned on for destination {dest-name}**
Explanation	SLOG processing has been turned on for the destination listed in the message (*dest-name*).
Action	No action is required for this informational message.

ADAA1R	**SLOG turned off for destination {dest-name}**
Explanation	SLOG processing has been turned off for the destination listed in the message (*dest-name*).
Action	No action is required for this informational message.

ADAA14	**Destination {destination} deactivated due to {reason}**
Explanation	The status for the identified destination has changed on the Audit Server.

The deactivation *reason* will be one of the following:

- auditing pool full

- request from ADADBS

- request via operator command/AOS

- SLOG-related error

- SLOG system file full

Action	If the deactivation reason is "auditing pool full", consider increasing the value of the LAP parameter. For other reasons this message is for information only.

ADAA15   **Subscription {subscription} deactivated due to {reason}**

Explanation   The deactivation *reason* will be one of the following:

- deactivation of last destination
- error in subscription processing
- request from ADADBS
- request via operator command/AOS

Action   No action is required for this informational message.

ADAA16   **Subscription {subscription} activated due to {reason}**

Explanation   The activation *reason* will be one of the following:

- request from ADADBS
- request via operator command/AOS

Action   No action is required for this informational message.

ADAA19   **SLOG cleanup process {started|aborted|completed|halted}**

Explanation   The SLOG cleanup process has started, aborted, completed or halted.

Action   No action is required for this informational message.

ADADBS Utility Messages

ERROR-242   **Invalid FILE number specified for auditing**

Explanation   The ADADBS AUDITING function is only valid for files participating in auditing (are included in an ADAANP FILE group).

Action   Investigate the cause of the error. Correct it, and rerun the job.

ERROR-243   **The specified FILE was already {active|inactive}**

Explanation   The database detected the file identified in the message was already active or inactive.

Action   Investigate the cause of the error. Correct it, and rerun the job.

ERROR-245	**AUDITSERVER function only allowed for use with the Audit Server**
Explanation	The ADADBS AUDITSERVER function is only valid for an Audit Server.
Action	Investigate the cause of the error. Correct it, and rerun the job.

ERROR-246	**The specified destination {destination} was not found**
Explanation	The destination identified in the message is not defined to the Audit Server.
Action	Investigate the cause of the error. Correct it, and rerun the job.

ERROR-247	**The specified subscription {subscription} was not found**
Explanation	The subscription identified in the message is not defined to the Audit Server.
Action	Investigate the cause of the error. Correct it, and rerun the job.

ERROR-248	**The specified destination {destination} was already {active|inactive}**
Explanation	The Audit Server detected the destination identified in the message was already active or inactive.
Action	Investigate the cause of the error. Correct it, and rerun the job.

ERROR-249	**The specified subscription {subscription} was already {active|inactive}**
Explanation	The Audit Server detected the subscription identified in the message was already active or inactive.
Action	Investigate the cause of the error. Correct it and, rerun the job.

ERROR-250	**Missing parameter: DESTINATION required**
Explanation	ADADBS detected that the required parameter DESTINATION was missing.
Action	Investigate the cause of the error. Correct it, and rerun the job.

ERROR-251	**Missing parameter: DESTINATION or SUBSCRIPTION required**
Explanation	ADADBS detected that the required parameter DESTINATION or SUBSCRIPTION was missing.
Action	Investigate the cause of the error. Correct it, and rerun the job.

ERROR-252   **SUBSCRIPTION parameter not allowed for OPEN or CLOSE: only DESTINATION**

Explanation   ADADBS detected invalid parameter SUBSCRIPTION with OPEN or CLOSE.

Action   Investigate the cause of the error. Correct it, and rerun the job.

ERROR-253   **The specified subscription {subscription} was not {activated|deactivated}**

Explanation   The Audit Server detected an unexpected response code from an attempt to activate
                   or deactivate the subscription identified in the message.

Action   Investigate the cause of the error. Correct it, and rerun the job. If the cause of the error
                   cannot be determined, contact your Software AG technical support representative.

ERROR-254   **Auditing not active for Adabas**

Explanation   The database is not an auditing database (`ADARUN AUDITING=YES`).

Action   Investigate the cause of the error. Correct it and rerun the job.

ERROR-255   **NUCID specified for a non-cluster Adabas DBID**

Explanation   The `NUCID=` parameter was specified for a non-cluster Adabas database.

Action   Investigate the cause of the error. Correct it, and rerun the job.

ERROR-257   **{CLOSE | OPEN} failed for specified destination {destination}**

Explanation   ADADBS detected that the Audit Server could not CLOSE or OPEN the destination
                   identified in the message.

Action   Investigate the cause of the error. The Audit Server job log will provide more details
                   on why the request failed.

ERROR-258   **The specified destination {destination} cannot be activated.Data still exists on the
                   SLOG system file for this inactive destination.**

Explanation   An attempt was made to activate the destination identified in the message while items
                   for the destination still exist on the SLOG system file from the last time it was active.

Action   Investigate the cause of the error. Correct it, and rerun the job.

ERROR-259   **The specified destination {destination} was already {open|closed}**

Explanation   The Audit Server detected the destination identified in the message was already open
                   or closed for all tasks.

Action   Investigate the cause of the error. Correct it, and rerun the job.

ERROR-260	**The specified destination {destination} was not {activated|deactivated}**
Explanation	The Audit Server detected an unexpected response code from an attempt to activate or deactivate the destination identified in the message.
Action	Investigate the cause of the error. Correct it, and rerun the job. If the cause of the error cannot be determined, contact your Software AG technical support representative.

ERROR-265	**Invalid value for parameter AUDCONNECT**
Explanation	Only valid DBIDs, auditing server IDs, or ALL are allowed.
Action	Supply the correct parameter value and rerun the job.

Adabas Response Codes

Response 136	
Origin	Mainframe systems only
Explanation	This response code is used for communication with Adabas utilities and Adabas Audit Administration (SYSALAA) or Adabas Online System (AOS). The system returns this error because the requested function could not be performed on the Adabas system and its files, or because an error occurred in a SYSALAA/AOS or utility function.
Action	For SYSALAA/AOS, a subcode is displayed in the error message or a descriptive message indicates the error that occurred. For utility functions, the subcodes may be described within the message text or a descriptive message indicates the error that occurred. If you do not understand the action to take, note the subcode and the function that caused the response code and call your Software AG technical support representative for assistance.

3 Audit Server Initialization Parameters

To use Adabas Auditing and customize its processing, you must supply various auditing definitions. These definitions are read from the Auditing system file at Adabas Audit Server start-up and are maintained using the Adabas Auditing Configuration.

No initialization parameters are required to be defined to the Adabas Audit server start-up job.

4 Adabas Nucleus Auditing Parameters (ADAANP)

The user needs to specify definitions for the auditing software in an Adabas nucleus running with `ADARUN AUDITING=YES`.

The auditing definitions are read from the sequential file or PDS member allocated to the DD name ADAANP (i.e. ADAbas Auditing Nucleus Parameter).

The auditing definitions are specified via parameters in the input file. The parameter syntax is the same as the parameter syntax for Adabas utilities. Input records are prefixed with 'ADAANP'.

For cluster or parallel services the auditing parameters are local and fixed for each nucleus.

The term 'local' means the parameters may be the same or different in different cluster or parallel services nuclei. The term 'fixed' means the parameters will only be read in during nucleus start-up.

There are two groups of auditing parameters:

- global parameters specific to a nucleus
- file parameters specific to one or more files

The term global here means the definitions are global for one nucleus. The term global here does not mean the definitions apply across cluster nuclei.

Global Auditing Parameters

Global auditing definitions are definitions defined for one nucleus that are not specific to a file.

The parameter GLOBAL denotes the start of a group of global auditing parameters. The GLOBAL parameter is specified as a word standing by itself.

The global auditing parameters are listed in the table below.

> **Note:** These input parameters must be prefixed with 'ADAANP' and are read from the file allocated to the DD name ADAANP.

Parameter	Description	Valid Values	Default Value
GSERVERID	Specifies the Audit Server ID to be used when the server ID is not specified at the file level for a file where auditing data will be collected.	The valid values for this parameter are 1 – 65,535. The server ID may not be set to the database ID associated with the currently active nucleus.	-

Parameter	Description	Valid Values	Default Value
GCONNECTCOUNT	Connect count – This definition specifies the number of connect attempts the nucleus should attempt for an Audit Server when the first attempt to connect to an Audit Server fails.	The valid values for this parameter are 0 – 2,147,483,647.	The default value for this parameter is zero (i.e. no subsequent connection attempts will be made after the first connection attempt fails).
GCONNECTINTERVAL	Connect interval – This definition specifies the interval in seconds between connection attempts when an attempt to connect to an Audit Server fails.	The valid values for this parameter are 0 – 2,147,483,647.	The default value for this parameter is be zero (i.e. no subsequent connection attempts will be made after a connection attempt fails).
GAPWARNINCREMENT	Auditing pool warning increment – This definition specifies the increment, in percent of auditing pool usage, at which auditing pool warning messages should be written. This definition is related to the next definition.	The valid values for this parameter will be 1 – 99.	The default value for this parameter is 10.
GAPWARNPERCENT	Auditing pool warning percent – This definition specifies the threshold, in percent of auditing pool usage, at which auditing pool warning messages should be written.	The valid values for this definition are 1 - 99.	The default value for this parameter is zero (i.e. auditing pool warning messages will not be written).
GAPWARNMESSAGELIMIT	Auditing pool warning message limit – This definition specifies the number of auditing pool usage messages that may be written before these messages are suppressed.	The valid values for this parameter are 1 – 2,147,483,647.	The default value for this parameter is 5.
GAPWARNINTERVAL	Auditing pool warning message interval – This definition specifies the interval in seconds during which auditing pool usage warning messages are suppressed on the console.	The valid values for this parameter are 1 – 2,147,483,647.	The default value for this parameter is 60.
SENDVIOLATION	When the SENDVIOLATION parameter is set to YES, the ACBX and client info will be sent to the audit server when a response code 200 thru 203 is received. When it's set to NO or not specified and it defaults to NO, nothing will be sent when a	YES or NO	NO

Parameter	Description	Valid Values	Default Value
	response code 200 thru 203 is received.		

File Auditing Parameters

The file auditing parameters are specified for one Adabas nucleus and are specific to one file or range of files.

The file auditing parameters are listed in the table below.

> **Note:** These input parameters must be prefixed with 'ADAANP' and are read from the file allocated to the DD name ADAANP.

Parameter	Description	Values	Default
FILE	The `FILE` parameter denotes the start of a group of auditing parameters related to one or more files. The `FILE` parameter is specified as a word standing by itself.	-	-
FLIST	The parameter `FLIST` is used to give the list of one or more files that relate to the current `FILE` definition. The `FLIST` parameter will allow the user to specify one or more files singly or as a range of files.	-	-
FSERVERID	Server ID – This definition specifies the Audit Server ID to be used when data is collected for the file(s). If this parameter is not specified for the file, the global server ID is used. The server ID may not be set to the database ID associated with the currently active nucleus.	1 - 65,535	-
FACCDS	Control collection of the data storage image for an access command.	FIELDS|NO|YES	YES
FACCFIELDS	Control collection of auditing data for an access command based on one or more field names - This definition specifies one or more field names that control whether auditing data is collected for an access command issued for the file(s). The auditing data for an access command will only be collected if one or more of the specified	One or more two-character field names may be specified for this parameter. Each two-character value must be a valid Adabas field name. Field names are separated by a comma. There is no validation that the field name actually exists on the related file(s).	-

Parameter	Description	Values	Default
	fields are referenced by an access command issued for the file(s).		
FDELDS	Control collection of the data storage image for a delete command.	FIELDS|NO|YES	YES
FINSDS	Control collection of the data storage image for an insert command.	FIELDS|NO|YES	YES
FUPDAI	Control collection of the data storage (after) image for an update command.	FIELDS|NO|YES	YES
FUPDBI	Collect the before image of data storage for an update command – This definition specifies whether the before image of data storage will be collected for an update command issued for the file(s).	FIELDS|NO|YES When value 'FIELDS' is set, the before image of data storage will be collected if one or more fields specified in the FUPDFIELDS parameter is referenced by an update command issued for the file(s). When value 'NO' is set, the before image of data storage will not be collected for an update command issued for the file(s). When value 'YES' is set, the before image of data storage will be collected for an update command issued for the file(s).	NO
FUPDFIELDS	Control collection of auditing data for an update command based on one or more field names – This definition specifies one or more fields that control whether auditing data will be collected for an update command issued for the file(s). The auditing data for an update command will only be collected if one or more of the specified fields are referenced by an update command issued for the file(s).	One or more two-character field names may be specified for this parameter. Each two-character value must be a valid Adabas field name. Field names are separated by a comma. There is no validation that the field name actually exists on the related file(s).	-
FADMPARM	Control collection of auditing data for an administrative command or utility.	NO|YES	NO
FADMINFO	Control collection of auditing client data for an administrative command or utility.	NO|YES	NO

Examples

This section gives an example of a set of auditing definitions.

```
ADAANP GLOBAL
ADAANP GAPWARNINCREMENT=5
ADAANP GAPWARNINTERVAL=30
ADAANP GAPWARNMESSAGELIMIT=50
ADAANP GAPWARNPERCENT=75
ADAANP GCONNECTCOUNT=10
ADAANP GCONNECTINTERVAL=60
ADAANP GSERVERID=2001
*
* A FILE example follows with every parameter specified
*
ADAANP FILE
ADAANP FLIST=25
ADAANP FACCFIELDS='AA,AB'
ADAANP FUPDBI=FIELDS
ADAANP FUPDFIELDS='BB,BC'
*
* A FILE example follows with a subset of parameters specified
*
ADAANP FILE
ADAANP FLIST=26
*
ADAANP FILE
ADAANP FLIST=20,22,300-309
ADAANP FSERVERID=2002
ADAANP FUPDFIELDS='AA,BB,CC'
```

5 Pertinent ADARUN Parameters for Auditing

This section describes the ADARUN settings pertinent to the Adabas Audit Server and Adabas nuclei (running with `ADARUN AUDITING=YES`).

Parameter	Use to	Adabas Audit Server	Adabas Nucleus
`ALOGDEV`	Specify the device type used for multiple audit log data sets. **Note:** This parameter must be specified with the ALOGSIZE parameter.	Yes	No
`ALOGSIZE`	Specify the number of blocks available for each audit log in multiple audit log data sets. **Note:** This parameter must be specified with the ALOGDEV parameter.	Yes	No
`AUDITING`	Enable auditing of Adabas nuclei.	No	Yes
`DUALALD`	Specify the device type to be used for dual audit logging.	Yes	No
`DUALALS`	Specify the number of blocks available for each dual audit log.	Yes	No
`LAP`	Specify the size of the Auditing Pool.	Yes	Yes
`LU`	Specify the length (in bytes) of the intermediate user buffer area.	Yes	Yes
`NAB`	Specify the number of attached buffers to be used during the session.	Yes	Yes
`NALOG`	Specify the maximum number of audit log data sets to be opened and logged in the PPT for the session.	Yes	No
`NC`	Specify the maximum number of command queue elements (CQEs).	Yes	Yes
`NT`	Specify the number of threads to be used during the session.	Yes	Yes
`V64BIT`	Specify whether virtual storage above the 2-gigabyte bar should be used.	Yes	Yes
`ZIIP`	Specify whether or not to activate Adabas for zIIP.	Yes	Yes

Quick Reference

ALOGDEV Parameter: Multiple Audit Logs Device Type

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

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

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

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

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

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

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

Example

The following example runs the Adabas Audit Server with audit logging. It uses six audit log data sets and a model 3380 device for audit logging. The size of each audit 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,NALOG=6,ALOGDEV=3380,ALOGSIZE=1350
```

ALOGSIZE Parameter: Multiple Audit Log Size

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

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

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

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

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

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

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

Example

The following example runs the Adabas Audit Server with audit logging. It uses six audit log data sets and a model 3380 device for audit logging. The size of each audit 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,NALOG=6,ALOGDEV=3380,ALOGSIZE=1350
```

AUDITING Parameter: Enable Nucleus Auditing

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

The AUDITING parameter enables auditing on Adabas nuclei.

> **Note:** Refer to the *Adabas Auditing* documentation > *Installation* chapter > *Activating Auditing for Application Files* for information on how to activate auditing for application files.

Example

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

DUALALD Parameter: Dual Audit Log Device Type

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

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

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

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

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

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

Example

The following example runs the Adabas Audit Server with audit logging and uses a model 3380 device for dual audit logging. The size of each audit 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,DUALALD=3380,DUALALS=1350
```

DUALALS Parameter: Dual Audit Log Size

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

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

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

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

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

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

Example

The following example runs the Adabas Audit Server with audit logging and uses a model 3380 device for dual audit logging. The size of each audit 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,DUALALD=3380,DUALALS=1350
```

LAP Parameter: Adabas Auditing Pool Size

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

> **Notes:**

1. The LAP parameter should be specified for an Adabas nucleus running with ADARUN parameter AUDITING=YES. In addition, the Auditing pool will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters AUDITING=YES and V64BIT=YES.

2. The LAP parameter should be specified for an Adabas Audit Server.

3. The LAP parameter values in the Adabas nucleus and the Adabas Audit Server may be different.

4. Software AG recommends setting LAP to a relatively large value (e.g. LAP=40M or greater). To determine what this setting should be for your site, you could run Adabas in a test environment with auditing turned on, simulate the activity in the production environment on this test system, and use the Adabas DRES operator command to view the current and high-water mark of auditing pool usage.

5. If an Adabas nucleus with AUDITING=YES is run with V64BIT=YES and LARGEPAGE=G64, the allocation request for the Auditing pool will be rounded up to the nearest multiple of two gigabytes. Any difference between the specified size and the allocated size will remain unused. For this reason, we recommend that if LARGEPAGE=G64, you specify the LAP value as a multiple of two gigabytes to avoid leaving a potentially large amount of allocated storage unused. If

`LARGEPAGE=G64` is specified and 2GB large pages are not available for allocation, the specified size will be rounded up to the nearest multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.

Example

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

LU Parameter: Length of Intermediate User Buffer Area

Parameter	Specify . . .	Minimum	Maximum	Default
LU	the size of the intermediate user buffer area.	none	none (see note 1)	65535 (see note 2)

The LU parameter is optional. The size specified must be large enough to accommodate all Adabas control information (204 bytes), all user buffers (format, record, search, value, ISN) that may be required for any given Adabas command, plus any user information passed from Adabas link routines to nucleus user exits. If the ADALNK has user exits, the addresses of the ACB and each buffer passed will be stored in this space. Also, if the ADALNK has declared LUSIZE and has user exits, this buffer size plus 64 bytes will also be stored in this space.

If the multifetch/prefetch option or a utility that needs large record/ISN buffers is to be used during the session, the setting of LU must be large enough to contain the buffers needed.

> **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 systems, LU cannot exceed a value greater than that produced by the following calculation: *NABvalue* x 4096.

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.

3. The ADACHK utility can use large record buffer lengths when making nucleus calls to verify spanned Data Storage records or an index structure with many levels. If this is the case, the settings of your LU and **NAB ADARUN** parameters may need to be increased.

The LU parameter syntax is:

```
LU={ n | 65535 }
```

Specific Product Recommendations

For an Adabas Audit Server, the LU parameter must be greater than or equal to 167,000.

In addition, if audit data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work LU parameter must be greater than or equal to the LU parameter setting for the Adabas Audit Server itself (greater than or equal to 167,000).

Example

The following example runs the Adabas Audit Server with an Adabas intermediate user buffer area of 200,000 bytes.

```
ADARUN PROG=ADANUC,LU=200000
```

NAB Parameter: Number of Attached Buffers

Parameter	Specify . . .	Minimum	Maximum	Default
NAB	the number of attached buffers to be used.	1	varies, depending on the amount of available virtual storage	16

The NAB parameter defines the number of attached buffers to be used during the session. An attached buffer is an internal buffer used for interregion communication. It is required in all environments. Adabas allocates an attached buffer pool with a size equal to the value of NAB multiplied by 4096 bytes.

> **Notes:**

1. The allocation for buffers in the attached buffer pool is done in 256-byte slots. This means that each allocation is rounded to a multiple of 256. For example, if a size of 300 bytes is needed, the allocated space is 512 bytes.

2. You may specify as many attached buffers as fit into the available virtual storage.

3. In environments running in 31-bit addressing mode, the attached buffer pool space is allocated above the 16-MB line.

4. The ADACHK utility can use large record buffer lengths when making nucleus calls to verify spanned Data Storage records or an index structure with many levels. If this is the case, the settings of your LU and NAB ADARUN parameters may need to be increased.

The NAB parameter syntax is:

```
NAB={ n | 16 }
```

Specific Product Recommendations

1. For an Adabas Audit Server, set parameter NAB to a value greater than or equal to:

```
41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
```

For example, if one Adabas nucleus will be sending data to the Adabas Audit Server, set the NAB parameter greater than or equal to 410 (e.g. `NAB=420`).

2. If data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work NAB parameter must also be set to a value greater than or equal to:

```
41 * 10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
```

NALOG Parameter: Number of Audit Log Data Sets

Parameter	Specify . . .	Minimum	Maximum	Default
NALOG	the maximum number of audit log data sets to be opened and logged in the PPT.	2	8	0

This parameter specifies the maximum number of audit log data sets to be opened and logged in the parallel participant table (PPT) for the current Audit Server. Valid values are 0 (the default) and 2-8. Other values produce a parameter error. A zero value disables the parameter.

If more than the number of log data sets specified in the NALOG parameter are allocated in the startup JCL, only the number specified in the NALOG parameter are opened and logged; any additional allocated data sets are activated using the dynamic allocation process.

NALOG must be specified with a nonzero value in conjunction with the ALOGDEV, ALOGSIZE, and UEX12 parameters.

NALOG is not valid if DUALALD, DUALALS, or UEX2 are specified.

UEX2 and UEX12 are mutually exclusive.

If a nonzero NALOG, ALOGDEV, and ALOGSIZE are specified, but `LOGGING=NO`, Adabas attempts to open and log the number of data sets specified by NALOG; however, no logging occurs until the LOGGING parameter has been set to YES using either an operator command or Adabas Online System.

Example

The following example runs the Adabas Audit Server with audit logging. It uses six audit log data sets and a model 3380 device for audit logging. The size of each audit 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,NALOG=6,ALOGDEV=3380,ALOGSIZE=1350
```

NC Parameter: Number of Command Queue Elements

Parameter	Specify . . .	Minimum	Maximum	Default
NC	the maximum number of command queue elements.	20	32767	200

The number of command queue elements (CQEs) established for an Adabas nucleus 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 the *Adabas Review Concepts Manual >Concepts and Facilities > Adabas Review Processing and Components > Storage Requirements*.

Specific Product Recommendations

1. For an Adabas Audit Server, set parameter NC to a value greater than or equal to:

   ```
   10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
   ```

   For example, if one Adabas nucleus will be sending data to the Adabas Audit Server, set the NC parameter greater than or equal to 10 (e.g. `NC=20`).

2. If data is sent through Entire Net-Work from one or more Adabas nuclei to an Adabas Audit Server, the Entire Net-Work NC parameter must also be set to a value greater than or equal to:

   ```
   10 * the-number-of-Adabas-nuclei-sending-data-to-the-Adabas-Audit-Server
   ```

Example

Run the Adabas nucleus with a maximum of 500 elements in the command queue.

```
ADARUN PROG=ADANUC,NC=500
```

NT Parameter: Number of Threads

Parameter	Specify . . .	Minimum	Maximum	Default
NT	the number of user threads to be used during the Adabas session.	4	250	5

A thread is used to execute a single command for any user (access or update). Each thread requires 8192 bytes.

In environments running in 31-bit addressing mode, the threads are allocated above the 16-megabyte line.

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

Specific Product Recommendations

For an Adabas Audit Server, the NT parameter must be greater than or equal to 15.

Example

The following example runs the Adabas Audit Server with 20 user threads.

```
ADARUN PROG=ADANUC,NT=20
```

V64BIT Parameter: 64-Bit Virtual Storage Option

Parameter	Specify . . .	Possible Values	Default
V64bit	whether or not the Adabas nucleus should use virtual storage above the two-gigabyte bar (64-bit virtual storage). For the nucleus to use 64-bit virtual storage, the operating system must also support 64-bit virtual storage.	YES | NO	NO

Currently, the following storage areas will be allocated above the two-gigabyte bar in an Adabas nucleus:

- the flush I/O pool (see ADARUN parameter LFIOP)

- the PLOG protection I/O buffers (see ADARUN parameter NPLOGBUFFERS)

- the Work part 1 protection I/O buffers (see ADARUN parameter NWORK1BUFFERS)

- the cast-out directory buffer, in cluster environments (automatically derived from the attributes of the global cache space for Adabas Cluster or Adabas Parallel Services)

The use of 64-bit virtual storage will help alleviate virtual storage constraints below the two gigabyte bar in installations that use large amounts of virtual storage in Adabas nucleus sessions (for example, nucleus sessions with ADARUN parameter settings of `LBP=1200M` and `LFIOP=200M`).

Value	Meaning
YES	The Adabas nucleus will use 64-bit virtual storage (storage above the two-gigabyte bar).
NO	The default setting. The Adabas nucleus will not use 64-bit virtual storage.

> **Note:** Users in z/OS environments should note that virtual storage above the two-gigabyte bar (64-bit virtual storage) is shown in dumps to SVCDUMP or SYSMDUMP, but not in dumps to MPMDUMP, SYSABEND, or SYSUDUMP. Therefore, we recommend that you use the SVCDUMP in the JCL of an Adabas nucleus running with `ADARUN V64BIT=YES` to ensure that 64-bit virtual storage is shown in any SVC dumps that are produced. If, for some reason, you do not want to produce SVC dumps, we recommend that you use SYSMDUMP to ensure 64-bit virtual storage is shown in a dump of the Adabas nucleus address space.

Specific Product Recommendations

The Auditing pool (LAP) will be allocated in 64-bit virtual storage when an Adabas nucleus runs with ADARUN parameters `AUDITING=YES` and `V64BIT=YES`.

Example

The following example indicates that the Adabas nucleus will make use of virtual storage above the two-gigabyte bar and should use 1M large pages.

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

ZIIP Parameter: Activate Usage of Adabas for zIIP

The ZIIP parameter controls whether Adabas activates Adabas for zIIP (AZPAD).

> **Note:** Adabas for zIIP requires its own license file (AZPAD) in addition to the standard Adabas license file. If the AZPAD license file is missing or found invalid, Adabas will start, but will run as if `ZIIP=NO` had been specified.

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

Value	Meaning
YES	Usage of Adabas for zIIP is enabled. Adabas offloads CPU time from the general processors to System z Integrated Information Processors (zIIP). The purpose of this configuration is to reduce the CPU consumption on the general processors.
NO	The default setting. Usage of Adabas for zIIP is disabled. Adabas runs without the option to offload CPU time to zIIPs.

Dynamic Modification

If Adabas was started with ZIIP=YES, the setting of the ZIIP parameter can be changed at runtime – that is, set to 'NO' and subsequently back to 'YES' – using the from the operator console, the ZIIP function, or the Modify Parameters menu of Adabas Online System. If Adabas was started with ZIIP=NO, the setting of the ZIIP parameter cannot be changed to 'YES' later in this session. ZIIP may be set differently for different nuclei in a cluster. It is a local, modifiable parameter.

Specific Product Recommendations

The Adabas Audit Server and Adabas nuclei running with ADARUN AUDITING=YES support the use of Adabas for zIIP subject to the availability of the appropriate license file (AZPAD).

Example

The following example activates the use of Adabas for zIIP.

```
ADARUN PROG=ADANUC,ZIIP=YES
```

6 Pertinent ADARUN Parameters for Auditing with Subscription Logging

This section describes the ADARUN settings pertinent to the Adabas Audit Server with Subscription Logging (SLOG) active.

ASYTVS Parameter: Asynchronous Buffer Flush

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

Specific Product Recommendations

Consider adjusting the value of this parameter when the Auditing SLOG file is defined for use on the Adabas Audit Server, causing audit data to be written and read from the SLOG file.

Example

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

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

FMXIO Parameter: Parallel LFIOP I/O Operations

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

The LFIOP parameter enables asynchronous buffer flush operation and sets the I/O pool size. The FMXIO parameter sets the limit on the number of I/O operations that can be started in parallel by LFIOP flush processing.

Parameter Dependencies

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

- When ASYTVS=YES (perform buffer flushes by volume), FMXIO specifies the number of I/Os to be started in parallel on each volume. The minimum and default number is 1; the maximum number is 16. If the specified number exceeds that, it is automatically reduced to 16.

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

Dynamic Modification

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

Specific Product Recommendations

Consider adjusting the value of this parameter when the Auditing SLOG file is defined for use with the Adabas Audit Server, causing audit data to be written and read from the SLOG file.

Example

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

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

LARGEPAGE Parameter: Large Page Support Option

Parameter	Specify . . .	Possible Values	Default
`LARGEPAGE`	whether or not the Adabas nucleus should use page-fixed one-megabyte (1MB) large pages of real storage above the two-gigabyte (2GB) bar (`LARGEPAGE=L64`) or page-fixed 2GB large pages above the 2GB bar (`LARGEPAGE=G64`).	G64 | L64 | NO | YES	NO

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

Value	Meaning
	multiple of one megabyte, and the storage will not be backed by large pages. A console message will report on this case.
L64 or YES	The Adabas nucleus will attempt to acquire virtual storage above the 2GB bar backed by page-fixed 1MB large pages. The virtual storage will be backed by large pages if the operating system supports them and if sufficient large pages are available in the system. If insufficient large pages are available, the virtual storage above the 2GB bar will be backed by pageable 4K pages. A value of 'YES' is equivalent to specifying a value of 'L64'. You can specify `LARGEPAGE=L64` or `LARGEPAGE=YES` only if the ADARUN V64BIT parameter is set to 'YES'.
NO	Virtual storage above the 2GB bar will be backed by 4KB pages. This is the default value.

The amount of storage in your operating system available for page-fixed 1MB or 2GB large pages is set using the LFAREA parameter in PARMLIB member IEASYS*xx*. For complete information on the LFAREA parameter, refer to the information provided by IBM (search for LFAREA IBM zOS).

Example

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

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

LBP Parameter: Length of Buffer Pool

Parameter	Specifies	Minimum	Maximum	Default
LBP	the maximum size of the Adabas buffer pool.	80000	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.

Specific Product Recommendations

Consider adjusting the value of this parameter when the Auditing SLOG file is defined for use on the Adabas Audit Server, causing audit data to be written and read from the SLOG file.

Example

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

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

LDEUQP Parameter: Length of Unique Descriptor Pool

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

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

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

Specific Product Recommendations

Consider adjusting the value of this parameter when the Auditing SLOG file is defined for use on the Adabas Audit Server, causing audit data to be written and read from the SLOG file.

Example

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

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

LFIOP Parameter: Asynchronous Buffer Pool Control

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

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

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

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

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

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

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

Specific Product Recommendations

Consider adjusting the value of this parameter when the Auditing SLOG file is defined for use on the Adabas Audit Server causing audit data to be written and read from the SLOG file.

Example

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

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

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

Parameter	Specifies	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 the *Adabas Operations Manual > Adabas Restart and Recovery*.

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

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

Specific Product Recommendations

Consider adjusting the value of this parameter when the Auditing SLOG file is defined for use on the Adabas Audit Server, causing audit data to be written and read from the SLOG file.

Example

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

```
ADARUN PROG=ADANUC,LP=2000
```

NH Parameter: Number of Hold Queue Elements

Parameter	Specifies	Minimum	Maximum	Default
NH	the maximum number of hold queue elements.	20	16777215	20

One hold queue element is required for each record (ISN) placed in hold status. A hold queue element is released when the record is released from hold status.

The hold queue is normally used to hold ISNs for the duration of a transaction. Therefore, NH should be large enough to contain at least the average number of ISNs held per transaction multiplied by the maximum number of transactions running in parallel.

One hold queue element takes 52 bytes of storage. In environments running in 31-bit addressing mode, the hold queue is allocated above the 16-megabyte line.

Each shared hold request with a different command ID (CID), as well as a (shared or exclusive) hold request without a CID, is counted against the NH limit.

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

Specific Product Recommendations

Consider adjusting the value of this parameter when the Auditing SLOG file is defined for use on the Adabas Audit Server, causing audit data to be written and read from the SLOG file.

Example

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

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

NISNHQ Parameter: Number of ISNs in Hold Queue per User

Parameter	Specifies	Minimum	Maximum	Default
NISNHQ	the maximum number of records that can be placed in hold status at the same time by one user.	1	16777215 (NH parameter maximum)	(NH / NU) * 2 or 20, whichever is larger

NISNHQ determines the maximum number of held ISNs the hold queue (see the NH parameter description) can contain from one user. The maximum value allowed for this parameter is the value set for the NH ADARUN parameter (which has a maximum value of 16777215). The default is the greater of either 20 or the number of ISNs the hold queue can contain divided by the number of users, multiplied by two.

A user who attempts to place more records in hold status than permitted receives response code 47 (ADARSP047), subcode 1.

Each shared hold request with a different command ID (CID), as well as a (shared or exclusive) hold request without a CID, is counted against the NISNHQ limit.

Specific Product Recommendations

Consider adjusting the value of this parameter when the Auditing SLOG file is defined for use on the Adabas Audit Server, causing audit data to be written and read from the SLOG file.

Overriding the Parameter Setting

You can override this parameter setting for an individual user by specifying a different value in the Adabas control block for an OP command. See the discussion of the OP command in the *Adabas for Mainframes > Command Reference*.

Example

In the following example, the maximum number of records that may be in hold status for a single user is set at 100.

```
ADARUN PROG=ADANUC,NISNHQ=100,NH=450...
```

In the following example, NISNHQ is not specified. The maximum number of records that may be in hold status for a single user is 80 (the NISNHQ default of (NH / NU) * 2, which is greater than the alternative default of 20).

```
ADARUN PROG=ADANUC,NU=50,NH=2000...
ADARUN PROG=ADANUC,NT=20
```

7 Pertinent Operator Commands for Auditing

The Adabas Audit Server processes the same operator commands that may be given to an Adabas nucleus. For example, ADAEND will terminate the Adabas Audit Server.

For more information about Adabas operator commands and how to enter them, refer to the *Adabas for Mainframes* documentation > *Operations*.

This section describes only those operator commands pertinent to the operation of the Adabas Audit Server and to Adabas Auditing activated in an Adabas nucleus.

The following table categorizes the commands as they apply to the Adabas nucleus and to the Adabas Audit Server and the following subsections describe each command in detail.

Command Category	Applicable Commands
Adabas nucleus	AUDCONNECT
	DAUDPARM
	DAUDSTAT
Adabas Audit Server	AUDCONNECT
	DAUDPARM
	DAUDSTAT
	FEOFAL

AUDCONNECT Command

Use AUDCONNECT to dynamically force a connection attempt to either a specific Auditing Server or Adabas database ID or to all related Auditing Server or Adabas database IDs.

One of the parameters of the AUDCONNECT command must be specified. There is no default. The parameters are described in the following table:

Parameter	Forces a reconnection attempt with
ALL	All known Auditing Server or Adabas database IDs
dbid	The specified Auditing Server or Adabas database ID

AUDCONNECT Command against an Adabas Nucleus

The `AUDCONNECT` operator command may be issued against an Adabas Nucleus as follows:

Syntax:

`F <Nucleus>,AUDCONNECT=[dbid|ALL]`	Force a connection attempt with the specified Auditing server or all related Auditing servers.

AUDCONNECT Command against an Adabas Audit Server

The `AUDCONNECT` operator command may be issued against an Adabas Audit Server as follows:

Syntax:

`F ↵` `<AuditServer>,AUDCONNECT=[dbid|ALL]`	Force a connection attempt with the specified Adabas database ID or all related Adabas database IDs.

DAUDPARM Command

Use `DAUDPARM` to display the audit parameters for an Adabas nucleus (with auditing turned on) or for an Adabas Audit Server.

- DAUDPARM Command against an Adabas Nucleus
- DAUDPARM Command against an Adabas Audit Server

DAUDPARM Command against an Adabas Nucleus

The `DAUDPARM` operator command may be issued against an Adabas nucleus (with auditing turned on) as follows:

Syntax:

`F <AuditServer>,DAUDPARM,GLOBALS`	Displays global parameters.
`F <AuditServer>,DAUDPARM,F=filelist`	Displays the specified file parameters.

A *filelist* is a comma separated list of file numbers and may also contain a range in the format n-m. A filelist with a range should be used with caution as it will list all files in the range whether they exist or not.

Example: Display Global Parameters

```
F AUDITDB,DAUDPARM,GLOBALS
```

Produces the following output:

```
ADAAEV 01701 2021-01-12 19:52:50 Auditing definitions:
ADAAEV 01701 2021-01-12 19:52:50 Global definitions:
ADAAEV 01701 2021-01-12 19:52:50  Connect count .... :              3
ADAAEV 01701 2021-01-12 19:52:50  Connect interval   :             60
ADAAEV 01701 2021-01-12 19:52:50  Message interval   :             60
ADAAEV 01701 2021-01-12 19:52:50  Message limit .... :             20
ADAAEV 01701 2021-01-12 19:52:50  Warn increment ... :             12
ADAAEV 01701 2021-01-12 19:52:50  Warn percent ..... :             70
ADAN41 01701 2021-01-12 19:52:50 Function completed             ↵
```

Example: Display File Parameters

```
F AUDITDB,DAUDPARM,F=100,101
```

Produces the following output:

```
ADAAEV 01701 2021-01-12 20:01:49 Auditing definitions:
ADAAEV 01701 2021-01-12 20:01:49
ADAAEV 01701 2021-01-12 20:01:49 File 100
ADAAEV 01701 2021-01-12 20:01:49  File is active
ADAAEV 01701 2021-01-12 20:01:49  Server ID ........ :           1702
ADAAEV 01701 2021-01-12 20:01:49  Access commands:
ADAAEV 01701 2021-01-12 20:01:49   ACBX ........... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Client info ..... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Format buffer ... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Search buffer ... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Value buffer .... : Yes
ADAAEV 01701 2021-01-12 20:01:49  Delete commands:
ADAAEV 01701 2021-01-12 20:01:49   ACBX ........... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Client info ..... : Yes
ADAAEV 01701 2021-01-12 20:01:49  Insert commands:
ADAAEV 01701 2021-01-12 20:01:49   ACBX ........... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Client info ..... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Format buffer ... : Yes
ADAAEV 01701 2021-01-12 20:01:49  Update commands:
ADAAEV 01701 2021-01-12 20:01:49   ACBX ........... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Client info ..... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Format buffer ... : Yes
ADAAEV 01701 2021-01-12 20:01:49
ADAAEV 01701 2021-01-12 20:01:49 File 101
ADAAEV 01701 2021-01-12 20:01:49  File is active
ADAAEV 01701 2021-01-12 20:01:49  Server ID ........ :           1702
ADAAEV 01701 2021-01-12 20:01:49  Access commands:
ADAAEV 01701 2021-01-12 20:01:49   ACBX ........... : Yes
```

```
ADAAEV 01701 2021-01-12 20:01:49   Client info ..... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Format buffer ... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Search buffer ... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Value buffer .... : Yes
ADAAEV 01701 2021-01-12 20:01:49  Delete commands:
ADAAEV 01701 2021-01-12 20:01:49   ACBX ............ : Yes
ADAAEV 01701 2021-01-12 20:01:49   Client info ..... : Yes
ADAAEV 01701 2021-01-12 20:01:49  Insert commands:
ADAAEV 01701 2021-01-12 20:01:49   ACBX ............ : Yes
ADAAEV 01701 2021-01-12 20:01:49   Client info ..... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Format buffer ... : Yes
ADAAEV 01701 2021-01-12 20:01:49  Update commands:
ADAAEV 01701 2021-01-12 20:01:49   ACBX ............ : Yes
ADAAEV 01701 2021-01-12 20:01:49   Client info ..... : Yes
ADAAEV 01701 2021-01-12 20:01:49   Format buffer ... : Yes
ADAN41 01701 2021-01-12 20:01:49 Function completed      ↵
```

DAUDPARM Command against an Adabas Audit Server

The `DAUDPARM` operator command may be issued against an Adabas Audit Server as follows:

Syntax:

`F <AuditServer>,DAUDPARM`	Displays all parameters.
`F <AuditServer>,DAUDPARM,GLOBALS`	Displays global parameters.
`F <AuditServer>,DAUDPARM,D=destination`	Displays the specified destination.
`F <AuditServer>,DAUDPARM,S=subscription`	Displays the specified subscription.

Example: Display Global Parameters

```
F AUDITSRV,DAUDPARM,GLOBALS
```

Produces the following output:

```
ADAI29 Oper cmd: DAUDPARM,GLOBALS
ADAA00 01702 2021-01-07 18:16:33 Auditing definitions:
ADAA00 01702 2021-01-07 18:16:33 Global definitions:
ADAA00 01702 2021-01-07 18:16:33  Auditing pool warning:
ADAA00 01702 2021-01-07 18:16:33   Increment ....... :          10
ADAA00 01702 2021-01-07 18:16:33   Message interval  :          60
ADAA00 01702 2021-01-07 18:16:33   Message limit ... :           5
ADAA00 01702 2021-01-07 18:16:33   Percent ......... :          50
ADAA00 01702 2021-01-07 18:16:33  Broker stub name   : BROKER
ADAA00 01702 2021-01-07 18:16:33  Connect count .... :          20
ADAA00 01702 2021-01-07 18:16:33  Connect interval   :          60
ADAA00 01702 2021-01-07 18:16:33  Maximum output size:     100,000
```

```
ADAA00 01702 2021-01-07 18:16:33  Open at start .... : Yes
ADAA00 01702 2021-01-07 18:16:33  Open retry count   :              10
ADAA00 01702 2021-01-07 18:16:33  Open retry interval:               0
ADAA00 01702 2021-01-07 18:16:33  Queue full delay   :              60
ADAA00 01702 2021-01-07 18:16:33  Subtasks ......... :               3
ADAA00 01702 2021-01-07 18:16:33  Subtask wait time  :              10
ADAA00 01702 2021-01-07 18:16:33  Log input ........ : No
ADAN41 01702 2021-01-07 18:16:33 Function completed              ↵
```

Example: Display Destination Parameters

```
F AUDITSRV,DAUDPARM,D=PAYROLLD
```

Produces the following output:

```
ADAI29 Oper cmd: DAUDPARM,D=PAYROLLD
ADAA00 01702 2021-01-07 19:47:52 Auditing definitions:
ADAA00 01702 2021-01-07 19:47:52
ADAA00 01702 2021-01-07 19:47:52 Audit Destination PAYROLLD
ADAA00 01702 2021-01-07 19:47:52  Active at start up : Yes
ADAA00 01702 2021-01-07 19:47:52  Open at start .... : Global
ADAA00 01702 2021-01-07 19:47:52  Allow logging .... : Yes
ADAN41 01702 2021-01-07 19:47:52 Function completed  ↵
```

Example: Display All Parameters

```
F AUDITSRV,DAUDPARM
```

Produces the following output:

```
ADAI29 Oper cmd: DAUDPARM
ADAA00 01702 2021-01-07 18:12:54 Auditing definitions:
ADAA00 01702 2021-01-07 18:12:54 Global definitions:
ADAA00 01702 2021-01-07 18:12:54  Auditing pool warning:
ADAA00 01702 2021-01-07 18:12:54   Increment ....... :              10
ADAA00 01702 2021-01-07 18:12:54   Message interval  :              60
ADAA00 01702 2021-01-07 18:12:54   Message limit ... :               5
ADAA00 01702 2021-01-07 18:12:54   Percent ......... :              50
ADAA00 01702 2021-01-07 18:12:54  Broker stub name   : BROKER
ADAA00 01702 2021-01-07 18:12:54  Connect count .... :              20
ADAA00 01702 2021-01-07 18:12:54  Connect interval   :              60
ADAA00 01702 2021-01-07 18:12:54  Maximum output size:         100,000
ADAA00 01702 2021-01-07 18:12:54  Open at start .... : Yes
ADAA00 01702 2021-01-07 18:12:54  Open retry count   :              10
ADAA00 01702 2021-01-07 18:12:54  Open retry interval:               0
ADAA00 01702 2021-01-07 18:12:54  Queue full delay   :              60
ADAA00 01702 2021-01-07 18:12:54  Subtasks ......... :               3
ADAA00 01702 2021-01-07 18:12:54  Subtask wait time  :              10
ADAA00 01702 2021-01-07 18:12:54  Log input ........ : No
ADAA00 01702 2021-01-07 18:12:54
```

```
ADAA00 01702 2021-01-07 18:12:54 Audit Destination PAYROLLD
ADAA00 01702 2021-01-07 18:12:54  Active at start up : Yes
ADAA00 01702 2021-01-07 18:12:54  Open at start .... : Global
ADAA00 01702 2021-01-07 18:12:54  Allow logging .... : Yes
ADAA00 01702 2021-01-07 18:12:54
ADAA00 01702 2021-01-07 18:12:54 Subscription PAYROLLS
ADAA00 01702 2021-01-07 18:12:54  Active at start up : Yes
ADAA00 01702 2021-01-07 18:12:54  Destination(s) ... : PAYROLLD
ADAA00 01702 2021-01-07 18:12:54  DBID / File ...... :   1701 /    100
ADAA00 01702 2021-01-07 18:12:54   Read/Find items . : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54     Search buffer   : No
ADAA00 01702 2021-01-07 18:12:54     Value buffer    : No
ADAA00 01702 2021-01-07 18:12:54   Insert items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54   Update items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54     Data storage BI : Yes
ADAA00 01702 2021-01-07 18:12:54   Delete items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54   Data AI global format     : FILE100
ADAA00 01702 2021-01-07 18:12:54   Data BI global format     : FILE100
ADAA00 01702 2021-01-07 18:12:54  DBID / File ...... :   1701 /    101
ADAA00 01702 2021-01-07 18:12:54   Read/Find items . : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54     Search buffer   : No
ADAA00 01702 2021-01-07 18:12:54     Value buffer    : No
ADAA00 01702 2021-01-07 18:12:54   Insert items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54   Update items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54     Data storage BI : Yes
ADAA00 01702 2021-01-07 18:12:54   Delete items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54   Data AI global format     : FILE101
ADAA00 01702 2021-01-07 18:12:54   Data BI global format     : FILE101
ADAA00 01702 2021-01-07 18:12:54  DBID / File ...... :   1701 /    161
ADAA00 01702 2021-01-07 18:12:54   Read/Find items . : Yes
```

```
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54     Search buffer   : No
ADAA00 01702 2021-01-07 18:12:54     Value buffer    : No
ADAA00 01702 2021-01-07 18:12:54   Insert items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54   Update items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54     Data storage BI : Yes
ADAA00 01702 2021-01-07 18:12:54   Delete items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54   Data AI global format      : FILE161
ADAA00 01702 2021-01-07 18:12:54   Data BI global format      : FILE161
ADAA00 01702 2021-01-07 18:12:54 DBID / File ...... :   1701 /   219
ADAA00 01702 2021-01-07 18:12:54   Read/Find items . : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54     Search buffer   : No
ADAA00 01702 2021-01-07 18:12:54     Value buffer    : No
ADAA00 01702 2021-01-07 18:12:54   Insert items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54   Update items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54     Format buffer   : Yes
ADAA00 01702 2021-01-07 18:12:54     Data storage BI : Yes
ADAA00 01702 2021-01-07 18:12:54   Delete items .... : Yes
ADAA00 01702 2021-01-07 18:12:54     Request info    : Yes
ADAA00 01702 2021-01-07 18:12:54     Client info .   : Yes
ADAA00 01702 2021-01-07 18:12:54   Data AI global format      : FILE219
ADAA00 01702 2021-01-07 18:12:54   Data BI global format      : FILE219
ADAA00 01702 2021-01-07 18:12:54   Client info global format  :
ADAA00 01702 2021-01-07 18:12:54   ACBX global format         :
ADAN41 01702 2021-01-07 18:12:54 Function completed                    ↵
```

DAUDSTAT Command

Use `DAUDSTAT` to display the audit statistics for an Adabas nucleus (with auditing turned on) or for an Adabas Audit Server.

- DAUDSTAT Command against an Adabas Nucleus
- DAUDSTAT Command against an Adabas Audit Server

DAUDSTAT Command against an Adabas Nucleus

The `DAUDSTAT` operator command may be issued against an Adabas nucleus (with auditing turned on) as follows:

Syntax:

`F <AuditServer>,DAUDSTAT`	Displays all statistics.
`F <AuditServer>,DAUDSTAT,GLOBALS`	Displays global statistics.
`F <AuditServer>,DAUDSTAT,F=ALL`	Displays all file statistics.
`F <AuditServer>,DAUDSTAT,F=filelist`	Displays statistics for the specified files.

A *filelist* is a comma separated list of file numbers and may also contain a range in the format n-m.

Example: Display All Statistics

```
F AUDITDB,DAUDSTAT
```

Produces the following output:

```
ADAAE0 01701 2021-01-12 20:48:23 Auditing statistics:
ADAAE0 01701  Total items ... :                    2,231
ADAAE0 01701  Pending items   :                        0
ADAAE3 01701 2021-01-12 20:48:23 File related Auditing statistics:
ADAAE3 01701   File                   Total        Pending      Incomplete
ADAAE3 01701   -------------------------------------------------------------
ADAAE3 01701      9                      0
ADAAE3 01701     10                      0
ADAAE3 01701    100                  1,122
ADAAE3 01701    101                      0
ADAAE3 01701    161                  1,107
ADAAE3 01701    219                      0
ADAN41 01701 2021-01-12 20:48:23 Function completed
```

Example: Display Global Statistics

```
F AUDITDB,DAUDSTAT,GLOBALS
```

Produces the following output:

```
ADAAE0 01701 2021-01-12 20:49:22 Auditing statistics:
ADAAE0 01701  Total items ... :                    2,231
ADAAE0 01701  Pending items   :                        0
ADAN41 01701 2021-01-12 20:49:22 Function completed      ↵
```

Example: Display All File Statistics

```
F AUDITDB,DAUDSTAT,F=ALL
```

Produces the following output:

```
ADAAE3 01701 2021-01-12 20:50:16 File related Auditing statistics:
ADAAE3 01701   File                    Total       Pending     Incomplete
ADAAE3 01701   -------------------------------------------------------------
ADAAE3 01701     9                          0
ADAAE3 01701    10                          0
ADAAE3 01701   100                      1,122
ADAAE3 01701   101                          0
ADAAE3 01701   161                      1,107
ADAAE3 01701   219                          0
ADAN41 01701 2021-01-12 20:50:16 Function completed
```

Example: Display Selected File Statistics

```
F AUDITDB,DAUDSTAT,F=100,101
```

Produces the following output:

```
ADAAE3 01701 2021-01-12 20:51:09 File related Auditing statistics:
ADAAE3 01701   File                    Total       Pending     Incomplete
ADAAE3 01701   -------------------------------------------------------------
ADAAE3 01701   100                      1,122
ADAAE3 01701   101                          0
ADAN41 01701 2021-01-12 20:51:09 Function completed
```

DAUDSTAT Command against an Adabas Audit Server

The `DAUDPARM` operator command may be issued against an Adabas Audit Server as follows:

Syntax:

`F <AuditServer>,DAUDSTAT`	Displays all statistics.
`F <AuditServer>,DAUDSTAT,GLOBALS`	Displays global statistics.
`F <AuditServer>,DAUDSTAT,D=destination`	Displays statistics for the specified destination.
`F <AuditServer>,DAUDSTAT,S=subscription`	Displays statistics for the specified subscription.
`F <AuditServer>,DAUDSTAT,TASKS`	Displays task statistics.

Example: Display Global Statistics

```
F AUDITSRV,DAUDSTAT,GLOBALS
```

Produces the following output:

```
ADAAOK 01702 2021-01-12 21:00:32 Global statistics:
ADAAOK 01702 2021-01-12 21:00:32 Input items ... :          2,231
ADAAOK 01702 2021-01-12 21:00:32  Pending ...... :              0
ADAAOK 01702 2021-01-12 21:00:32 Output items .. :          2,230
ADAAOK 01702 2021-01-12 21:00:32  Pending ...... :              0
ADAAOK 01702 2021-01-12 21:00:32 Items w/errors  :              0
ADAN41 01702 2021-01-12 21:00:32 Function complete
```

Example: Display Destination Statistics

```
F AUDITSRV,DAUDSTAT,D=PAYROLLD
```

Produces the following output:

```
ADAAOK 01702 2021-01-12 21:18:27 Destination PAYROLLD
ADAAOK 01702 2021-01-12 21:18:27  Output items   :          2,230
ADAAOK 01702 2021-01-12 21:18:27  Pending items  :              0
ADAAOK 01702 2021-01-12 21:18:27  Item - source  : 2021-01-12 20:42:07
ADAN41 01702 2021-01-12 21:18:27 Function completed                    ↵
```

Example: Display Task Statistics

```
F AUDITSRV,DAUDSTAT,TASKS
```

Produces the following output:

```
ADAAOK 01702 2021-01-12 21:01:19 Output subtask items:
ADAAOK 01702 2021-01-12 21:01:19  Main ......... :          2,230
ADAAOK 01702 2021-01-12 21:01:19    1 ......... :              0
ADAAOK 01702 2021-01-12 21:01:19    2 ......... :              0
ADAAOK 01702 2021-01-12 21:01:19    3 ......... :              0
ADAAOK 01702 2021-01-12 21:01:19
ADAAOK 01702 2021-01-12 21:01:19 Output subtask message totals:
ADAAOK 01702 2021-01-12 21:01:19  Main ......... :              0
ADAAOK 01702 2021-01-12 21:01:19    1 ......... :              0
ADAAOK 01702 2021-01-12 21:01:19    2 ......... :              0
ADAAOK 01702 2021-01-12 21:01:19    3 ......... :              0
ADAN41 01702 2021-01-12 21:01:19 Function completed          ↵
```

Example: Display Subscription Statistics

```
F AUDITSRV,DAUDSTAT,S=PAYROLLS
```

Produces the following output:

```
ADAAOK 01702 2021-01-12 21:22:40 Subscription PAYROLLS
ADAAOK 01702 2021-01-12 21:22:40  Input items    :          2,229
ADAAOK 01702 2021-01-12 21:22:40  Output items   :          2,229
ADAAOK 01702 2021-01-12 21:22:40  Items w/errors :              0
ADAAOK 01702 2021-01-12 21:22:40  Item - source  : 2021-01-12 20:42:12
ADAAOK 01702 2021-01-12 21:22:40  DBID  1701        100
ADAAOK 01702 2021-01-12 21:22:40   ACBX          :          1,122
ADAAOK 01702 2021-01-12 21:22:40   Client info   :              0
ADAAOK 01702 2021-01-12 21:22:40   Format buffer :          1,121
ADAAOK 01702 2021-01-12 21:22:40   Search buffer :              0
ADAAOK 01702 2021-01-12 21:22:40   Value buffer  :              0
ADAAOK 01702 2021-01-12 21:22:40   Inserts       :              0
ADAAOK 01702 2021-01-12 21:22:40   Deletes       :              0
ADAAOK 01702 2021-01-12 21:22:40   Updates       :              7
ADAAOK 01702 2021-01-12 21:22:40   Reads/Finds   :          1,114
ADAAOK 01702 2021-01-12 21:22:40   Filtered out  :              0
ADAAOK 01702 2021-01-12 21:22:40  DBID  1701        101
ADAAOK 01702 2021-01-12 21:22:40   ACBX          :              0
ADAAOK 01702 2021-01-12 21:22:40   Client info   :              0
ADAAOK 01702 2021-01-12 21:22:40   Format buffer :              0
ADAAOK 01702 2021-01-12 21:22:40   Search buffer :              0
ADAAOK 01702 2021-01-12 21:22:40   Value buffer  :              0
ADAAOK 01702 2021-01-12 21:22:40   Inserts       :              0
ADAAOK 01702 2021-01-12 21:22:40   Deletes       :              0
ADAAOK 01702 2021-01-12 21:22:40   Updates       :              0
```

```
ADAAOK 01702 2021-01-12 21:22:40    Reads/Finds   :            0
ADAAOK 01702 2021-01-12 21:22:40    Filtered out  :            0
ADAAOK 01702 2021-01-12 21:22:40  DBID  1701        161
ADAAOK 01702 2021-01-12 21:22:40    ACBX          :        1,107
ADAAOK 01702 2021-01-12 21:22:40    Client info   :            0
ADAAOK 01702 2021-01-12 21:22:40    Format buffer :        1,107
ADAAOK 01702 2021-01-12 21:22:40    Search buffer :            0
ADAAOK 01702 2021-01-12 21:22:40    Value buffer  :            0
ADAAOK 01702 2021-01-12 21:22:40    Inserts       :            0
ADAAOK 01702 2021-01-12 21:22:40    Deletes       :            0
ADAAOK 01702 2021-01-12 21:22:40    Updates       :            0
ADAAOK 01702 2021-01-12 21:22:40    Reads/Finds   :        1,107
ADAAOK 01702 2021-01-12 21:22:40    Filtered out  :            0
ADAAOK 01702 2021-01-12 21:22:40  DBID  1701        219
ADAAOK 01702 2021-01-12 21:22:40    ACBX          :            0
ADAAOK 01702 2021-01-12 21:22:40    Client info   :            0
ADAAOK 01702 2021-01-12 21:22:40    Format buffer :            0
ADAAOK 01702 2021-01-12 21:22:40    Search buffer :            0
ADAAOK 01702 2021-01-12 21:22:40    Value buffer  :            0
ADAAOK 01702 2021-01-12 21:22:40    Inserts       :            0
ADAAOK 01702 2021-01-12 21:22:40    Deletes       :            0
ADAAOK 01702 2021-01-12 21:22:40    Updates       :            0
ADAAOK 01702 2021-01-12 21:22:40    Reads/Finds   :            0
ADAAOK 01702 2021-01-12 21:22:40    Filtered out  :            0
ADAN41 01702 2021-01-12 21:22:40 Function completed          ↵
```

FEOFAL Command

Use FEOFAL to close the current dual or multiple audit log and switch to another dual or multiple audit log. This command is valid only if dual or multiple audit logging is in effect and only for Adabas Audit Servers.

In nucleus cluster environments, the GLOBAL option can be used to switch the dual or multiple audit log in all cluster nuclei at the same time.

FEOFAL Command against an Adabas Audit Server

The FEOFAL operator command may be issued against an Adabas Audit Server as follows:

Syntax:

`F <AuditServer>,FEOFAL[,GLOBAL]`	Close the current audit log and switch to another audit log.

8 Utilities Used with Adabas Auditing

Various Adabas utilities and specific utility functions are used in auditing. Some of these utilities and functions apply only to the source Adabas nucleus, some of them apply only to the Adabas Audit Server, and some of them apply to both the Adabas nucleus and the Adabas Audit Server.

The following table lists the utilities and utility functions which apply to the use of Adabas Auditing and indicate whether they are relevant to the Adabas Audit Server and/or Adabas database.

Utility and Function	Adabas Audit Server	Adabas Database
ADACHK ALOGPRINT	Yes	No
ADACMP COMPRESS AUDITNM=	No	Yes
ADADBS ADDALOG	Yes	No
ADADBS AUDITSERVER	Yes	No
ADADBS DELALOG	Yes	No
ADADBS MODFCB AUDITNM=	No	Yes
ADADBS OPERCOM AUDCONNECT	Yes	Yes
ADADBS OPERCOM DAUDPARM and DAUDSTAT	Yes	Yes
ADADBS OPERCOM FEOFAL	Yes	No
ADADEF DEFINE AUDITSERVER	Yes	No
ADADEF MODIFY AUDITSERVER	Yes	No
ADAFRM ALOGFRM	Yes	No
ADALOD LOAD	Yes	Yes
ADAORD STORE	Yes	No
ADAREP REPORT	Yes	Yes
ADARES ALCOPY	Yes	No

The following sections describe these utilities and functions in detail.

ADACHK ALOGPRINT

The ADACHK ALOGPRINT function can be used to print the contents of any block or range of blocks in the audit log (ALOG) data set while concurrent updates are running.

```
ADACHK ALOGPRINT RABN={rabn|rabn1-rabn2}[,{rabn|rabn1-rabn2}]...
                        [ABEND]
                        [DEVICE=device]
                        [ERRLIM={error-threshold-count|100}]
                        [LAYOUT={SHORT|MEDIUM|LONG}]
                        [NOUSERABEND]
                        [NUMBER={number|1}]
                        [TEST]
```

The following sections cover the topics:

- Essential Parameters
- Optional Parameters and Subparameters
- Example

Essential Parameters

RABN: RABNs to Be Processed

The RABNs or ranges of RABNs to be printed or dumped.

Optional Parameters and Subparameters

ABEND34: Change User Abend 35 to 34

This optional parameter can be used to change a user abend 35 to user abend 34 if a utility error occurs. This ensures that a dump is produced when the utility terminates abnormally.

The NOUSERABEND, TEST, and ABEND34 parameters affect the processing of the entire ADACHK run.

DEVICE: Device Type

The device type that contains the data set to be printed. This parameter is required if the device type is different from the standard device type assigned by the ADARUN DEVICE parameter.

ERRLIM: Error Threshold

The maximum number of errors that this ADACHK utility function will tolerate before terminating. Valid values are any positive integer equal or less than 5000. If no valid value is specified for this parameter, a default of "100" is used.

If a value less than 0 or greater than 5000 is specified, the following error will result, and the default value will be used:

```
CHK413E, ERROR: Parameter ERRLIM is incorrect.
```

NUMBER: Command Log Data Set Number

The number of the multiple (two through eight) audit log (ALOG) data set from which the blocks are to be printed. If NUMBER is not specified, the blocks are taken from DD/ALOGR1 (the default).

LAYOUT: Report Detail Level

The level of data produced for the report or dump. Valid values are described in the following table. For more information about the output sections described below, read the *Adabas for Mainframes* documentation > *Utilities* > *Functional Overview* > *ADACHK Print Function Output Format*.

Valid Values	Description
SHORT	Specify this value to produce the minimum output, printing sections 1 and 2 of the possible output. This is the default.
MEDIUM	Specify this value to produce medium-level output, printing sections 1 and 3 of the possible output.
LONG	Specify this value to produce extensively-detailed output, printing all three sections of the possible output.

> **Note:** The results from the `LAYOUT=LONG` setting are what you get when you run the equivalent ADAICK utility function.

NOUSERABEND: Termination without Abend

When a parameter error or a functional error occurs while this utility function is running, the utility ordinarily prints an error message and terminates with user abend 34 (with a dump) or user abend 35 (without a dump). If NOUSERABEND is specified, the utility will not abend after printing the error message. Instead, the message "*utility* TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

> **Note:** When NOUSERABEND is specified, we recommend that it be specified as the first parameter of the utility function (before all other parameters). This is necessary to ensure that its parameter error processing occurs properly.

TEST: Test Syntax

The TEST parameter tests the operation syntax without actually performing the operation. Note that the validity of values and variables cannot be tested: only the syntax of the specified parameters can be tested. See section *Adabas for Mainframes* documentation > *Utilities* > *ADACHK Utility: Database Consistency Check Utility that Runs Concurrently with Normal Database Operations* > *Functional Overview* > *Syntax Checking with the TEST Parameter* for more information about using the TEST parameter in ADACHK functions.

The NOUSERABEND, TEST, and ABEND34 parameters affect the processing of the entire ADACHK run.

Example

In the following example, audit log RABNs 1, 2, 30-31, and 2000 are printed from DD/ALOG1.

```
ADACHK ALOGPRINT RABN=1,2,30-31,2000
```

ADACMP COMPRESS AUDITNM=

The ADACMP COMPRESS parameter AUDITNM= is used to assign an Audit Name to an Adabas file when it is compressed by ADACMP.

The Audit Name is an 8-character alphanumeric identifier used to track auditing information for Adabas files and must be unique across all Adabas files used in auditing. The Audit Name must be enclosed in single quotes if it contains embedded blanks.

Example

Assign the Audit Name ACCOUNTS to file 100:

```
ADACMP COMPRESS FILE=100,AUDITNM=ACCOUNTS
```

ADADBS ADDALOG

The ADDALOG function allows you to dynamically add a new audit log (ALOG) data set without terminating your current Audit Server session. Using this utility function, you can specify up to eight ALOG data sets. This will reduce the chances of a wait condition in the Audit Server when the Audit Server waits for an available ALOG. You might find this particularly useful during busier times of the month or year.

To add an ALOG data set dynamically, the Audit Server must know about its JCL at startup time. We recommend that you set up your Audit Server startup jobs to include definition statements for the maximum number of ALOG data sets as you plan to use, but limit the actual usage of the ALOGs using the ADARUN NALOG parameter. For example, you might start a nucleus with eight ALOG definitions in the Audit Server startup JCL, but limit the number of ALOGs actually used during Audit Server processing to three ALOGs by setting the NALOG parameter to "3". When the Audit Server starts up, only three ALOGs will be opened and logged in the PPT, even though eight are defined in the JCL. The additional ALOG data sets can then be dynamically added using this ADADBS ADDALOG utility.

> 📄 **Note:** Any ALOG data sets you add dynamically will not be retained once you recycle your Audit Server. To retain these new ALOG data sets when the Audit Server is stopped and restarted, alter the Audit Server startup JCL as well. You must ensure that the number of ALOG definition statements in the JCL matches the increased number of ALOG data sets and that the NALOG ADARUN parameter setting includes the new ALOG data sets.

Running the ADADBS ADDALOG utility function is invalid when the Audit Server is running with dual ALOGs.

```
ADADBS ADDALOG NUMBER=alog-ds-number
                    [NOUSERABEND]
                    [ALOGDEV=device-type]
                    [TEST]
```

The following sections cover the topics:

- Essential Parameters
- Optional Parameters
- Example

Essential Parameters

NUMBER: ALOG Data Set Number

Use the NUMBER parameter to specify the number of the nonsequential ALOG data set to be added. Valid values are integers ranging from "2" through "8".

> 📄 **Note:** Be sure that the Audit Server startup JCL allows for this additional ALOG data set by including a definition statement for the data set. If a definition statement is not already specified for this ALOG data set in the Audit Server startup JCL, you will need to add it now and recycle the server. Ideally, you would already have included definition statements in the JCL for all potential ALOG data sets, even though they are not all in use when the Audit Server starts up.

Optional Parameters

ALOGDEV

Use the optional ALOGDEV parameter to specify the device type to be used for the new ALOG data set. This parameter is required only if a different device type from the device type specified by the ADARUN DEVICE parameter is to be used. The default is to use the device type specified by the ADARUN DEVICE parameter.

NOUSERABEND: Termination without Abend

When a parameter error or a functional error occurs while this utility function is running, the utility ordinarily prints an error message and terminates with user abend 34 (with a dump) or

user abend 35 (without a dump). If NOUSERABEND is specified, the utility will not abend after printing the error message. Instead, the message "*utility* TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

> **Note:** When NOUSERABEND is specified, we recommend that it be specified as the first parameter of the utility function (before all other parameters). This is necessary to ensure that its parameter error processing occurs properly.

TEST: Test Syntax

The TEST parameter tests the operation syntax without actually performing the operation. TEST checks only the syntax of the specified parameters; not the validity of values and variables. See the *Adabas for Mainframes* documentation > *Utilities* > *ADADBS Utility: Database Services* > *Functional Overview* > *Syntax Checking with the TEST Parameter* for more information about using the TEST parameter in ADADBS functions.

Example

In the following example, ALOG data set 3 is dynamically added using a 3390 device.

```
ADADBS ADDALOG NUMBER=3,ALOGDEV=3390
```

ADADBS AUDITING

The ADADBS AUDITING function provides activation and deactivation of Adabas files that are participating in auditing. This function should be used for Adabas databases with auditing enabled. It is not valid for use with Adabas Audit Servers.

```
ADADBS AUDITING {ACTIVATE | DEACTIVATE }
                            FILE=filenumber
                            NUCID=nucid  ↵
```

The following sections cover the topics:

- Essential Parameters
- Optional Parameters

■ Examples

Essential Parameters

One of the parameters, ACTIVATE or DEACTIVATE, as well as the FILE parameter must be set. There is no default.

ACTIVATE: Adabas Database File Activation Request

The ACTIVATE parameter requests activation for the specified file. The specified file must not be actively participating in auditing.

DEACTIVATE: Adabas Database File Deactivation Request

The DEACTIVATE parameter requests deactivation for the specified file. The specified file must be actively participating in auditing.

FILE: Adabas Database File Number

The FILE=*filenumber* parameter indicates the ADABAS file to be activated or deactivated.

Optional Parameters

NUCID: Nucleus ID for a Cluster Database

The NUCID=*nucid* parameter identifies the nucleus ID in a cluster database. It applies only to cluster databases and is not valid for non-cluster databases.

Examples

The following example requests that the Adabas Database deactivate file 100 from sending audit information to the Audit Server.

```
ADADBS AUDITING ACTIVATE,FILE=100
```

The following example requests that the Adabas Database deactivate file 100 from sending audit information to the Audit Server.

```
ADADBS AUDITING DEACTIVATE,FILE=100
```

ADADBS AUDITSERVER

The ADADBS AUDITSERVER function provides activation, deactivation, open, and close control of Adabas Audit Server resources. It also provides a means to display the databases and files that audit information is being collected for. This function should be used with Adabas Audit Servers; it is not valid for use with Adabas databases.

```
ADADBS AUDITSERVER {ACTIVATE | DEACTIVATE | OPEN | CLOSE | DISPLAYDB}
                        DESTINATION=dest-name
                        SUBSCRIPTION=sub-id  ↵
```

The following sections cover the topics:

- Essential Parameters
- Optional Parameters
- Examples

Essential Parameters

One of the parameters, ACTIVATE, DEACTIVATE, OPEN, CLOSE, or DISPLAYDB must be set. There is no default.

ACTIVATE: Audit Server Resource Activation Request

The ACTIVATE parameter requests activation for the specified destination or subscription.

DEACTIVATE: Audit Server Resource Deactivation Request

The DEACTIVATE parameter requests deactivation for the specified destination or subscription.

OPEN: Audit Server Resource Open Request

The OPEN parameter requests that a closed destination be opened. When this parameter is specified, the DESTINATION parameter must be specified.

CLOSE: Audit Server Resource Close Request

The CLOSE parameter requests that an open destination be closed. When this parameter is specified, the DESTINATION parameter must be specified.

DISPLAYDB: Display Audit Server Databases and Files Request

The DISPLAYDB parameter requests that all databases and files known to the audit server be listed. Neither DESTINATION nor SUBSCRIPTION are specified.

> **Note:** The source Adabas Database may or may not be actively generating audit information for the files listed.

Optional Parameters

DESTINATION: Audit Destination

The DESTINATION parameter can only be specified when the ACTIVATE, DEACTIVATE, OPEN, or CLOSE parameters are specified. The DESTINATION parameter supplies the name of the destination that should be activated, deactivated, opened, or closed. The destination specified must be defined to the Audit Server.

If ACTIVATE is specified, the destination must already be inactive. If DEACTIVATE is specified, the destination must already be activated.

If OPEN is specified, the destination must be in a closed state. If CLOSE is specified, the destination must be in an open state.

The DESTINATION parameter is mutually exclusive with the SUBSCRIPTION parameter.

SUBSCRIPTION: Audit Subscription

The SUBSCRIPTION parameter can only be specified when the ACTIVATE or DEACTIVATE parameters are specified.

The subscription specified for the SUBSCRIPTION parameter must be defined to the Audit Server. It specifies the ID of the subscription definition to use.

If ACTIVATE is specified, the subscription must already be deactivated. If DEACTIVATE is specified, the subscription must already be activated.

The SUBSCRIPTION parameter is mutually exclusive with the DESTINATION parameter.

Examples

The following example requests that the Audit Server activate the destination defined by the DEST0001 destination. The destination is currently inactive.

```
ADADBS AUDITSERVER ACTIVATE,DESTINATION=DEST0001
```

The following example requests that the Audit Server activate the subscription defined by the SUBS0001 subscription. The subscription is currently inactive.

```
ADADBS AUDITSERVER ACTIVATE,SUBSCRIPTION=SUBS0001
```

The following example requests that the Audit Server deactivate the destination defined by the DEST0001 destination. The destination is currently active.

```
ADADBS AUDITSERVER DEACTIVATE,DESTINATION=DEST0001
```

The following example requests that the Audit Server deactivate the subscription defined by the SUBS0001 subscription. The subscription is currently active.

```
ADADBS AUDITSERVER DEACTIVATE,SUBSCRIPTION=SUBS0001
```

The following example requests that the Audit Server open destination DEST0001.

```
ADADBS AUDITSERVER OPEN,DESTINATION=DEST0001
```

The following example requests that the Audit Server close destination DEST0001.

```
ADADBS AUDITSERVER CLOSE,DESTINATION=DEST0001
```

ADADBS DELALOG

The DELALOG function allows you to dynamically delete an audit log (ALOG) data set without terminating your current Audit Server session.

> **Note:** Any ALOG data sets you delete dynamically may reappear once you recycle your Audit Server. To ensure the ALOG data set is dropped when the Audit Server is stopped and restarted, alter the Audit Server startup JCL as well, ensuring that the NALOG ADARUN parameter setting is reduced to account for the dropped ALOG data sets.

Running the ADADBS DELALOG utility function is invalid when the Audit Server is running with dual ALOGs.

```
ADADBS DELALOG NUMBER=alog-ds-number
                            [NOUSERABEND]
                            [TEST]
```

The following sections cover the topics:

- Essential Parameters
- Optional Parameters

- Example

Essential Parameters

NUMBER: ALOG Data Set Number

Use the NUMBER parameter to specify the number of the nonsequential ALOG data set to be deleted. Valid values are integers ranging from "2" through "8".

Optional Parameters

NOUSERABEND: Termination without Abend

When a parameter error or a functional error occurs while this utility function is running, the utility ordinarily prints an error message and terminates with user abend 34 (with a dump) or user abend 35 (without a dump). If NOUSERABEND is specified, the utility will not abend after printing the error message. Instead, the message "*utility* TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

> **Note:** When NOUSERABEND is specified, we recommend that it be specified as the first parameter of the utility function (before all other parameters). This is necessary to ensure that its parameter error processing occurs properly.

TEST: Test Syntax

The TEST parameter tests the operation syntax without actually performing the operation. TEST checks only the syntax of the specified parameters; not the validity of values and variables. See the *Adabas for Mainframes* documentation > *Utilities* > *ADADBS Utility: Database Services* > *Functional Overview* > *Syntax Checking with the TEST Parameter* for more information about using the TEST parameter in ADADBS functions.

Example

In the following example, ALOG data set 3 is dynamically deleted from its 3390 device.

```
ADADBS DELALOG NUMBER=3,ALOGDEV=3390
```

ADADBS MODFCB AUDITNM=

The ADADBS MODFCB parameter AUDITNM= is used to assign an Audit Name to an Adabas file that is participating in auditing.

The Audit Name is an 8-character alphanumeric identifier used to track auditing information for Adabas files and must be unique across all Adabas files used in auditing. The Audit Name must be enclosed in single quotes if it contains embedded blanks.

Example: Assign the Audit Name PAYROLL to file 100

```
ADADBS MODFCB FILE=100,AUDITNM=PAYROLL
```

ADADBS OPERCOM AUDCONNECT

This command is used to dynamically force a connection attempt to a specific Auditing Server or Adabas database ID, or to all related Auditing Server or Adabas database IDs.

```
ADADBS OPERCOM AUDCONNECT={dbid|ALL}
```

The following sections cover the topics:

- Essential Parameters

Essential Parameters

Either of the two parameters, *dbid* or ALL, must be set. There is no default.

dbid: **Database ID**

The specified Auditing Server or Adabas database ID.

ALL

All known Auditing Server or Adabas database IDs.

ADADBS OPERCOM DAUDPARM and DAUDSTAT

ADADBS OPERCOM has two new functions: DAUDPARM and DAUDSTAT. These functions are available for Adabas Audit Servers and Adabas nuclei.

- ADADBS OPERCOM DAUDPARM
- ADADBS OPERCOM DAUDSTAT

ADADBS OPERCOM DAUDPARM

Use `DAUDPARM` to display the audit parameters for an Adabas nucleus (with auditing turned on) or for an Adabas Audit Server.

When DAUDPARM is issued against an Adabas nucleus (with auditing turned on), the parameter definitions listed include global parameter definitions and file related parameter definitions.

For example:

```
ADADBS OPERCOM DAUDPARM

Auditing definitions:

Global definitions:
  Connect count .... :            10
  Connect interval   :            60
  Audit pool warning:
   Message interval   :            50
   Message limit .... :            20
   Warn increment ... :            12
   Warn percent ..... :            70

File definitions:

File   100
  Server ID ........ :    1702
  Access commands:
   ACBX ............ : Fields
   Data Storage..... : No
   Client info ..... : Fields
   Format buffer ... : Yes
   Search buffer ... : No
   Value buffer .... : Yes
  Delete commands:
   ACBX ............ : No
   Data Storage..... : Yes
   Client info ..... : No
  Insert commands:
   ACBX ............ : No
```

```
  Data Storage..... :  Yes
  Client info ..... :  No
  Format buffer ... :  Yes
 Update commands:
  ACBX ............ :  No
  Before Image .... :  Fields
  After Image ..... :  Yes
  Client info ..... :  No
  Format buffer ... :  Yes
 Access Fields: AA,AB,AC,AD,AE,AF,AG,AH,AI,AJ,AK,AL,
                AM,AN,AO,AP,AQ,AR,AS,AT,AU,AV,AW
 Update Fields: AA,AB,AC,AD

File  101
  Server ID ........ :   1702
 Access commands:
  ACBX ............ :  Yes
  Data Storage..... :  Fields
  Client info ..... :  Yes
  Format buffer ... :  No
  Search buffer ... :  Yes
  Value buffer .... :  No
 Delete commands:
  ACBX ............ :  Yes
  Data Storage..... :  No
  Client info ..... :  Yes
 Insert commands:
  ACBX ............ :  Yes
  Data Storage..... :  No
  Client info ..... :  Yes
  Format buffer ... :  No
 Update commands:
  ACBX ............ :  Fields
  Before Image .... :  Yes
  After Image ..... :  No
  Client info ..... :  Fields
  Format buffer ... :  No
 Access Fields: A1,A2,A3,A4,A5,A6,A7,A8,A9,AA,AB,AC
 Update Fields: AA
```

When DAUDPARM is issued against an Adabas Audit Server, the following parameters are listed:

■ Global Parameters

■ Destinations

■ Subscriptions

　■ Format buffers used

　■ Filters used

　■ Filter definitions

For example:

```
ADADBS OPERCOM DAUDPARM

Audit Server definitions:

Global definitions:
  Audit pool warning:
   Increment ....... :               10
   Message interval  :              120
   Message limit ... :                6
   Percent ......... :               60
  Broker stub name   : BROKER
  Connect count .... :                3
  Connect interval   :               30
  Maximum output size:          100,000
  Open at start .... : YES
  Open retry count   :               10
  Open retry interval:                0
  Queue full delay   :               60
  Subtasks ......... :                0
  Subtask wait time  :               10
  Log input ........ : NO

 Audit Destination PAYROLL                                                    ↵

   Active at start up : Yes
   Open at start .... : Yes
   Allow logging .... : Yes

Subscription PAYROLLS
  Active at start up : Yes
  Destination(s) ... : PAYROLL

  DBID / File ...... :  1701 /     1
   Data origin       : Mainframe
   Read/Find items . : Yes
      Request info    : Yes
      Client info .   : Yes
      Format buffer   : Yes
      Search buffer   : No
      Value buffer    : No
    Insert items .... : Yes
      Request info    : Yes
      Client info .   : Yes
      Format buffer   : Yes
    Update items .... : Yes
      Request info    : Yes
      Client info .   : Yes
      Format buffer   : Yes
      Data storage BI : No
```

```
   Delete items .... : Yes
     Request info    : Yes
     Client info .   : Yes
Data AI global format      : PAYROL
Data BI global format      : PAYROL
Data filter format         : PAYROL
Data filter                : PAYFLTR
 Include filter: PAYFLTR
  Group:    1
   Condition:    1
    Source Field: AT
     Begin byte    1, Length    9
     Before Image
     After Image
      PE   1, MU   2
         greater than or equal to
    Target Field: AT
     Begin byte    1, Length    9
     Before Image
     After Image
      PE   1, MU   3
LUW Client info format     : CLIENTL
Client info global format  : CLIENT
ACBX global format         : REQUEST
```

ADADBS OPERCOM DAUDSTAT

Use DAUDSTAT to display the audit statistics for an Adabas nucleus (with auditing turned on) or for an Adabas Audit Server.

When DAUDSTAT is issued against an Adabas nucleus (with auditing turned on), the statistics listed include:

- The global statistics

  - The total number of audit items completely processed

  - The current number of pending audit items (items that have been committed, but not yet processed)

  - The current number of incomplete items that will be audited (but are not yet committed)

  - The number of items rejected due to error

- The file related statistics for every file

  - The total number of audit items completely processed for a file

  - The current number of pending audit items for a file (items that have been committed, but not yet processed)

  - The current number of incomplete items for a file that will be audited (but are not yet committed)

When DAUDSTAT is issued against an Adabas Audit Server, the statistics related to destinations, global values, and subscriptions in the database are listed.

ADADBS OPERCOM FEOFAL

The ADADBS OPERCOM FEOFAL command closes the current dual or multiple audit log and switches to another audit log. This command is valid only if dual or multiple audit logging is in effect and only for Adabas Audit Servers.

In nucleus cluster environments, the GLOBAL option can be used to close and switch dual or multiple audit logs in all active cluster nuclei.

Command Syntax:

```
ADADBS OPERCOM FEOFAL[,GLOBAL]
```

ADADEF DEFINE AUDITSERVER Function

The AUDITSERVER parameter controls whether the database you are defining is an Adabas database or an Adabas Audit Server.

AUDITSERVER: Set the Adabas Audit Server

The AUDITSERVER parameter is used with ADADEF DEFINE to control whether a normal Adabas database is being defined, or whether an Adabas Audit Server is being defined.

Possible values are "YES" or "NO" (the default). Specify "YES" to define an Adabas Audit Server and "NO" to define a normal Adabas database.

Examples

Define an Adabas Audit Server:

```
ADADEF DEFINE AUDITSERVER=YES
```

Define a normal Adabas database:

```
ADADEF DEFINE AUDITSERVER=NO
```

ADADEF MODIFY AUDITSERVER Function

The AUDITSERVER parameter controls whether the database that is running is an Adabas database or an Adabas Audit Server.

AUDITSERVER: Set the Adabas Audit Server

The AUDITSERVER parameter is used with ADADEF MODIFY to control whether the running database is to be a normal Adabas database or an Adabas Audit Server.

Possible values are "YES" or "NO" (the default). Specify "YES" to modify a normal Adabas database to run as an Adabas Audit Server and "NO" to modify an Adabas Audit Server to run as a normal Adabas database.

Examples

Modify a normal Adabas database to run as an Adabas Audit Server:

```
ADADEF MODIFY AUDITSERVER=YES
```

Modify an existing Adabas Audit Server to run as a normal Adabas database:

```
ADADEF MODIFY AUDITSERVER=NO
```

ADAFRM ALOGFRM

The ALOGFRM function allows you to format audit logs (ALOGs) in a similar way that PLOGFRM is used to format protection logs (PLOGs).

Formatting must be performed before any new audit log data set can be used by the Audit Server.

```
ADAFRM ALOGFRM SIZE=size
    [DEVICE=device-type]
    [{FROMRABN={starting-rabn | NUMBER={dataset-number|1}}]
    [NOUSERABEND]
    [VOLIOCOUNT=nnn|4]
```

The following sections cover the topics:

- Essential Parameters
- Optional Parameters

- Example

Essential Parameters

SIZE: Size of Area to be Formatted

SIZE specifies the size of the area to be formatted (or reset). Blocks (a decimal value followed by a "B") or cylinders may be specified.

Optional Parameters

DEVICE: Device Type

DEVICE is the physical device type upon which the area to be formatted is contained. If DEVICE is not specified, the device type specified by the ADARUN DEVICE parameter is used.

FROMRABN: Starting RABN

FROMRABN specifies the RABN at which formatting is to begin. This parameter may only be used for an existing data set. NUMBER cannot be specified in the same ADAFRM job as FROMRABN.

When FROMRABN is specified with the ALOGFRM function, formatting begins at the FROMRABN point and continues up to the highest complete track before the RABN computed from FROMRABN + SIZE (assuming a size specified in or converted to blocks). This means that the last track within the specified range (FROMRABN + SIZE) will be formatted only if all the track's RABNs are within that range.

On z/OS, FROMRABN should only be used to reformat existing blocks as the last record pointer in the VTOC cannot be modified by function FROMRABN.

NOUSERABEND: Termination without Abend

When a parameter error or a functional error occurs while this utility function is running, the utility ordinarily prints an error message and terminates with user abend 34 (with a dump) or user abend 35 (without a dump). If NOUSERABEND is specified, the utility will not abend after printing the error message. Instead, the message "*utility* TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

> **Note:** When NOUSERABEND is specified, we recommend that it be specified as the first parameter of the utility function (before all other parameters). This is necessary to ensure that its parameter error processing occurs properly.

NUMBER: Data Set Number

NUMBER selects the nonsequential audit log to be formatted. The default is 1 (first data set). Values allowed are 1 through 8. ADAFRM ALOGFRM function statements cannot specify (and

will not default to) a NUMBER value if other ADAFRM statements in the same job specify a FROMRABN value.

NUMBER must match the number suffix of the related ALOG data definition (DD) statement.

VOLIOCOUNT: Number of Concurrent I/Os per Volume per Data Set

VOLIOCOUNT specifies the number of concurrent I/Os per volume to process for a data set. The minimum value you can specify is "1"; maximum is "256". The default value is "4". The setting of this parameter can affect the processing speed of the ADAFRM utility.

Example

In the following example, one cylinder for nonsequential audit log data set 1, and 1 cylinder for nonsequential audit log data set 2 are to be formatted.

```
ADAFRM ALOGFRM SIZE=1,DEVICE=3390,NUMBER=1
ADAFRM ALOGFRM SIZE=1,DEVICE=3390,NUMBER=2
```

ADALOD LOAD Parameters

ADALOD LOAD has additional auditing-related parameters for both the Adabas Audit Server and Adabas databases.

- ADALOD LOAD Parameters for the Adabas Audit Server
- ADALOD LOAD Parameters for the Adabas Database

ADALOD LOAD Parameters for the Adabas Audit Server

The following parameters are applicable to the loading of auditing-related system files into the Adabas Audit Server.

AUDITING: Loading the Auditing system file

The AUDITING parameter is used to request the load of an Auditing system file into the Audit Server. It may not be specified for loading a file on a normal Adabas database.

The Auditing system file stores auditing-related definitions and is maintained by Adabas Auditing Configuration. When it is loaded into the Audit Server, it can be read during Audit Server start-up.

The contents of DD/EBAND are ignored when loading an Auditing system file.

For more information about Adabas system files, read the FILE parameter of the ADALOD LOAD function in the *Adabas for Mainframes* documentation > *Utilities* > *ADALOD Utility: File Loader*.

SLOG: Loading the Auditing Subscription Logging (SLOG) system file

The SLOG parameter is used to request the load of an Auditing SLOG system file into the Audit Server. It may not be specified for loading a file on a normal Adabas database.

The use and maintenance of the Auditing SLOG system file is controlled by Adabas Auditing Configuration.

The contents of DD/EBAND are ignored when loading an Auditing system file.

> **Note:** User application files should not be loaded on the Adabas Audit Server.

Examples:

Load the Auditing system file into the Audit Server:

```
ADALOD LOAD FILE=8,AUDITING
```

Load an Auditing SLOG system file into the Audit Server:

```
ADALOD LOAD FILE=30,SLOG
```

ADALOD LOAD Parameters for the Adabas Database

The ADALOD LOAD parameter AUDITNM= is used to assign an Audit Name to an Adabas file when it is loaded by ADALOD.

The Audit Name is an 8-character alphanumeric identifier used to track auditing information for Adabas files and must be unique across all Adabas files used in auditing. The Audit Name must be enclosed in single quotes if it contains embedded blanks.

Example: Assign the Audit Name VEHICLES to file 100

```
ADALOD LOAD FILE=100,AUDITNM=VEHICLES
```

ADAORD STORE

ADAORD STORE has the following additional auditing-related parameters for use with the Adabas Audit Server.

AUDITING: Store the Auditing system file

The AUDITING parameter is used to request the store of the Auditing system file from the DD-FILEA/FILEA tape as the new Auditing system file for the Audit Server. The new Auditing system

file must have the same file number as the old Auditing system file. AUDITING may not be specified for storing a file on a normal Adabas database.

When the AUDITING parameter is not specified, the Auditing system file on the DDFILEA/FILEA is not stored in the Audit Server, even if it is specified by a FILE or ALLFILES parameter.

SLOG: Store the Auditing Subscription Logging (SLOG) system file

The SLOG parameter is used to request the store of the Auditing SLOG system file from the DD-FILEA/FILEA tape as the new Auditing SLOG system file for the Audit Server. The new Auditing SLOG system file must have the same file number as the old Auditing SLOG system file. SLOG may not be specified for storing a file on a normal Adabas database.

When the SLOG parameter is not specified, the Auditing SLOG system file on the DDFILEA/FILEA is not stored in the Audit Server, even if it is specified by a FILE or ALLFILES parameter.

> **Note:** User application files should not be loaded on the Adabas Audit Server.

Examples:

Store the Auditing system file from DDFILEA/FILA into the Audit Server:

```
ADAORD STORE FILE=8,AUDITING
```

Store the Auditing SLOG system file from DDFILEA/FILEA into the Audit Server:

```
ADAORD STORE ALLFILES,SLOG
```

ADAREP REPORT

The ADAREP utility displays information about the status of auditing for the database, files, and the Adabas Audit Server.

The following sections cover the topics:

- General Information Section
- File Information Section

- Checkpoint Information Section

General Information Section

When ADAREP is run against an Audit Server, the general information section includes the following additional information:

When the Audit Server is defined:

```
Auditing            = Yes
```

When an Auditing system file is loaded:

```
Auditing File       = <file number>
```

When an Auditing SLOG system file is loaded:

```
Audit SLOG File     = <file number>
```

File Information Section

The file information section includes the following when an Audit Name is defined for a file in an Adabas database:

```
Audit Name          <auditname>
AUDIT-ID            <timestamp when audit Name first created>
```

Note:

The Audit Name is the value assigned by:

- The ADALOD utility with ADALOD LOAD AUDITNM=

- The ADADBS utility with ADADBS MODFCB AUDITNM=

- The ADACMP utility with ADACMP COMPRESS AUDITNM=

Example Report

```
Audit Name          MYAUDTID
AUDIT-ID            2021-01-05 22:59:09.387214
```

Checkpoint Information Section

The following new auditing-related checkpoints are provided if the CPLIST or CPEXLIST parameters are specified:

Type	Name	Originator	Description
85	SYNS	ADADBS	ADADBS AUDITING function
86	SYNS	ADADBS	ADADBS AUDITSERVER function

ADARES ALCOPY

The ALCOPY function allows you to convert dual/multiple audit logs to sequential logs. The data set that has the earlier time stamp is copied to a sequential data set. Once the ALCOPY function is completed successfully, the copied data set is marked as empty. This function may, therefore, be used only once for any given data set.

Once the ADARES ALCOPY job has run for an ALOG data set, the ADARES utility checks the PPT to determine whether any additional ALOG data sets need to be copied. If so, it invokes the Audit Server user exit 12 to accommodate the number of data sets that need copying. For example, if NALOG=8, once the initial ALCOPY job completes, the ADARES utility will issue a call to the Audit Server to invoke user exit 12 for each uncopied ALOG data set it detects. When it invokes user exit 12, it uses the flag "F" to identify this type of call. This flag is processed as a new EX12TYPE value in user exit 12.

For more information about user exit 12, see *Adabas for Mainframes > User, Hyperdescriptor, Collation Descriptor, and SMF Exits > User Exit 12 (Multiple Data Set Log Processing)*.

```
ADARES ALCOPY [DUALALD=device-type]
    [NOUSERABEND]
    [OPENOUT]
    [TEST]
    [TWOCOPIES]
```

The following sections cover the topics:

- Optional Parameters

- Example

Optional Parameters

ADARES ALCOPY can be specified with no parameters.

DUALALD: Dual Audit Log Device Type

DUALALD specifies the device type used for the dual audit log data sets. This parameter is required if the device type used for the audit log data set is different from the one specified with the ADARUN DEVICE parameter.

NOUSERABEND: Termination without Abend

When a parameter error or a functional error occurs while this utility function is running, the utility ordinarily prints an error message and terminates with user abend 34 (with a dump) or user abend 35 (without a dump). If NOUSERABEND is specified, the utility will not abend after printing the error message. Instead, the message "*utility* TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

> **Note:** When NOUSERABEND is specified, we recommend that you specify it as the first parameter of the utility function (before all other parameters). This way you ensure that NOUSERABEND's parameter error processing occurs properly.

OPENOUT: Open DDSIAUS1/2 or SIAUS1/2 Data Sets

The OPENOUT parameter specifies that ADARES opens the DD/SIAUS1/2 output data sets, even if no data is actually to be copied. Without OPENOUT, the sequential output data sets are not opened if ADARES detects an end-of-file condition while attempting to read the first input record. This may cause problems in some operating system environments. With OPENOUT, the output data sets are opened before the first input record is read.

TEST: Test Syntax

The TEST parameter tests the operation syntax without actually performing the operation. TEST checks only the syntax of the specified parameters; not the validity of values and variables.

TWOCOPIES: Create Two Copies of Output

TWOCOPIES causes two copies of the output to be created.

Example

In the following example, the oldest dual audit log is to be copied to a sequential data set.

```
ADARES ALCOPY
```