UDBAR

DB2 UDB for LUW Access Recording

Software Support: sup@sprdb2.com

1 Product Description

UDBAR is a software tool that provides collecting, archiving and scanning services for the audit data produced by DB2 LUW, operating in the Linux and zLinux environments.

1.1 UDBAR functions for the Linux environment

These functions process the output files produced by the DB2AUDIT EXTRACT command and forward the processed output using FTP to the UDBAR server UDBARP, executing in the z/VM environment.

1.2 UDBAR functions for the z/VM environment

The software components executing in the z/VM environment provide following functions:

- The **UDBAR Processor** receives the audit data FTP-ed from the Linux environment. Audit data processed are stored into the UDBAR Audit Log, which is a CMS file, compatible with the SQLAF AUDIT1 record structure.

- The **UDBAR Archiving Program** transfers the current Audit Log to a sequential dataset, typically on magnetic tape.

- The **UDBAR Audit Log Scan** function allows to search the Audit Log or an Audit Archive interactively or in batch.

2 Installing UDBAR

2.1 Prerequisites

- Acquire a minidisk or SFS directory where UDBAR will be installed. The software requires about 10 3390-cylinders on the product disk.

- Ensure that the VM userid performing the installation has write access to the UDBAR product disk.

2.2 Process the distribution material

Note that the following paragraphs are also contained in the READ.ME file, which is part of the distribution file.

UDBAR is distributed as a ZIP file. Process the ZIP file as follows:

- create a PC directory for the product
- transfer the ZIP file to the PC directory
- change to the directory and unzip the file
- after unzip, your directory should contain:

    - the evaluation agreement, if an evaluation installation
      (When installing and using the software, you automatically agree with the terms in this agreement.)
    - the READ.ME file
    - the product manual in PDF format
    - the zLinux procedures
    - the z/VM materials, with the installation program

Proceed as follows:

- Ensure that the VM userid performing the installation has write access to the UDBAR product disk.

- Change to the "VM_Materials" directory and execute the UDBARFTP.EXE by double-clicking it.

- UDBARFTP establishes an FTP session to store the z/VM related product components to the UDBAR product disk.

- At the start of this FTP session, you will be requested to enter:

    - the IP address of your z/VM system
    - the z/VM userid that will own the UDBAR product disk
    - the password of the above userid

- When the FTP session has completed, you should continue installation as described below.

2.3 Install the zLinux components

- Create the Linux directory **/usr/udbar/archive**.
- Copy the folllowing files from the distribution directory **/Linux_Procedures** to the **/usr/udbar/archive** directory.

2.4 Install the CMS components

With the UDBAR product disk accessed in filemode A, enter **UDBARPI** at the CMS prompt. The UDBAR modules are now linked onto the product disk.

3 UDBAR Setup

3.1 Configuring the DB2_LUW audit facility

Issue the following command in the zLinux DB2 environment:

db2audit configure archivepath /usr/udbar/archive

3.2 Defining the audit policy to DB2_LUW

To define the UDBAR audit policy, issue the following SQL statement to the target database:

```
CREATE AUDIT POLICY UDBARS
CATEGORIES EXECUTE WITH DATA
STATUS BOTH
ERROR TYPE NORMAL
```

3.3 Start the DB2_LUW audit facility

The DB2_LUW audit facility must be started using the command **db2audit start**. The command **db2audit stop** will terminate auditing.

3.4 Configuring the UDBARFTP script

Edit the /usr/udbar/archive/udbarftp script and specify the following variables:

- export IP_ADDR=<IP-address of z/VM>
- export FTP_USER_PASSWD=<password of the UDBARP VM-id>

3.5 Setup the Audit Processor

- Create a virtual machine named **UDBARP**. It will run the **Audit Processor** function (the **UDBARP** program).
- The virtual storage for this machine should be specified as 16 MB at least. A virtual storage size of 32 Mb is recommended.
- It is suggested that the VM single console facility be enabled for this machine.
- In the PROFILE of the Audit Processor, insert the CMS commands necessary to access the UDBAR product disk.
- The UDBARP virtual machine will hold the Audit Log UDBAR AUDIT1.

3.6 Space requirements for the Audit Log

For each audited table, there will be a log entry:

- for each SELECT, INSERT, DELETE or UPDATE statement
- for each cursor-based SELECT opened
  (there are no audit log entries for cursor FETCH statements)

Each audit event stored on the log dataset UDBAR AUDIT1 requires 132 bytes of disk space plus the text of the audited SQL statement. Assuming a statement text length of 380 characters as an average, a log record would require about 512 bytes, giving the following space estimates:

Unit of space (in 4K CMS blocks)	**Number of log records held**
one CMS block	8
one 3380 cylinder	1200
one 3390 cylinder	1440
one Megabyte of disk space	2048

3.7 Starting the Audit Processor

The Audit Processor runs in the virtual machine UDBARP and is started by including the **UDBARP** command in the PROFILE EXEC.

The syntax of the UDBARP command is as follows:

UDBARP [ARCHLIM n]

Where **ARCHLIM** specifies the archive threshold as the number of records to be written on the audit log (UDBAR AUDIT1) before automatic archiving occurs. If omitted, automatic archive occurs only when the primary log is full.

Prerequisites

- Prior to invoking UDBARP, access must have been established to the UDBAR product disk.

- The audit log is accessed with filemode A. If the audit log is SFS resident and if the audit directory has not been accessed as the **top directory** at IPL, you must issue the ACCESS command before starting the Audit Processor.

3.8 Defining auditing options

Additional processing options for the UDBAR program product can be defined in the CMS file UDBAR **OPTIONS**. This optional dataset should be located on the UDBAR product disk. If the dataset does not exist, default values will apply.

The UDBAR OPTIONS file may contain following statements:

3.8.1 Specifying an archive cycle

Syntax : ARCHIVE_CYCLE {DAY | WEEK | MONTH | YEAR}

During the archive procedure, the audit log is appended from disk to the current archive tape. The archive_cycle command allows you to specify whether you will keep an archive tape per day, per week, per month or per year. If you omit the statement, ARCHIVE_CYLE will be set to **MONTH**.

The archive procedure uses this specification

- to detect the begin of a new archiving cycle
- to request mounting of a new or an existing archive tape
- to create a new archive or to append to an existing archive tape
- to build the archive tape label

3.8.2 Specifying t he SFS directory of the audit log

Syntax : AUDIT_DIR <directory-name>

This statement should be specified only when the audit log dataset is located in an SFS directory. Since the Audit Processor writes the log and its other control files using filemode A, the entire A-disk should be placed in that directory. The AUDIT_DIR statement is provided to allow external components, such as the User Interface, to access this directory. (This is done by the UDBARLNK EXEC, using the CMS command **ACCESS directory-name filemode**). Therefore **directory-name** should specify the complete directory name, including the name of the filepool. If AUDIT_DIR is not specified, UDBAR assumes that the audit log is placed on the A-minidisk of the Audit Processor.

3.9 Initiating an audit archive scan server

The archive scan server allows users to send archive scan parameter files for processing in "batch" mode.

To setup the server, following actions should be taken:

- Define a virtual machine with an A-disk, large enough to contain the largest scan output file.
- Ensure that the server is autologged.
- In the PROFILE EXEC of the machine, insert the command:
    - **EXEC UDBARBTS [hh:mm]**
    - where hh:mm is the time when UDBARBTS should start processing all the scan parameter files it received in its reader. When a start time is not declared, processing starts as soon as a scan parameter file arrives.
- Unless you are using tape manager software, the server will issue messages to attach a tape drive and mount the archive tape(s). In this case, a secondary console should be defined for the server machine. If a tape manager is used, mounting and dismounting can be controlled using the UDBARUTX user exit.

4 UDBAR Operation

4.1 Archiving the DB2_LUW Log

At regular time intervals and for every audited database, the zLinux script udbar should be executed by submitting the Linux command **sh udbar <databasename>.** Alternatively, the Linux cron facility can be used to automatically schedule udbar at defined time intervals.[1]

The udbar script will perform the following functions:

- archive the named database (using db2audit archive)
- extract the archive to the DB2 files **auditlobs** and **execute.del** (using db2audit extract)
- execute the Java program UDBAR_Merge to postprocess the above extract files into a Linux file named **DByymmdd AUhhmmss**
- FTP the **DB* AU*** file to the UDBARP VM_id (using the udbarftp script)

4.2 Processing the DB2_LUW audit data in z/VM

At regular times the VM-user UDBARP should be autologged to process the archive files mentioned above.

When started, UDBARP will process all **DB* AU*** files FTPed from zLinux. UDBARP will store their data into the CMS file UDBAR AUDIT1. The latter file is format compatible with the SQL/AF product.

When all extract files have been processed, UDBARP will issue a logoff.

Since UDBARP is able to process several FTP-ed audit files in a single run, it is not necessary to start UDBARP as often as the Linux **udbar** script.

---

[1] **0 * * * * /usr/udbar/archive/udbar <databasename>** for example would schedule udbar every hour of every day.

5 UDBAR Log Scan

The LogScan program displays selected audit entries from the primary audit log (UDBAR AUDIT1 on the A-disk or directory of the Audit Processor) or from an audit archive tape.

Before invoking the UDBAR Log Scan from a z/VM virtual machine, ensure that access has been established to the UDBAR product disk. To start the scan, enter the command **UDBAR** at the CMS prompt. On the next screen, choose between scanning the primary log or a log archive.

5.1 Entering Audit Log scan criteria

At entry into log scan, the program displays the **search criteria panel** for the current log scan.

The upper part of the screen allows you to enter the scan values for the **audit record fields**.

The bottom lines of the screen are used to enter up to 4 **table column scan criteria** to be applied against the text of the audited SQL statements.

The search criteria specified at the previous invocation of the utility are displayed as current defaults, which you may accept or modify.

Following PFkeys can be used on the above panel:

PF1 Displays the help file.
PF3 Quits the selection panel.

5.2 Scan using log record fields

Following log record fields can be used as search arguments:

- name of database
- table creator and tablename
- statement type (enter *SELECT, INSERT, UPDATE* or *DELETE*)
- date of auditing as *yyyy-mm-dd*
- time of auditing as *hh:mm:ss*
- DB2 userid
- client user name
- package creator and package name
- statement's SQLCODE as *-nnn*
- related information
- LUW number

The field **Extend select** can be used to extend the current log scan selection set, resulting from a previous log scan during the current CMS session. The latest log scan selection data are always kept in memory, until the next CMS IPL. When extend select is set to **Y**, the newly selected log rows will be appended to the existing selection set.

Syntax rules

- When a criteria field is blank, no test is performed on that field during scan.
- By default an "equal" test is performed on criteria fields.
- However, a generic test is possible, as follows:
  - by specifying a trailing % sign, you can scan the leading positions of a record field
  - by specifying a leading % sign, you can test whether the specified scan string occurs in the field
  - by placing one of the logical operators >, <, >=, <=, or <> after the search value, a criteria field can be tested greater than, lower than or not equal.
- Scan values should **not** be enclosed in quotes.
- Multiple field scan criteria may be specified on the same panel. An audit record will be selected only when it satisfies **all** the specified criteria.

EXAMPLES

Package Name DRI005	selects accesses done by package DRI005
Package Name DRI%	selects package names starting with "DRI"
Package Name %SQL	selects package names containing "SQL"
Auditing Time 15>=	selects auditing times greater than or equal to 15h.

5.3 Scan using the audited statement text

Column-names and column-values appearing in the audited statement text may be used as scan criteria.

If a column-name is specified alone, audited statements are selected if the column-name appears in the statement text.

If both a column-name and a column-value are specified, selection will occur if the statement text refers to the column-name with the specified value.

Column-name / column-value pairs are recognized as follows:

SELECT, DELETE and UPDATE statements

- The columnname-columnvalue pair is searched in all WHERE predicate expressions having the format <column-name> <operator> <constant-value>. The scan logic examines the predicate operators = , ^=, <>, <, <=, >, >=, LIKE, IN and BETWEEN and uses them in matching the scan criterion.

- For example: the scan criterion **"COL 100"** will be satisfied by the WHERE clauses COL=100, COL<200, COL <> 50 etc..

- However in the present version of the product, ANDed and ORed combinations of WHERE predicate clauses are **not** handled: a scan terminates as soon as a *true* result has been obtained from a predicate clause.

For example: the search criterion "COL 100" will be satisfied for the predicate *WHERE COL > 50 AND COL < 200* but also for the predicate *WHERE COL > 50 AND COL < 90*, since in both cases the expression COL > 50 terminates the scan. The search criterion "COL 100<" on the other hand, will be satisfied for the predicate *WHERE COL > 50 AND COL < 90* only.

INSERT statements

The column-name is searched in the INSERT column-list and the column-value in the corresponding position of the VALUES clause.

UPDATE statements

In addition to an eventual UPDATE WHERE clause, the SET clauses are examined for "columnname, columnvalue" occurrences.

Syntax rules

- Column character values should **not** be enclosed in quotes.
- Numerical column values appear in the audited statement text as edited character strings and are scanned as such. Therefore, the following rules should be observed when supplying a numeric search value:
  - leading zeroes in numeric values should **not** be specified
  - trailing zeroes of a fractional value may be specified or omitted
  - negative values should start with a **-** sign
  - a decimal point is entered as a period
- You can perform a **generic** test on column values, as follows:
  - by specifying a trailing % sign, you can scan the leading positions of the value
  - by specifying a leading % sign, you can test whether the specified scan string occurs in the column value
  - testing whether a numeric column is negative can be done using the search strings **-%** or **%-** since a negative value is stored in the log as the +character string -nnn
- Up to 4 column scan criteria may be specified on the same panel. An audit entry will be selected only when it satisfies all the specified record field and column value criteria.

5.4 Processing the scan results

The results of the audit log scan are presented on the screen as a report. Following PFkey interface is provided to process the report.

PF1
Displays help.

PF2
Invokes the **Statement text** function. This function displays the full and formatted SQL statement text. Long statements may take several pages to display. In this case, use PF7 and PF8 to browse through the pages. Press PF3 to terminate the scan result list.

PF3
Terminates the scan utility.

PF4
Takes a hardcopy of the current report to the **UDBARPRT** dataset. The physical destination of the hardcopy dataset is determined by the **UDBARPFD** EXEC. By default, the hardcopy is directed to the virtual printer. The UDBARPFD EXEC may be modified to direct the output to another destination, such as a CMS file.

PF5
Performs the report formatting function.

When invoked, the report format function displays a function menu. Enter the code corresponding to the desired formatting function, as follows:

L moves the viewing window to the left margin
R moves the viewing window to the right margin
F reformats the report

The **F** function shows all columns in the report, preceded by a + sign if the column is currently displayed or by a - sign if the column has been hidden previously. You may override the + or - sign in the following manner:

Entering + unhides a previously hidden column, i.e. the column will be displayed again.

Entering - hides a column so that it is no longer displayed, although it remains in the report.

Entering < sorts the report on this column in ascending sequence (low to high). Only one ordering column can be designated.

Entering > sorts the report on this column in descending sequence (high to low). Only one ordering column can be designated.

Entering a number from 1 to 9 will move the column to the corresponding position in the report. The column previously on that position will take the position of the column moved.

PF6
Shows the first page of the report.

PF7
Shows the previous page of the report.

PF8
Shows the next page of the report.

PF9
Shows the last page of the report.

PF10
When in list mode, moves the viewing window to the left.
When in page mode, calls the **list search** function.

PF11
Moves the viewing window to the right.

PF12
Shows the report in "page" mode.

5.5 Searching the scan results

The search function is called by pressing **PF10** when the scan result list is displayed in **page mode.**

The search function displays the **search criteria panel** for entering following search arguments:

Search column
Enter the name of the list column to be searched. Use the column name that appears as column header on the display screen.

Search value
- Enter the value to find in the search column.
- The specified search value can be partial. The search column is examined over the length of the specified value only.
- By default, an **equal** search is performed. By prefixing the search value with one of the logical operators **<, >, <=, >=, ^=, <>** you can perform a not-equal, a lower-than or a greater-than search. Blanks may, but need not, intervene between the operator and the value.
- You can also specify **MIN** or **MAX** as the search value, to find the minimum or maximum value of the search column in the list.

Search direction
Enter the **>** sign to search the list in forward direction, which is the default.
Enter the **<** sign to search the list in backward direction.

Notes

(1) Except for MIN or MAX search, which always scans the entire list, searching starts at the position of the **current report line + 1**. When the search is productive, the current line is positioned on the list entry found. This allows to repeatedly apply the same search arguments.

(2) The last search criteria entered are redisplayed on the next search panel.

(3) Logical operators and MIN/MAX functions can also be applied to non-numerical list columns.

PF key definitions:
PF1 requests help
PF3 exits search

5.6 Scanning an Audit Archive

If the scan program does not find the UDBARUTX tape exit, it assumes that no tape managing software is available and it will request to manually attach a tape device to your virtual machine. When this has been done, press ENTER to continue.

Next, the program requests the label of the archive tape to process. The *"tape label"* screen field shows the format of the tape label, which depends on the ARCHIVE_CYCLE specification in the UDBAR OPTIONS file.

By default, a monthly archive tape cycle is assumed and the tape label will have the format **UDBAR-yyyy-mm**.

The tape label which you entered at the last invocation of the utility will be shown as the default.

Complete the tape label and press ENTER. Processing continues as for a primary audit log scan.

5.7 Scanning an Audit Archive in batch

5.7.1 Preparing the scan parameter file

Copy the file **UDBARBAT COPY** from the UDBAR product disk to your A-disk as <yourname> **UDBARBAT A**.

XEDIT <yourname> **UDBARBAT A** and complete the REXX variable assignments it contains. These REXX variables are used:

- To define the time of execution for your scan as **hh:mm** in the variable **Exec_time**. The execution time should be specified only if you intend to execute the scan in your own machine.
- To specify the label(s) of the archive tape(s) to be scanned in the variables **Filedef.1**, **Filedef.2** etc. There is no limit to the number of tape labels that can be named.
- To specify in the variable **Secuser** the name of the machine to which the tape attach and mount messages should be directed. If a tape manager is installed and the UDBARUTX EXEC has been coded, these messages will not be issued.
- To define the scan criteria in the variables **Database, Table_Creator, Table_Name, Statement_Type, Auditing_Date, Auditing_Time, DB2_Userid, Client_Userid, Package_Creator, Package_Name, SQLcode, Related** and **LUW_ID**.
- To define the eventual column names and values to be used in the scan in the variables **Column_name.1** to **Column_name.4** and **Column_value.1** to **Column_value.4**.
- If a variable need not be assigned, it should be set to blank.
- Issue an XEDIT **file**.

5.7.2 Executing the scan parameter file in your own machine

- Issue the command **[EXEC] UDBARBAT <yourname> UDBARBAT A** to start the archive tape scan.
- If an execution time was specified, UDBARBAT will pause until that time.
- Unless you are using a tape manager, messages will be issued to attach a tape drive and mount the archive tape(s).
- The output of the scan is stored on your A-disk in a file named **AFyymmdd SLhhmmss**.
- To inspect the scan results, issue a CMS FILELIST and type **UDBAR** on the corresponding filelist line.

5.7.3 Submitting the scan parameter file to a scan server

- To have your scan parameter file processed by the scan server, forward the scan parameter file to the server, using a CMS SENDFILE command.
- The server returns the scan results as a virtual reader file named **AFyymmdd SLhhmmss**. Issue a CMS RECEIVE.
- To inspect the scan results, issue a CMS FILELIST and type **UDBAR** on the filelist line for **AFyymmdd SLhhmmss**.

How to setup an archive scan server is explained in the section Initiating an audit archive scan server earlier in this manual.

6 UDBAR Archiving

6.1 Audit Archiving

Archiving the disk-resident audit log dataset **UDBAR AUDIT1** is done by appending the dataset to the current audit archive tape. Depending on the expected volume of audit data generated and the characteristics of the tape devices used, the installation may choose to keep an archive tape per day, per week, per month or per year. The archive processor detects the begin of a new archiving period and will request a new archive volume during tape mount. In the course of an archiving period, the mount message will display the tape label of the current archive.

The archive processor uses the archiving period, as defined in the UDBAR OPTIONS dataset, to build the archive tape label, as follows:

Archive_cycle	**Tape label generated**
DAY	UDBAR-yyyy-mm-dd
WEEK	UDBAR-yyyy-mm-Wn (where n is a value between 1 and 5)
MONTH	UDBAR-yyyy-mm
YEAR	UDBAR-yyyy

The archive processor uses FILEDEF and LABELDEF commands to pass the label to the operating system, or to a tape manager, if present. The FILEDEF VOLID operand is not used (the volume label is not checked) and the FILEDEF DISP is set to MOD when appending to the current archive volume.

6.2 Log Archive Procedure

The UDBAR Processor UDBARP performs automatic archiving of the UDBAR log file UDBAR AUDIT1 when the log is full or when its ARCHLIM specification has been reached.

An explicit log archive is scheduled using the command:

UDBARCMD ARCHIVE [write_password]

UDBARCMD prerequisites

- The UDBARP virtual machine should be logged off.
- The VM user requesting the archive should be able to link to the A-disk (or directory) of UDBARP in write mode. If the UDBARP minidisk is write protected, the write_password should be specified as an UDBARCMD argument.
- The UDBARUTX tape exit controls the mounting of the archive tape.

7 UDBARMIG Utility

The UDBARMIG utility can be used to convert the list of audited tables, as present in the SQLAF RULES file, into a number of AUDIT TABLE statements.

7.1 Syntax

UDBARMIG RULES <database>

The utility reads the SQLAF RULES for the TABLE entries following the named DATABASE statement and generates a CMS file named UDBARMIG OUT. This file contains, for every table previously audited by SQL/AF, the command:

AUDIT TABLE <creator>.<tablename> USING POLICY UDBARS

7.2 Operation

- Ensure that the UDBAR audit policy has been defined to DB2_LUW as follows:

  ```
  CREATE AUDIT POLICY UDBARS
        CATEGORIES EXECUTE WITH DATA
        STATUS BOTH
        ERROR TYPE NORMAL
  ```

- Link to the A-disk of the SQLAFP machine and copy the file **<database> TABAUDIT** to your A-disk.

- Issue **UDBARMIG RULES <database>**

- If required, modify UDBARMIG OUT using XEDIT.

- FTP UDBARMIG OUT to zLinux.

- Execute the file at the Linux command prompt as follows:
  db2 –f UDBARMIG.OUT

8 UDBAR User Exit

When the Audit Processor UDBARP finds the user exit program **UDBARUXP EXEC** on its file search chain, it will invoke that EXEC before a record is written to the audit log. The exit can access all fields of the audit record and take appropriate action. There is no restriction regarding the processing allowed in the user exit. However, since the user exit runs as an extension of the Audit Processor, care should be taken not to engage in long-running transactions or to produce long wait states of any kind. For performance reasons, UDBARP initialization loads the UDBARUXP EXEC in storage and subsequently invokes the storage resident EXEC. This implies that changes to the disk-resident UDBARUXP EXEC will take effect at the next restart of UDBARP.

On entry, the UDBARUXP user exit receives the audit logrecord fields as invocation arguments on the ARG statement. A sample exit is provided with the product as UDBARUXP SAMPLE. The sample defines the invocation arguments on an ARG statement. User code may be inserted following that initial statement.

On exit from UDBARUXP, returncode 0 will write the audit log record. To bypass writing the log record, specify returncode 8.

Logrecord fields passed to UDBARUXP

LDATE	the current date in DB2 format yyyy-mm-dd
LTIME	the current time as hh:mm:ss
LSQLID	the DB2 username performing the statement
LVMID	the client username performing the statement
LDBN	the current databasename
LTABCR	the creator of the table affected by the audited statement
LTABN	the name of the table affected by the audited statement
LPROGC	the creator of the package containing the audited statement
LPROGN	the name of the package containing the audited statement
LSECT	the package section number of the audited statement
LAGENT	the DB2 agent number performing the audited statement
LLUWID	the DB2 LUW number for the audited statement
LROWS	the number of rows processed
LSQLCODE	the SQLCODE resulting from the audited statement
LTERMID	the "Related" information (depending on the client's DB2 platform)
LCMND	the type of the audited statement (SELECT, INSERT, UPDATE, DELETE, COMMIT, ROLLBACK)
LTEXT	text of the audited statement

9 UDBARUTX tape exit

When your installation uses tape managing software, the Archive process and the LogScan program may be requested to call the user exit **UDBARUTX**, for tape mounting and dismounting control.

A dummy UDBARUTX is distributed with filetype **EXECS**. To activate the exit, change its filetype to **EXEC** and insert your coding in the EXEC as described in the following section.

Please note that following sample EXECs are also distributed with UDBAR:

- UDBARUTX DYNAMT: sample exit for use with DYNAM/T of Computer Associates
- UDBARUTX VMTAPE: sample exit for use with VMTAPE

If you are using one of these tape managers, you may find the sample useful and modify it to suit the requirements of your installation.

At the start of the archive, UDBARUTX is called with function code **Q_EXTEND** to determine whether your tape manager allows tape extend via the FILEDEF DISP MOD operand. (Tape extend occurs when audit log entries must be written to the archive tape of the current archive period.) If your tape manager allows tape extend, code the statement **returncode = 1** in the **query_extend** routine of UDBARUTX. If not, leave **returncode = 0** as set at the beginning of UDBARUTX.

9.1 User exit calls during archive (with extend enabled)

When you did enable tape extend, following UDBARUTX calls will be made during archive:

UDBARUTX CREATE <tape_label>

The CREATE function is called when a tape must be allocated for a new archive period. The exit should request allocation of a scratch tape labelled <*tape-label*>. The tape must be mounted on TAP1.

UDBARUTX OPEN <tape_label>

The OPEN function is called when the tape for the current archive period must be mounted. The exit should request mounting of the tape labelled <*tape-label*> as device TAP1.

UDBARUTX CLOSE <tape_label>

The CLOSE function is called at the end of the archive. The exit should request dismounting of the tape labelled <*tape-label*>.

9.2 User exit calls during archive (with extend disabled)

When tape extend has been disabled, UDBARUTX will subsequently be called as follows:

UDBARUTX CREATE <tape_label>

The CREATE function is called when a tape must be allocated for a new archive period. The exit should request allocation of a scratch tape labelled <*tape-label*>. The tape must be mounted on TAP1.

UDBARUTX OPEN <tape_label>
UDBARUTX OPEN_AUX <tape_label>

The OPEN function is called when the tape for the current archive period must be mounted. The exit should request mounting of the tape labelled <*tape-label*> as device TAP1. The OPEN_AUX function requests allocation of a second tape as device TAP2. The archive processor will copy the contents of the primary tape (TAP1) to the auxiliary tape (TAP2) and append the disk audit log to the auxiliary tape. Typically, the primary and secondary tape files will be members of a generation group.

UDBARUTX CLOSE <tape_label>
UDBARUTX CLOSE_AUX <tape_label>

The CLOSE and CLOSE_AUX functions are called at the end of the archive. The exit should request dismounting of the primary and auxiliary tapes labelled <*tape-label*>.

9.3 User exit calls during Log Scan

The Logscan utility can be requested to process an audit archive tape. If the UDBARUTX EXEC is found, the exit will be called during logscan as follows:

UDBARUTX OPEN <tape_label>

The OPEN function is called at the begin of Logscan to mount the tape label specified by the logscan user. The exit should request mounting of the tape labelled <*tape-label*> as device TAP1.

UDBARUTX CLOSE <tape_label>

The CLOSE function is called at the end of the logscan. The exit should request dismounting of the tape labelled <*tape-label*>.

Note:

If the tape exit executes satisfactorily for archiving, it will also work correctly for Log scanning.

10 UDBAR Logrecord Layout

The layout of the logrecord on both the disk log and the log archive tapes is as follows:

CHAR(8)	Record timestamp (hardware clock value)
CHAR(1)	Flag = '**P'**
CHAR(10)	Date of auditing in DB2 format yyyy-mm-dd
CHAR(8)	Time of auditing in DB2 format hh:mm:ss
CHAR(8)	DB2 user name submitting the statement
CHAR(8)	Client name submitting the statement
CHAR(8)	Database containing the audited table
CHAR(8)	Creator of the audited table
CHAR(18)	Name of the audited table
CHAR(8)	Creator of the package
CHAR(8)	Name of the package
SMALLINT	Package section number of the statement
SMALLINT	Reserved
INTEGER	DB2 LUW_ID
INTEGER	SQLCODE
INTEGER	Number of rows processed by the statement
CHAR(8)	Statement type
CHAR(8)	Related information (depending on the client's DB2 platform)
SMALLINT	Reserved
SMALLINT	Length of the statement text
VCHAR(8192)	Statement text (varying length)

11 UDBAR Messages

11.1 Messages issued by the UDBAR Processor

UDBARP001 UDBAR Writer loaded at xxxxxxxx

Informatory message showing the load address of the UDBARP program.

UDBARP002 UDB Audit Log DBxxxxxx AUxxxxxx processed and erased

Informatory message indicating that the named audit log file has been processed successfully.

UDBARP007 Invalid startup argument xxx

An invalid argument was specified when starting UDBARP from the profile exec.

UDBARP008 ARCHLIM invocation syntax is in error

The ARCHLIM value is missing or not numerical.

UDBARP018 CMS rc xxx NUCXloading UDBARUXP

A CMS error occurred when loading the UDBAR user exit. Refer to the CMS documentation for an explanation of rc xxx.

UDBARP051 Audit Log is full

The file UDBAR AUDIT1 is full. The Audit Processor will utomatically initiate an archive and continue processing when archiving terminates.

UDBARP053 Auto-archive has been initiated

Informatory message shown after UDBARP051.

UDBARP054 Archive limit reached

<ARCHLIM> audit records have been written to UDBAR AUDIT1. An automatic archive is initiated.

UDBARP997 UDBAR software license has expired

Self-explanatory. Request a new software key if appropriate.

UDBARP999 Audit Processor shutdown completed

The audit processor has completed its auto-logoff procedure.

11.2 Messages issued by the UDBAR LogScan program

UDBARIR001 OPEN error on archive tape dataset

An open error occurred when scanning an UDBAR archive tape. Ensure that the correct tape has been mounted.

UDBARIR003 RC xxx reading UDBAR AUDIT1

An error occurred when reading the UDBAR Audit Log. RC xxx can be found in the documentation of the CMS macro FSREAD.

UDBARIR002 RC xxx appending to $$LIST

An error occurred when appending selected audit records to the internal display list. This is most likely a storage problem. Increase the size of you virtual storage.

UDBARIR004 RC xxx writing save file

A request to save the logscan criteria to file failed. RC xxx can be found in the documentation of the CMS macro FSWRITE.

11.3 Messages issued by the UDBARCMD program

UDBARC103 HNDIUCV RC is xxx

The UDBARCMD invocation syntax is invalid.

11.4 Messages issued by the UDBAR User Exit facility

UDBARUXP01 RC xxx after EXECCOMM

Unexpected error during initiation of the User Exit. Contact Software Support.

UDBARUXP02 SHVRET xx after EXECCOMM

Unexpected error during initiation of the User Exit. Contact Software Support.