DBARS

Db2
Access
Recording
Services

Version 4.1

---

1. Product Description

Many corporations maintain sensitive or confidential data that are vital to corporate operations. In today's distributed IT environments, these data need special protection. As the initial security measure, Db2 privileges are granted to authorized users who are allowed to access these data. As a further measure, all access to sensitive data should be recorded, to verify that accesses are made for legitimate reasons and not for fun, curiosity or dishonest motives.

Today, there is a growing concern (and legislation) about data privacy. Databases often contain confidential information about individuals. This information has been entrusted by the individual to the corporation, in the expectation that the information will be handled with respect to personal privacy. Legislations may state that companies owning personal information, implement procedures to ensure that personal data are accessed by legitimate sources only. Legislation may also impose that accesses to confidential information can be retraced whenever needed.

Once Db2 privileges have been granted to a user, the user has access to the entire table. For reasons of security and privacy, it is necessary to control data access at the table-row level. It is therefore advisable to maintain an access log, either by application programming or by means of a system tool. The access log should record which data have been accessed, by whom and when. The log will be a valuable source of information for auditors or security officers.

1.1   Db2 Auditing Facilities

Sensitive data are normally stored in Db2 tables created with the AUDIT clause of the CREATE TABLE statement. A Db2 audit trace can be activated to verify that only authorized users access auditable tables and to record the SQL statements that access these tables. Finally, Db2 log analysis can be used to view the actual modifications to the data.

However, these Db2 facilities are not sufficient to fully record all access to sensitive data, for the following reasons:

- The audit trace records only the first read or write SQL statement in a logical unit of work.

- The trace records do not provide the contents of the input variables submitted with the SQL statement. Without this information, access recording is incomplete.

- Db2 auditing requires that an auditing trace be enabled. If many tables are audited or intensely used, the operating cost of tracing may be excessive.

1.2 Facilities provided by DBARS

Db2 Access Recording Services - "**DBARS**" - is a product developed by Software Product Research.

- DBARS records all accesses to sensitive data in auditable Db2 tables, by recording the SQL statements that perform the access. Depending on the degree of auditing defined, read (SELECT) and write (DELETE, INSERT and UPDATE) access will be recorded.
- DBARS will record access to:
  - the Db2 tables specified in the AUDITNAMES section of the DBARS startup parameters
  - **all** Db2 tables when the ALLTABLES option is specified at DBARS startup
  - when none of the above is specified, the Db2 tables created with the AUDIT CHANGE or AUDIT ALL attribute
- DBARS records all input variables ("hostvariables") associated with the SQL statement.
- DBARS stores the intercepted SQL statement into the DBARS **Recorder**.
- DBARS provides a powerful utility to scan the Recorder for recorded events.
- DBARS has its own interface to Db2 and does not depend on Db2 tracing. As a result, recording overhead will be acceptable.
- DBARS provides following additional functions, when requested by the installation:

  - Issues alerts for questionable SQL statements
  - Blocks Db2 access for such SQL statements
  - Integrates with external security systems, using FTP or HTTP.

1.3 The Recorder

For each access to an audited table, DBARS inserts following data into the DBARS **Recorder**:

- The **context** of statement execution:
    - the date and time of access
    - the name of the Db2 application server accessed
    - the name of the Db2 application requester (identical to the application server for local access)
    - the Db2 and z/OS userid
    - the correlation ID (e.g. the z/OS job name for batch access)
    - the type of connection (e.g. batch, CICS)
    - the type of access (dynamic or static)
    - the LUW_id
    - the external application name for distributed access
    - the workstation name for distributed access
    - the name of the program used for access
    - the number of rows modified by the statement
    - the statement's SQLCODE (indicating successful or failing access)

- The **text of the SQL statement** executed, with all input variables ("host variables") in the statement replaced by their contents. [1]

At the choice of the user, the Recorder is a Db2 table, a VSAM ESDS cluster or a sequential file (BSAM dataset).

Recorder Considerations

- When defining a VSAM ESDS Recorder during product installation, the supplied jobstream will also define Recorder Index clusters. These KSDS clusters are provided to allow faster scan of the Recorder. The Recorder Indexes are on date and time. A Recorder Index record is written when a predefined number of records have been written on the Recorder.
- A VSAM or a BSAM Recorder can be "exported" to a Db2 Recorder using the archive program. If exporting occurs frequently, the advantages of both Recorder options can be combined. For more details, read section 11.8.
- The sequential file option provides the best performance and flexibility. The option should be chosen when intensive access recording is performed.

---

[1] When VARCHAR variables are replaced, the length of the replacement is limited to 128 characters. Should the application specify an invalid length for the VARCHAR, the variable will be displayed as ???.

1.4   Inspecting the Recorder

The **Recorder Scan** utility searches the Recorder for specific access events. The user may supply following search criteria:

- One or more columns of the Recorder.
- Table column names used in the recorded SQL statement. This will report all SQL statements that reference the named table column.
- Table column names with a specified value. This will report all statements that reference the table column with the specified value. This option will report all recorded access for a given table "key".

The Recorder Scan program includes following additional facilities:

- The search values specified for the recorder table columns may contain Db2 expressions. For example: **Program Name LIKE DSQ%**
- The program can be used to scan the archive result table, described in the paragraph "Archiving the Recorder".
- The search arguments supplied by the user are saved in a Db2 table and may be recalled in a later scan session. The USERPARMS table saves up to 32 sets of search arguments.

1.5   Access Summary Report

The Access Summary program accepts the same search criteria as the Recorder scan. The program provides an hierarchical view of recorded access events.

- Typically, access summary is requested to display, for a given table, the **list of all users** that have accessed the table in a defined period of time.
- For a designated user in the user list, a **program list** may be requested, to list all programs by which that user has accessed the table.
- In the program list, a **statement list** may be requested for a selected program. The list will show the program's SQL statements, executed by the user against the Db2 table.

1.6   Archiving the Recorder

The DBARS archiving function transfers the Recorder to a sequential disk or tape dataset, so that recorded information can be kept for a longer period of time.

DBARS supplies a utility to scan an archived Recorder using the search criteria, described above. At the choice of the user, the archive scan produces a printed report or a result Db2 table. The latter table can be used as input for an interactive Recorder scan.

1.7   Customizing DBARS

- When the Recorder has been defined as a Db2 table, it can be processed as such by customer procedures. When the Recorder is a VSAM cluster or a sequential file, the Recorder scan can be used to transfer the Recorder entries to a Db2 table.

- An installation may provide a user exit that will be invoked whenever an access is stored in the Recorder. The exit is written in REXX and receives all the Recorder columns as its input arguments.

1.8   Access Exception Reporting

The Access Exception Reporting facility is designed for implementation within a DBARS User Exit, in order to detect and report unusual data access events. For more details see "Reporting Access Exceptions").

1.9   The DBARSGUI Facility

The facility allows a Windows workstation to scan the DBARS Recorder. DBARSGUI is a client-server application, where the server is implemented by the DBARSGUI program executing in a z/OS address space. The client application consists of a number of Java classes.

DBARSGUI is self-contained: it has no special software prerequisites. For full details, please read the DBARSGUI chapter.

2 Installation

2.1 Software Prerequisites

DBARS prerequisites:

- z/OS Version 1.6 or later.
  z/OS Version 2.3 is supported.

- Db2 Server for z/OS Version 9 or later.
  Db2 Server for z/OS Version 12 is supported.

2.2 Define the DBARS userid's

The userid performing DBARS installation should have the necessary Db2 privileges:

- to create the DBARS database
- to create the DBARS tablespaces and tables
- to bind the DBARS plans

The userid that will execute the DBARS access recording function should have the Db2 privileges:

- to access the system catalogs
- to write to the DBARS tables
- to execute the DBARS plans

The userid that executes the DBARS recorder scan functions should have the privilege to dynamically access the DBARS tables.

The DBARS runtime requirements are met when the userid running the DBARS plans has SYSADM authority. If it is impossible to grant SYSADM, explicit grants should be issued, as explained later in this section.

2.3 Define the DBARS Libraries

Before starting the installation process, four DBARS libraries should be created, with the characteristics listed in the table below.

Library Name	Record Format and Length	Block Size	Primary Extent (3390 tracks)	Secondary Extent (3390 tracks)	Directory Blocks
DBARS.OBJLIB	FB 80	8000	1200	200	10
DBARS.DBRMLIB	FB 80	8000	250	20	4
DBARS.JCLLIB	FB 80	8000	10	2	2
DBARS.LOADLIB	U 0	by system	5000	500	10

If you wish to use DBARS library names other than the default ones, specified above, the next sections show how to do so.

Please note that DBARS.LOADLIB **MUST** be defined in the active PARMLIB PROGxx as an **authorized library**.

2.4   Uploading DBARS using FTP

The DBARS product is electronically delivered as a ZIP file, which should be processed as follows:

- Unzip the file to a PC directory, for example C:\DBARS.
- Go to the newly created DBARS directory.
- If non-default DBARS library names are used, edit and change the library names DBARS.OBJLIB, DBARS.JCLLIB and DBARS.DBRMLIB in the DBARSFTP.SKT file.
- Execute the DBARSFTP.EXE program.
- DBARSFTP will prompt you for:
  - the name of the z/OS FTP server
  - your FTP userid
  - your FTP password
- DBARSFTP will then transfer the DBARS software from the PC directory to following z/OS libraries:
  - DBARS.OBJLIB
  - DBARS.DBRMLIB
  - DBARS.JCLLIB
- When errors have occurred during the FTP session, you will get an error message. Examine the file DBARSFTP.LOG in the DBARS directory to determine the nature of the error.

2.5   Installing DBARS

Following jobstreams should be submitted from the DBARS JCLLIB to z/OS **in the specified order**:

- DBARSLNK
- links the DBARS modules into the DBARS.LOADLIB
- DBARSTAB
- creates the DBARS tables
- DBARSXT1
- creates the DBARS EXCEPTIONS table
- DBARSBND
- binds the DBARS plans
- DBARSGRN
- grant execution authorities
- DBARSREC
- defines the Recorder as a VSAM cluster
- DBARSCSR
- defines the Recorder as a BSAM dataset

When the recorder will be a Db2 table:

- Submit the DBARSTAB job.
- Submit the DBARSXT1 job.
- Do not submit DBARSREC or DBARSCSR.

When the recorder will be a VSAM cluster, submit the DBARSREC job. (The DBARSREC job will also define the Recorder Index clusters. These KSDS clusters are provided to allow faster scan of the Recorder.)

When the recorder will be a BSAM dataset, submit the DBARSCSR job.

The DBARSLNK, DBARSBND, DBARSTAB and DBARSXT1 jobs must be submitted in all cases. If the Recorder is not a Db2 table, DBARSTAB may be submitted with minimal space allocations. If the Access Exception facility will not be used, DBARSXT1 may be submitted with minimal space allocations.

If the userid running DBARS components does not have SYSADM authority, the DBARSGRN job should be submitted as well.

Before submitting, edit and verify each jobstream as follows:

DBARSLNK

Must be executed in each z/OS system that will run DBARS.

- Complete the JOB statement.
- Define the Db2 load library on the SET DB2LIB statement that appears at the beginning of DBARSLNK.
- If library names other than the DBARS defaults are used, modify DBARS.OBJLIB and DBARS.LOADLIB to your names.

DBARSTAB

Must be executed in each Db2 system. If a Db2 Recorder is not used in that system, create the Db2 tables with minimal space allocations.

- Complete the JOB statement.
- Specify the target Db2 system in the DSNUTILB PARM operand.
- Complete the SDSNEXIT and SDSNLOAD DD's for the STEPLIBs.
- Verify the following Db2 statements and modify them to meet your installation requirements and standards:
  - CREATE DATABASE DBARS
  - CREATE TABLESPACE DBARS
  - CREATE TABLESPACE DBARSAUX
- The BUFFERPOOL BP32K implicitly defines the maximum length of the SQL statement stored in the DBARS RECORDER table. If the DBARS and DBARSAUX tablespaces are created in other bufferpools, DBARS will automatically adapt the maximum SQL statement length. It is recommended that a 32K buffer pool be used.
- The DBARS AUX_RECORDER is used for recovery purposes only. When errors (such as no available space) prevent DBARS from writing to the RECORDER table, the AUX_RECORDER table will be used automatically.

DBARSBND

Must be executed in each z/OS system that will run DBARS.

- Complete the JOB statement.
- Define the Db2 load libraries on the SET DB2EXIT and DB2LOAD statements that appear at the beginning of DBARSBND.
- Update the DSN SYSTEM(xxxx) statement that follows the SYSTSIN DD.
- If a library name other than the DBARS default is used, modify DBARS.DBRMLIB to your library name.

Note

A warning will be issued when binding the DBARSW component. The warning should be ignored: TEMP_TAB is a temporary table created and dropped during DBARS initialization.

DBARSGRN

Must be executed in each z/OS system that will run DBARS, when the DBARS userid has no SYSADM authority.

- Complete the JOB statement.
- Specify the Db2 subsystem on the DSNUTILB PARM operand.
- Specify the Db2 load libraries on the STEPLIB DD statement.
- Change the grantee name on the GRANT statements.

DBARSXT1

Must be executed in each Db2 system that will hold the DBARS EXCEPTIONS table.

- Complete the JOB statement.
- Specify the target Db2 system in the DSNUTILB PARM operand.
- Complete the SDSNEXIT and SDSNLOAD DD's for the STEPLIBs.
- Verify the following Db2 statement and modify them to meet your installation requirements and standards:
  CREATE TABLESPACE EXCEPT
- The BUFFERPOOL BP32K implicitly defines the maximum length of the SQL statement stored in the DBARS EXCEPTION table. If the tablespace is created in other bufferpools, DBARS will automatically adapt the maximum SQL statement length. It is recommended that a 32K buffer pool be used.

Note:

The DBARSXT1 job assumes that the DBARS database has been created by a previous DBARSTAB execution. If this is not the case, you should submit the CREATE DATABASE statement manually.

2.5.1 Define a VSAM Recorder

DBARSREC

When you plan to use a VSAM Recorder, you should submit DBARSREC.

- Complete the JOB statement.
- Specify the volume operand in the DEFINE CLUSTER statements.

2.5.2 Define a sequential (BSAM) Recorder

DBARSCSR

When you plan to use a BSAM Recorder, submit DBARSCSR.

- Complete the JOB statement and the DD statements for the Recorder.
- The BSAM recorders should be named *<prefix>*.RECORDER.S*<x>* where *<x>* equals A for the primary Recorder and B for the secondary Recorder. For *<prefix>* specify the same prefix in the DBARS startup parameter as shown in section SAM_PREFIX.
- If DBARS should store its audit records in XML format, specify PARM='XML' on the EXEC DBARSCSR statement and insert an OPT XML in the DBARS startup parameters. Please note that the XML option is intended only for transferring the Recorder to an external security system, such as the Oracle AUDIT Vault and Database Firewall©.
- Update the DD SPACE operand by changing the default (512,256) to the actual space requirement. In the distributed installation JCL, space is expressed in blocks of 27800 bytes. If preferred, space may also be expressed in tracks or cylinders.

> The number of audit records per block depends on the length of the recorded SQL statement. In the average, 50 audit records should fit in a Recorder block. A 3390 track will store 2 Recorder blocks. The same space should be allocated to the primary and secondary Recorders.

2.6   DBARS Post-installation steps

Some DBARS programs should be defined to TSO as authorized programs, by including them in the SYS1.PARMLIB IKJTSOxx member, under the AUTHPGM statement. The updated IKJTSOxx is activated by an IPL or using the PARMLIB command. Failure to insert these DBARS programs in IKJTSOxx will result in a 047 system abend when using them.

Following programs should be defined as authorized:

- **DBARSIR** (only when the SSID argument is used during its invocation)
- **DBARSRRI** because it submits a CLOSE_REC command to the DBARS address space
- **DBARSRXI** when the DBARS Access Exception facility is used

2.7   Db2 version or release migration

When a new Db2 version or release has been installed, the DBARS software should be rebound. The bind of the DBARS plans is achieved by submitting the DBARSBND JCL job, which is part of the DBARS distribution material.

3   Operating DBARS

3.1   Specifying the tables to be recorded

- By coding the names of the tables following the **AUDITNAMES** statement in the startup parameter dataset. **This is the preferred method for designating recordable tables**.

- By coding ALLTABLES in the startup parameters. This will record accesses to all Db2 tables. The ALLTABLES option should be used with caution, due to the possible processing overhead. The ALLTABLES option may be followed by a number of EXCLUDE statements.

- By creating or altering a table with the AUDIT ALL or AUDIT CHANGE clause. This method creates plan-to-table dependencies and may cause rebind when the audit table column is altered. DBARS uses the SYSTABLES.AUDIT column only to determine auditable tables. Db2 audit tracing is not used.

3.2   Recording a newly created table

When a new table should be recorded after creation, following rules apply:

- The RECORD_DDL start option should be in effect.

- When the CREATE TABLE statement specifies the AUDIT clause, the new table is automatically inserted in the DBARS table list.

- When the ALLTABLES option is in effect, the new table is automatically inserted in the DBARS table list.

- When AUDITNAMES is in effect, the new table is automatically inserted in the DBARS table list, if the new tablename meets a [generic] name specification in the AUDITNAMES table list.

The table list is automatically rebuilt at the start of a DBARS session.

3.3   Recording row fetching

When the COUNT_ROWS option is specified for a table in the startup AUDITNAMES section, DBARS will record the number of rows fetched from the table. The facility is intended to be used in conjunction with a DBARS user exit.

3.4   Coding DBARS Startup Parameters

By default, DBARS startup parameters are specified in the **EXECPARM** member of the **DBARS.JCLLIB** library.

If the startup parameters are stored elsewhere, specify the membername and eventually the libraryname on the PARM operand of the EXEC DBARS statement, as follows:

> PARM='membername'
> or:
> PARM='libraryname(membername)'
>
> In the PARM='membername' case, the libraryname defaults to DBARS.JCLLIB.

EXECPARM syntax rules:

- An EXECPARM statement should be coded on a single line and should not exceed 80 characters.
- Continuation lines are not allowed, as there is no need for.
- Most startup parameters have default values. If these defaults are acceptable, there is no need to specify the parameter at all.
- Comments are allowed in the EXECPARM dataset. A line starting with an asterisk is treated as a comment. Comments may also follow he statement parameters. In this case the comment should be enclosed by /* and */.

> For example: DB2SYS xxxx   /* Db2 subsystem to audit */

ALERT_PROC Datasetname(Membername)

If DBARS should invoke a user exit for an ALERTed access, specify the library and membername of the REXX program.

For details about writing DBARS user exits, see section "Writing an Alert Exit" below**.**

If ALERT_PROC and EXIT specified, both members should reside in the same library.

Default: No exit is invoked.

ALLTABLES [CHANGE]

Requests that DBARS records the accesses to all Db2 tables. When ALLTABLES CHANGE is specified, only update accesses will be recorded. Otherwise, all access, including SELECT, is recorded.

ALLTABLES may be followed by one or more EXCLUDE statements. The EXCLUDE statement is described later in this section.

When ALLTABLES is not specified, recording will be done only for the Db2 tables defined with the AUDIT CHANGE or AUDIT ALL attribute or for the tables specified after **AUDITNAMES**.**ARCHIVE_PROC procedurename**

When a SWITCH_REC command is supplied or when a no-more-space condition is detected for the Recorder, DBARS automatically switches to the Auxiliary Recorder.

If the ARCHIVE_PROC statement has been provided, the named procedure will be started before switching to the alternate recorder.

For more information, refer to Automated Recorder Archiving**.**

AUDITNAMES

This optional statement starts the list of tablenames to be recorded. It should be the last statement in the parameter dataset and be followed by a tablename list. See also Specifying the tables to be recorded.

Each line in the tablename list has the format:

owner.tablename [CHANGE] [COUNT_ROWS]

A generic tablename can be specified by appending a % sign to the end of the generic name. A generic owner specification is not allowed.

If the **CHANGE** keyword is specified, only write access will be recorded for the designated tables: SELECT statements will not be recorded. If CHANGE is omitted, all access, including SELECT, will be recorded.

If the **COUNT_ROWS** keyword is specified, DBARS will record the number of rows fetched from the table. The facility is intended to be used in conjunction with a DBARS user exit. For more details see Reporting Access Exceptions).

For example:

```
AUDITNAMES
DSN8810.EMP
CUSTOMER.STATS CHANGE
ACCOUNT.ACT%
```

DB2_ADDRSPACE nnnnnnnn

Specify **nnnnnnnn** as the name of the Db2 Database Address Space.

Default: <Db2_subsystemname>DBM1

DB2_SUBSYS nnnn

Specify **nnnn** as the name of the Db2 subsystem where DBARS should be installed.

Default: the subsystem name found in DSNHDECP.

DYNONLY

When this option is specified, only dynamic SQL is recorded.

Warning
This option should be specified only on systems where DBARS runs as an SQL recorder. If DBARS is used as a security product, the DYNONLY option should not be specified.

EXCLUDE owner.tablename

The EXCLUDE statement should be specified only when using the ALLTABLES option. It specified the owner and tablename(s) to be removed from the previous ALLTABLES list. A generic tablename specification is allowed by terminating the name with a % sign.

For example:

ALLTABLES
EXCLUDE SYSIBM.%
EXCLUDE Q.ORG

EXIT Datasetname(Membername)

If DBARS should invoke a user exit for every recorded access, specify the library and membername of the REXX program.

For details about writing DBARS user exits, see section Writing a DBARS User Exit**.**

If EXIT and ALERT_PROC specified, both members should reside in the same library.

Default: No exit is invoked.

EXITMOD modulename [parameters]

If DBARS should invoke a **compiled** user exit for every recorded access, specify the load module name of the exit. The load module should reside in a z/OS STEPLIB or JOBLIB.

Optionally, up to 8 parameter strings can be specified on the EXITMOD statement. The exit will receive the address of each parameter in its parameterlist.

For details about writing an EXITMOD, see chapter 14

Default: No exit is invoked.

IGNORE

The IGNORE statement allows to bypass access recording for defined Db2 users, programs or jobs.

The syntax of the IGNORE statement is as follows:

IGNORE [JOB nnn] [USER nnn] [PROGRAM nnn]

IGNORE JOB   ignore all accesses from the named z/OS job
IGNORE USER   ignore all accesses by the named Db2 user
IGNORE PROGRAM   ignore all accesses from the named Db2 program

Notes:

Names appearing on the IGNORE can be generic by providing a trailing %. For example: IGNORE JOB REORG%.

If multiple ignore criteria appear on the same IGNORE statement, access will be ignored only when all the criteria are true.

For example: IGNORE PROGRAM ADBMAIN USER DBADM will ignore accesses by the ADBMAIN program, only when that program is executed by Db2 userid DBADM.

INCLUDE

The INCLUDE statement defines the Db2 users, subject to access auditing. If no INCLUDE statements are specified, all Db2 users will be audited.

The syntax of the statement is as follows:
INCLUDE USER nnnnnnnn

Notes:

Names appearing on the INCLUDE statement can be generic by providing a trailing %. For example: INCLUDE USER ACCT%.

An unlimited number of INCLUDE statements can be supplied.

NOINDEX

When the DBARS Recorder is a VSAM ESDS, an indexing KSDS cluster is maintained by the DBARS Writer task. The purpose of this index is to speed up the DBARS Recorder Scan program. If an external reporting system is used instead of DBARS reporting, the NOINDEX statement will provide increased performance in the DBARS Writer

OPTION [NODDF] [NOMASK] [XML]

OPTION NODDF

Inhibits recording of SQL accesses from the DDF environment.

OPTION NOMASK

Turns the DBARS data masking facility off. By default, DBARS masks the data values in the recorded SQL statements.

OPTION XML

Requests that the DBARS Writer stores the audit records in XML format, for transmission to an external security system.

PROGCACHE n

For static SQL statements longer than 4K, DBARS must retrieve the statement text from SYSIBM.SYSSTMT. To avoid repetitive catalogue access for the same statement, a program cache is maintained for the most recently retrieved statements.

The PROGCACHE parameter specifies the maximum number of entries in the cache. If that value is exceeded, the oldest cache entries are removed. The PROGCACHE value specified should be a power of 2. If necessary, DBARS will round the specification to the next power of 2.

Default size of the program cache: 4096

Q2_SIZE n

Before being inserted into the Recorder table, intercepted SQL statements are inserted into a queue (Q2) owned by the DBARS address space and shared by the DBARS Initiator in the Db2 DBM1 address space.

Q2 is a 64-bit object, residing above the 2 Gigabyte bar. The queue consists of a number of chained 128 K pages, whose total size equals Q2_SIZE. Q2 pages are not accessed until data is stored into them. Hence, there is no performance penalty in requesting a large Q2, if the pages in the higher part of the queue are never used.

Specify the size of Q2 in Megabytes.
Default: 1024 (which results in an 1 GB queue)

RECORD [DDL] [BIND] [UTIL] [CMND]

RECORD DDL
Requests DBARS to record DDL requests submitted to Db2. When omitted, DBARS will record DML requests only.

RECORD BIND

Requests DBARS to record BIND requests submitted to Db2.

RECORD UTIL

Requests DBARS to record Db2 utility runs.

RECORD CMND

Requests DBARS to record submitted Db2 commands.

Notes:

RECORD BIND, RECORD UTIL and RECORD CMND require the MONITOR1 privilege for the userid executing the DBARS program in the DBARS address space, since these requests are recorded using an IFI trace.

RECORDER_LOC nnnn

If DBARS is active in multiple Db2 systems, the DBARS Writer can be requested to store the intercepted SQL statements into a centralized Recorder, if the latter is a Db2 table.

The RECORDER_LOC statement provides the name of the Db2 system that holds the centralized Recorder.

Default: current Db2 system

For more information about Recorder sharing (Db2 and VSAM), please refer to **Shared Recorder Considerations.** A sequential Recorder cannot be shared.

RULES membername

If DBARS should block Db2 access requests, specify the name of the member that contains the rules for blocking SQL statements. The member should be in the same z/OS library as the DBARS startup member.

For details about DBARS blocking, please read the Chapter 4 .

SAM_PREFIX nnnnnnnn

Specify SAM_PREFIX if sequential (BSAM) datasets should be used for the DBARS Recorders. Replace nnnnnnnn with one or more dataset qualifiers of your choice. The length of SAM_PREFIX should not exceed 32.

The name of the BSAM Recorder will start with SAM_PREFIX, followed by 'RECORDER.SA' for the primary Recorder and 'RECORDER.SB' for the secondary Recorder.

> With SAM_PREFIX DBARS the Recorders will be named DBARS.RECORDER.SA and DBARS.RECORDER.SB.

> With SAM_PREFIX TEST.DBARS the Recorders will be named TEST.DBARS.RECORDER.SA and TEST.DBARS.RECORDER.SB.

SCAN n

The DBARS main task polls Q2 for new recordings at a defined time interval. The SCAN parameter specifies this interval in 1/100 seconds.

Default = 100 (1 second)

SEQREC

Specify SEQREC only when access recording should be done to a VSAM Recorder, which should have been defined previously using the DBARSREC job.

VSAM_SHARED

Specify this option when using a VSAM Recorder defined with SHAREOPTIONS(4,3).

3.5   Initiating DBARS as a started task

- The sample JCL (member DBARS.JCLLIB.DBARS) can be a JOB (located in a member of a data set defined in the IEFJOBS or IEFPDSI concatenation of master JCL) or a procedure (located in a subsystem procedure library, for example, SYS1.PROCLIB).

- The REGION parameter on the JOB or EXEC statement should be coded as REGION=0M.

- Please note that the RECORDA, RECORDAX,RECORDB and RECORDBX DD statements present in member DBARS.JCLLIB.DBARS are not needed when the Recorder has been defined as a Db2 table or as a sequential dataset.

- Issue the RACF commands to assign privileges to the started task:

    RDEFINE STARTED DBARS.DBARS STDATA(USER(xxxx) GROUP(yyy))
    SETROPTS RACLIST(STARTED) REFRESH

- Issue a START DBARS or insert the START command into the z/OS startup procedure.

3.6   Initiating DBARS using JCL

Submit the source JCL (member DBARSJ in DBARS.JCLLIB) to JES. Please note that the RECORDA, RECORDAX,RECORDB and RECORDBX DD statements are not needed when the Recorder has been defined as a Db2 table or as a sequential dataset.

The REGION parameter on the JOB or EXEC statement should be coded as REGION=0M.

3.7   Implicit connection considerations

DBARS does not issue explicit connect but assumes that its first SQL statement will implicitly connect to the intended Db2 subsystem. Implicit connections use the default subsystem name specified in the module DSNHDECP. If the default DSNHDECP is not pointing to the desired Db2 subsystem, specify the correct library in a STEPLIB or JOBLIB concatenation.

3.8   Sample DBARS start JCL

```
//MAIN      EXEC PGM=DBARS,MEMLIMIT=1152M
//STEPLIB   DD DSN=DBARS.LOADLIB,DISP=SHR
//RECORDA   DD DSN=DBARS.RECORDER.A,DISP=SHR
//RECORDAX  DD DSN=DBARS.RECORDER.A.INDEX,DISP=SHR
//RECORDB   DD DSN=DBARS.RECORDER.B,DISP=SHR
//RECORDBX  DD DSN=DBARS.RECORDER.B.INDEX,DISP=SHR
//SYSPRINT  DD SYSOUT=A,SPIN=UNALLOC,FREE=CLOSE
//SYSUDUMP  DD SYSOUT=A
//SYSTSPRT  DD SYSOUT=A
```

Please note that the RECORDA, RECORDAX,RECORDB and RECORDBX DD statements are not needed when the Recorder has been defined as a Db2 table.

3.9   Terminating DBARS

On the z/OS console enter the command **MODIFY DBARS,STOP.** This command can also be inserted in the z/OS shutdown procedure to stop the DBARS address space.

If Db2 command recording has been enabled, using the **RECORD** statement of the DBARS startup member, DBARS will automatically stop when Db2 terminates.

3.10 Operational Considerations

- The DBARS components:

  - The **master task** (DBARS): conducts the DBARS initiation and termination processes and provides an operator command interface.

  - The **Db2 interface task** (DBARSI): intercepts and queues all SQL statements executed into a storage queue. DBARSI executes in the Db2 DBM1 address space.

  - The **writer task** (DBARSW): inserts the SQL statements queued from the storage queue into the Recorder table or cluster.

  The intercepted SQL statements are transmitted through the different layers of the DBARS software using the above mentioned queue (Q2):

  - The DBARSI component stores the intercepted SQL statements into Q2, which is a 64-bit object owned by the DBARS address space.

  - The DBARS Writer stores the Q2 records into the Recorder. The Writer executes in the DBARS address space.

- The execution priority assigned to the DBARS address space should allow fast retrieval of the buffers built by DBARSI in Q2. This is especially true when a large number of SQL requests will be recorded.

- Buffers acquired by DBARS have storage key 7. This protects DBARS data from access by application processes, which execute in key 8.

- Initiating DBARS as a started task during z/OS IPL is the recommended method for starting DBARS, as it ensures that all accesses to audited tables will be recorded. When started before Db2 is active, DBARS will wait until Db2 is up.

- DBARS must be initiated in all Db2 subsystems where access to audited tables should be recorded.

- DBARS records all SELECT INTO, INSERT, UPDATE and DELETE statements. For cursor-based SELECT, only the OPEN of the cursor is recorded.

- SQL statements issued by the Db2 utilities are not recorded. However, SQL statements executed be means of the DSNUTIL EXEC SQL facility are recorded.

Recorder Considerations

When the Recorder has been defined as a Db2 table, the following applies:

> The DBARS Writer inserts recorded SQL statements into the primary RECORDER table. When a recoverable error, such as insufficient space, is detected, diagnostic information is written to the console and an automatic switch to the auxiliary recorder table AUX_RECORDER, is performed. This allows to clear the error situation on the primary recorder, at which time the primary recorder should be re-reactivated using the SWITCH_REC command, described in the DBARS Commands section of this manual.

When the Recorder has been defined as a VSAM cluster or a sequential dataset, the following applies:

> Two Recorders have been defined at product installation (RECORDER.A and RECORDER.B). When one of the Recorders has no more space, an automatic switch is done to the alternate Recorder. Switching to another recorder for writing will automatically delete its previous contents.

> The Recorder switching described above can also be forced using the SWITCH_REC command. In any case, after Recorder switching has been performed, the inactive Recorder should be archived. Automated archiving may be used for this purpose. (See Automated Recorder Archiving) If the inactive Recorder is not archived, its contents will be lost when performing the next switch.

3.11 Recorder Archiving Considerations

When a non-automatic archive is initiated, access recording switches to the alternate Recorder cluster, while the primary cluster is being archived. Archiving will not interrupt the access recording process and accesses continue to be inserted while archiving progresses.

3.11.1 Automated Recorder Archiving

When Recorder switching occurs, a user provided archiving procedure will be invoked, if this procedure has been defined on the **ARCHIVE_PROC** startup statement. DBARS invokes the procedure using the following z/OS command: **START *procedurename*,RECID='x'**

The RECID argument indicates to the called procedure which Recorder must be archived. RECID='A' indicates the primary Recorder, RECID='B' indicates the auxiliary Recorder. The called procedure should use the RECID argument on the DBARSA statement, followed by the procedure variable &RECID, as shown in the example below.

3.11.2 Automated Recorder Archiving Sample

```
//DBARSA JOB (xxxx),'...................
//DBARSA EXEC PGM=DBARSA,PARM='SEQREC RECID &RECID'
//ARCHIVE DD DSN=DBARS.ARCHIVE,DISP=(MOD),UNIT=SYSDA
//RECORDA DD DSN=DBARS.RECORDER.A,DISP=SHR
//RECORDB DD DSN=DBARS.RECORDER.B,DISP=SHR
//STEPLIB DD DSN=DBARS.LOADLIB,DISP=SHARE
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
```

3.12 **Shared Recorder Considerations**

If DBARS is initiated in multiple Db2 systems, the DBARS Writers can be requested to store the intercepted SQL statements into a centralized Recorder, that is, into the RECORDER of a named Db2 system. This is achieved by specifying the RECORDER_LOC startup parameter..

Sharing of the Recorder (either Db2 table or VSAM cluster) occurs in the DBARS Writer, at the time of writing to the Recorder.

If several Db2 systems must be recorded, each Db2 system must be equipped with a corresponding DBARS address space. The name of the target Db2 subsystem should be specified in the startup parameter DB2_ADDRSPACE of each DBARS. Each of these DBARS address spaces has its own Q2. Each DBARS also has its own writer task to store Q2 into the Recorder.

The Recorder should be setup by the customer as a shared Db2 table or VSAM cluster.

Scenario 1: Recorder as a Db2 table

Two Db2 systems Db2A and Db2B are recorded. The Recorder table is held in Db2B. Two DBARS address spaces are required: DBARS1 to record Db2A and DBARS2 for Db2B.

Db2A is captured by the DBARS1 address space with the following startup parms:

```
DB2_SUBSYS        DB2A
DB2_ADDRSPACE     DB2ADBM1
RECORDER_LOC      DB2B
```

Db2B is captured by the DBARS2 address space with the following startup parms:

```
DB2_SUBSYS        DB2B
DB2_ADDRSPACE     DB2BDBM1
```

In such a setup, there are 2 recorder writer tasks: one in DBARS1 and one in DBARS2. However, due to the RECORDER_LOC Db2B statement, the DBARS1 writer will connect to Db2B before performing the insert into the Recorder table. The DBARS2 writer will simply insert using its default connection.

The customer should ensure that a DRDA connection is available to DBARS1 for connecting and inserting into the recorder table maintained in Db2B.

Scenario 2: Recorder as a VSAM cluster

Two Db2 systems Db2A and Db2B are recorded. Two DBARS address spaces are required: DBARS1 to record Db2A and DBARS2 for Db2B.

Db2A is captured by the DBARS1 address space with the following startup parms:

      DB2_SUBSYS   DB2A
      DB2_ADDRSPACE   DB2ADBM1

Db2B is captured by the DBARS2 address space with the following startup parms:

      DB2_SUBSYS   DB2B
      DB2_ADDRSPACE   DB2BDBM1

The recorder is defined using the VSAM share options (4,3). This will allow to insert into the VSAM cluster from the 2 DBARS systems.

3.13   DBARS Storage requirements

3.13.1 Storage pool 241 requirements

Following DBARS control blocks reside in storage pool 241:

Buffer	Size
DBARS Common Area	4 K
Dynamic user list	64 K
Ignore list	Number of ignore items * 24 bytes

3.13.2 Storage requirements in the Db2 address space

- In Db2 versions 9 and 10, the DBARS Initiator requires 1MB of working storage in the Db2 address space.

- A 256K storage block is acquired for trace buffers.

- The size of the DBARS Initiator component is 24K.

3.13.3 Storage requirements in the DBARS address space

- The DBARS logic modules require less than 1 MB in the address space.

- Dynamic recording buffers are acquired above the "bar" in 64-bit storage. These buffers are owned by the DBARS address space and are shared with the DBARS component executing in the Db2DBM1 address space.

- The DBARS Q2 is owned by the DBARS address space and resides in 64-bit storage. Its size, in Megabytes, is specified by the Q2_SIZE startup parameter. The parameter defaults to 1024 MB.

- The suggested DBARS region size is 0M.

3.13.4 MEMLIMIT specification for DBARS startup

The MEMLIMT specification should provide for:

- The amount of 64-bit storage allocated to Q2 using the Q2_SIZE startup parameter.

- The size of the work segments required by the DBARS Initiator. The size of these segments should not be inferior to 64 MB. For busy systems, 128 MB is suggested.

- **The MEMLIMIT specification must be expressed in Megabytes.**

Example:

Given a Q2_SIZE of 4096 MB (4 GB) and a work segment size of 128 MB, MEMLIMIT=4224 should be specified.

Please note that the use of the above buffers is dynamic: non-used buffers will not compete for real storage.

3.14 DBARS Commands

DBARS commands are entered on the z/OS console using the **MODIFY** command (which can be abbreviated to **F**), for example:

MODIFY DBARS,STOP

F DBARS,STATS

Following commands are available:

AUDITTAB

Displays the list of Db2 tables that are currently recorded by DBARS.

EXECPARMS

Shows the contents of the active startup parameter dataset.

LICENSE

Displays information from the DBARS license key on the console.

RECSTATS

Shows statistical information about the DBARS Recorder table or cluster using messages DBARSM14I through DBARSM18I. See Messages issued by the DBARS Queue Manager for details.

REFRESH

Rebuilds the list of tables to be recorded by DBARS. Please note that this command may take a considerable time to complete. It may also cause contention on the Db2 catalogs.

Since the creation of Db2 objects is automatically intercepted by DBARS, use of this command should be exceptional. See "Recording a newly created table".

RELOAD_RULES

If DBARS blocking or alerting is active, the command will load the latest version of the rules from the library member.

RELOAD_UX

If a DBARS user exit is active and changes have been made to the exit coding, the RELOAD_UX command can be used to activate the new version of the exit. If the command is not used, the latest exit coding will be loaded during the next start of DBARS.

STATS

Shows the DBARS queuing statistics, using messages DBARSM03I through DBARSM10I. See Messages issued by the DBARS Queue Manager.

STOP

Terminates DBARS when all recorded access events have been written to the Recorder table.

SWITCH_REC

When the Recorder is a sequential dataset or a VSAM cluster, the command switches between the Recorder clusters A and B in a flip-flop manner. If cluster A is active, switching will be to cluster B and vice-versa.

When the Recorder is a Db2 table, the command switches between the DBARS.RECORDER and the DBARS.AUX_RECORDER tables.

Note that a Recorder full condition automatically switches Recorders.

When switching has occurred, the inactive Recorder should be archived. Automated archiving may be used for this purpose. See Automated Recorder Archiving.

The following commands are used for problem determination and should be entered only when requested by SPR software support.

- TRACE
- ITRACE
- PTRACE

4 Blocking Database Access

BLOCK statements are placed in a z/OS dataset whose name is specified on the DBARS startup statement RULES (see "RULES membername"). The dataset should be in the same z/OS library as the DBARS startup statement member.

Syntax

```
BLOCK    WHEN  <data_item> <operator> <data_value>
         [AND <data_item> <operator> <data_value>]
         …
         [AND <data_item> <operator> <data_value>]
```

Data_item

ACCESS	SELECT | INSERT | UPDATE | DELETE
CONNECTION	TSO | CICS | DDF | BATCH
DYNAMIC_SQL	true if SQL executed in dynamic mode
EXTAPPL	application name when remote access
EXTSERVER	server name (IP-address) if remote (DDF) access
	location name if local access
EXTWS	name of workstation when remote access
JOB	name of batch job or correlation
PROGRAM	name of application program
ROWS	number of rows updated, deleted or inserted
SERVER	name of Db2 subsystem
TABLE	creator.tablename of a table
	(generic tablenames not allowed)
TIME	time values must have the format hhmm
USER	Db2 userid

Operator

omitted	tests equal (greater or equal for data_item ROWS)
NOT	tests not equal
BETWEEN x AND y	tests within range

Data_value

The value to compare <data-item> with. Comparison is done over the specified length of data-value, thus allowing for a generic test.

The data-value **ANY** always evaluates the expression as true.

Additional Syntax Rules

- Alphanumeric values should not be enclosed in quotes.
- Rules that do not fit on one dataset line should be broken at the AND keyword.
- If multiple conditions are specified on a blocking line using the AND keyword, all conditions must be true for blocking to be performed.
- If OR-ed conditions are needed, they should be specified on separate BLOCK statements.
- Comment lines are allowed: such lines should start with an asterisk.

When an access has been blocked, the following message is written to the console:

DBARSI551A Access has been blocked.

The user receives a 60F z/OS abend and the access is recorded with an SQLCODE -90001.

Examples

During the night-shift, user N457 can use only program names beginning with NCB:

    BLOCK WHEN USER N457

    AND TIME BETWEEN 1800 AND 0800

    AND PROGRAM NOT NCB

Only users in the accounts receivable department can update the customer table using CICS:

    BLOCK WHEN TABLE NTR.CUSTOMER

    AND ACCESS NOT SELECT

    AND USER NOT ACR

    AND CONNECTION NOT CICS

The Employee table cannot be accessed from an IP-address:

    BLOCK WHEN TABLE DSN8A10.EMP

    AND EXTSERVER ANY

The Employee table cannot be accessed using dynamic SQL:

    BLOCK WHEN TABLE DSN8A10.EMP

    AND DYNAMIC_SQL

5   Alerting Database Access

ALERT statements are placed in a z/OS dataset whose name is specified on the DBARS startup statement RULES (see "RULES membername"). The dataset should be in the same z/OS library as the DBARS startup statement member.

Syntax

    ALERT   WHEN <data_item> <operator> <data_value>

                [AND <data_item> <operator> <data_value>]

                …

                [AND <data_item> <operator> <data_value>]

Data_item

ACCESS	SELECT | INSERT | UPDATE | DELETE
CONNECTION	TSO | CICS | DDF | BATCH
DYNAMIC_SQL	true if SQL executed in dynamic mode
EXTAPPL	application name when remote access
EXTSERVER	server name (IP-address) if remote (DDF) access
	name of database if local access
EXTWS	name of workstation when remote access
JOB	name of batch job or correlation
PROGRAM	name of application program
ROWS	number of rows updated, deleted or inserted
SERVER	name of Db2 subsystem
TABLE	creator.tablename of a table
	(generic tablenames not allowed)
TIME	time values must have the format hhmm
USER	Db2 userid

Operator

omitted	tests equal (greater or equal for data_item ROWS)
NOT	tests not equal
BETWEEN x AND y	tests within range

Data_value

The value to compare <data-item> with. Comparison is done over the specified length of data-value, thus allowing for a generic test

Additional Syntax Rules

- Alphanumeric values should not be enclosed in quotes.

- Rules that do not fit on one dataset line should be broken at the AND keyword.

- If multiple conditions are specified on a line using the AND keyword, all conditions must be true for alerting to be performed.

- If OR-ed conditions are needed, they should be specified on separate ALERT statements.

- Comment lines are allowed: such lines should start with an asterisk.

When an alert is issued for an access and an ALERT_PROC is not specified, the SQL statement is stored into the DBARS Exception table. When an ALERT_PROC is specified, the designated REXX procedure is invoked. In both cases the access is recorded with an SQLCODE -90002.

For details on alert exits, refer to the next section "Writing a DBARS alert exit".

Examples

Alert when during the night-shift, user N457 uses programs other than those beginning with NCB:

ALERT WHEN USER N457

AND TIME BETWEEN 1800 AND 0800

AND PROGRAM NOT NCB

Alert when a user updates the customer table and his userid does not start with ACR and connection is not via CICS:

ALERT WHEN TABLE NTR.CUSTOMER

AND ACCESS NOT SELECT

AND USER NOT CR

AND CONNECTION NOT CICS

Writing a DBARS ALERT exit

A DBARS alert exit is declared with an ALERT_PROC statement in the EXECPARM member of the DBARS JCLLIB. The user exit is written in REXX and called by the DBARS Writer task before the access is stored in the Recorder table.

The exit runs under the DBARS userid. At entry, there is a connection to the Db2 system that holds the DBARS Recorder table. The exit may perform any necessary action including SQL access. However, if SQL statements are issued, a COMMIT should be used before returning to DBARS.

Following arguments are passed to the exit:

Argument Number	Argument Content	Argument Length
1	Access timestamp	26
2	TCB address	8
3	Name of Db2 application server	16
4	Db2 userid	8
5	z/OS userid	8
6	Correlation name	12
7	Connection type	8
8	LUW_id	12
9	External Server Name	16
10	External Application Name	32
11	External Workstation Name	18
12	RDI Call Type	4
13	RDI Statement Type	4
14	Table Creator	8
15	Table Name	128
16	Program Name	8
17	Program Section number	4
18	SQLCODE	8
19	Number of rows processed	16
20	Operation Code	8
21	Dynamic access flag	1
22	Length of SQL Statement_text	8
23	Pointer to SQL Statement_text	8

6 Writing to the Windows Event log

6.1 Description

The DBARSLOG functions write DBARS exceptions to the Windows event log named "**DBARS Alert Log"**. DBARS exceptions are generated when a Db2 access is blocked (as described above) or when an alert is issued for an access (as described above).

The DBARSLOG component that executes on the mainframe transmits DBARS exception data to the DBARSLOG application that executes in Windows and that actually stores the exceptions to the Windows event log.

Data in the DBARS alert log are then available to software that exploits the Windows logs for processing (big data systems for example) .

A sample log entry is shown below.

6.2 Installing DBARSLOG

6.2.1 Installing on z/OS

The standard DBARS installation also installs the mainframe DBARSLOG component.

6.2.2 Installing on Windows

The DBARS distribution file contains a folder, named DBARSLOG. The entire folder should be moved to a Windows folder accessible to the process that will invoke DBARSLOG.

6.3 Configuring DBARSLOG

6.3.1 Configuring on z/OS

The DBARS distribution file contains in its JCL folder, the DBARSLOG SAMPJCL file, for starting the DBARSLOG server on z/OS:

EXEC PGM=DBARSLOG,PARM='PORT=...'

Specify the TCP/IP port to be used by DBARSLOG for communication with the DBARSLOG program in Windows. The default port number is 1946.

6.3.2 Configuring on Windows

The DBARS distribution folder DBARSLOG contains the DBARSLOG.BAT which starts the DBARSLOG application as follows:

DBARSLOG <IP_address> 1946

In the above command, replace:

- <IP_address> with the IP address of your z/OS system
- the default port 1946 (only if the default port was modified in the DBARSLOG PARM described above)

6.3.3 Starting the DBARSLOG server on z/OS

DBARSLOG can be started by submitting the customized DBARSLOG JCL.

Alternatively, DBARSLOG can be invoked as a started task. In this case, issue the following RACF commands to assign privileges to the started task:

```
RDEFINE STARTED DBARSLOG.DBARSLOG STDATA(USER(xxxx)
GROUP(yyy))SETROPTS RACLIST(STARTED) REFRESH
```

DBARSLOG issues TCP/IP Listen commands to wait for workstation requests. Note that there is no command provided to shutdown the DBARSLOG server. Use the z/OS cancel command instead.

6.3.4 Starting DBARSLOG on Windows

Start the DBARSLOG.BAT configured above.

Event Viewer

File   Action   View   Help

DBARS Alert Log   Number of events: 13

Level	Date and Time	Source	Event ID	Task Category
⚠ Warning	21/10/2014 10:43:42	DBARS Alert Log	0	None
⚠ Warning	21/10/2014 10:40:19	DBARS Alert Log	0	None
⚠ Warning	21/10/2014 10:40:19	DBARS Alert Log	0	None
⚠ Warning	21/10/2014 10:40:19	DBARS Alert Log	0	None
⚠ Warning	21/10/2014 10:39:33	DBARS Alert Log	0	None
⚠ Warning	21/10/2014 10:39:33	DBARS Alert Log	0	None
⚠ Warning	21/10/2014 10:39:33	DBARS Alert Log	0	None
⚠ Warning	21/10/2014 10:39:21	DBARS Alert Log	0	None
⚠ Warning	21/10/2014 10:39:21	DBARS Alert Log	0	None

Event 0, DBARS Alert Log   ✕

General | Details

Timestamp=2014-10-21 10.43.36.906763
Location=DALLAS9
Connection=TSO
User=IBMUSER
Application=DSQFFSQ7
Workstation=N/A
Exception=Blocked
SQL statement=SELECT * FROM SYSIBM . SYSTABLES FOR FETCH ONLY

Log Name:	DBARS Alert Log		
Source:	DBARS Alert Log	Logged:	21/10/2014 10:43:42
Event ID:	0	Task Category:	None
Level:	Warning	Keywords:	Classic
User:	N/A	Computer:	DESKTOP
OpCode:			
More Information:	Event Log Online Help		

7 Interactive Access Reporting

The online Access Reporting program DBARSIR is used to scan a Recorder that has been defined as a Db2 table. Alternatively, DBARSIR can be used to scan a result Db2 table produced by the other DBARS reporting programs.

DBARSIR is called from a TSO session using the TSO command

CALL 'DBARS.LOADLIB(DBARSIR)' ['SSID=xxxx']

or using an installation defined ISPF shortcut.

The SSID parameter on the above CALL statement can be used if the TSO session is not implicitly connected to the Db2 subsystem where the DBARS tables reside, that is, if the TSO login procedure does not contain a DD statement for the corresponding SDSNEXIT dataset.

DBARSIR can be invoked only if the TSO terminal has at least 32 lines.

7.1 Entering Report Criteria

At entry into the DBARSIR program, the report selection criteria are requested. Report criteria can be entered for the columns of the Recorder table and for expressions in the recorded SQL statement text.

7.1.1 Recorder Table Column Criteria

Enter selection values for one or more of the following table columns:

Recording date	Date of the recorded access (as a valid Db2 date expression)
Recording time	Time of the access (as a valid Db2 time expression)
Table creator	Creator of the table accessed
Table name	Name of the table accessed
Appl server	Name of the Db2 system where the SQL statement was executed
Db2 authid	Db2 userid issuing the recorded SQL statement
z/OS userid	z/OS userid issuing the recorded SQL statement
Correlation name	Name of the process issuing the recorded SQL statement (for example the z/OS job name)
Connection type	Type of the users Db2 connection ('batch', 'tso', ...)
Appl requester	Name of the Db2 system sending the SQL statement
Ext_application	Name of the external application issuing the SQL statement (distributed data access only)

Ext_workstation	Name of the workstation issuing the SQL statement (distributed data access only)
Program name	Name of the program containing the recorded SQL statement
Statement type	Type of the recorded SQL statement as SELECT, INSERT, DELETE or UPDATE
SQLCODE	SQLCODE resulting from statement execution
SQLROWS	Number of rows modified by the statement
Location	Name of the Db2 subsystem where the recorded statement was executed

Dynamic SQL

- Y will report SQL statements executed in dynamic mode
- N will report SQL statements executed in static mode
- blank will report both static and dynamic statements

Order by   Insert the column-names or column-numbers for sorting the report. If multiple columns are specified, they should be separated by a comma. The keywords ASC and DESC request an ascending or descending order.

For example:

ORDER BY Db2_ID, ACCESS_STAMP DESC

or

ORDER BY 5, 1 DESC

Criteria syntax rules:

- Criteria can be entered as a simple value, for example:
  TABLE NAME : **CUSTOMER**

- A generic value may be supplied using a trailing % sign, for example:
  PROGRAM NAME : **DSQ%**

- Selection criteria may be entered as a Db2 expression, for example:
  STATEMENT TYPE : **<> 'SELECT'**
  or
  RECORDING DATE : **> CURRENT DATE - 2 MONTHS**

- Db2 expressions on recorder table columns are executed using Db2 calls. Therefore, these expressions must obey Db2 syntax rules. All columns of the Recorder table, except SQLCODE and SQLROWS, have the CHARACTER format. Search values entered for CHARACTER columns must be enclosed in quotes.

- When multiple selection values are entered, the Recorder table rows must satisfy all criteria before being selected.

ORDER BY column-names and column-numbers

Column-nr	Column-name
1	ACCESS_STAMP
2	DATE(ACCESS_STAMP)
3	TIME(ACCESS_STAMP)
4	LOCATION
5	DB2_ID
6	MVS_ID
7	CORRELATION
10	TCREATOR
11	TNAME
12	PROGRAM
13	STMNTNR
14	SQLCODE
15	SQLROWS
16	OPCODE
17	DYNAMIC
18	CONNECT_ID
19	LUWID
20	EXT_SERVER
21	EXT_APPL
22	EXT_STATION

7.1.2 SQL text expressions

Recorded accesses may be selected by examining the text of the recorded SQL statement.

An SQL text expression consists of a **column_name** with an optional **column_value**. When a column_value is supplied, it is connected to the column_name by an **operator**.

When a column_name is specified alone, recorded statements will be reported as soon as they contain a reference to the column_name.

When a column_name with an operator and a column_value are specified, recorded statements are reported when they reference the column_name with the specified value.

The supplied expression is checked against:

- the INSERT VALUES clause
- the UPDATE SET clause
- the WHERE clause when present

Expression Syntax Rules

- The column expression has the format:

  column_name operator column_value

- The operator should be entered as:

=	to test equal
<	to test lower than
>	to test higher than
<=	to test lower or equal than
>=	to test higher or equal than
<>	to test not equal
LIKE	to perform a generic test using a trailing % sign

- Column_values may, but need not, be enclosed in quotes.

- Leading zeroes need not be supplied for a column that is logically numerical, even if it has been defined as character to Db2.

- The elements of the expression may, but need not, be separated by one or more blanks.

- Up to 3 column expressions can be specified on the criteria panel.

- When multiple column expressions are specified, a recorded statement will be reported only when it satisfies all the expressions.

Examples

(1) Table-Name   EMP_MED_HIST
   Statement-Type SELECT
   Text-expression-1 EMPNO = 100

   Reports all recorded SELECT's on the medical history of employee 100.

(2) Table-Name   EMPLOYEE
   Statement-Type UPDATE
   Text-expression-1 EMPNO = 100
   Text-expression-2 SALARY

   Reports all recorded UPDATE's on the SALARY of employee number 100.

(3) Table-Name   CUSTOMERS
   Text expression-1 CUSTNAME LIKE FREI%

   Reports all recorded accesses to customer names matching the generic specification.

7.2   PFkey Assignments on the Report Criteria Screen

PF1   Displays a help file.

PF2   Calls the User Access Summary Report using the specified report criteria.

PF3   Terminates the report program.

PF4   Allows to specify the name of a Db2 table into which DBARS recorder data have been stored[2]. By default, the DBARS RECORDER table is used for reporting. The Db2 tablename selected here will be remembered across DBARSIR sessions.

PF5   Recalls the previous report selection parameters[3]

PF6   Recalls the next report selection parameters

---

[2] For example:
- an archive table created using the TABLE keyword of the DBARSA program
- a report output table created using the OUTTABLE keyword of the DBARSRR program

[3] All report selection parameters are saved into a Db2 table, for the active TSO userid. Up to 32 sets of selection parameters are retained. Using the PF5 and PF6 keys, the user can browse these saved parameter sets.

7.3   Processing the Access Report

When the report criteria have been entered, DBARSIR selects the requested rows from the Recorder table into a list and displays the first page of that list on the terminal.

Program Function keys (PFkeys) are used to process the report list. Some of the PFkeys operate on the "current row", that is the row on the line pointed to by the cursor. To execute these functions, move the cursor to the line showing the object and press the corresponding PF key.

Only the PF keys labelled with a function name on the bottom of the screen are enabled.

PF1   Requests help.

PF2   Shows the SQL statement text in formatted mode.

PF3   Terminates list display.

PF4   If in list mode, takes a hardcopy of the entire list.
       If in page mode, takes a hardcopy of the current screen.

PF5   Invokes the "format" function.

       The format function displays all column names in the list, preceded by a +
       sign if the column is displayed or by a - sign if the column has been hidden
       previously. You may override the + or - sign in the following manner:

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

       -       Hides a column so that it is no longer displayed, although it
               remains in the list.

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

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

       2-9   Entering a number from 2 to 9 will move the column to the
               corresponding position in the list. The column previously on that
               position will take the position of the column moved.

PF6   Places the current line on the first line of the list.

PF7   Displays the previous page in the list.

PF8   Displays the next page in the list.

PF9   Makes the first line of the last page in the list current.

PF10   When displaying the list in pagemode, calls the "Search" function which allows to search list columns on their value. Search can be done using the logical operators = ^= > >. Functions are also provided to locate the highest and lowest column value in the list.

When in listmode, moves the display window to the left.

PF11   Moves the display window to the right if the screen capacity does not allow to display the entire list line.

PF12   Swaps the listformat between column and page mode.

In column mode, several table rows are displayed on one screen. In page mode, only one row is displayed.

ENTER Will make the object on the line pointed to by the cursor to the current object and highlight it. Other functions may operate on the "current object". Pressing ENTER is not required : it is sufficient to place the cursor on the required line and press the function PFkey. On return from the function, the object will be shown as current automatically.

7.4   User Access Summary Report

The User Access Summary reports accesses to audited Db2 tables in an hierarchical manner.

Before entering the Access Summary program, the user has specified search criteria that create a subset of the Recorder table rows.

On entry, the Access Summary displays the **User List** containing all distinct Db2 userids that appear in the selected Recorder table rows.

From the **User List**, the Detail function (PF4) displays the **Program List** for the selected user.

From the **Program List**, the Detail function displays the **Statement List** for the selected user and program.

Following PF keys are used to execute a particular function on a summary list. Some of the functions operate on the "current row", that is the row on the line pointed to by the cursor. When executing these functions, move the cursor to the line showing the object and press the corresponding PF key.

Only the PF keys labelled with a function name on the bottom of the screen are enabled.

PF1   Requests help.

PF3   Terminates list display or returns to a previous list, if any.

PF4   When in the User List, the Detail function produces a list of all programs by which the current userid has accessed Db2.

        When in the Program List, the Detail function produces a list of all program statements by which the current userid has accessed Db2.

PF6   Places the current line on the first line of the list.

PF7   Displays the previous page in the list.

PF8   Displays the next page in the list.

PF9   Makes the first line of the last page in the list current.

ENTER Will make the object on the line pointed to by the cursor to the current object and highlight it. Other functions may operate on the "current object". Pressing ENTER is not required : it is sufficient to place the cursor on the required line and press the function PFkey. On return from the function, the object will be shown as current automatically.

8 Interactive Reporting from a sequential or a VSAM Recorder

The online Access Reporting program DBARSRRI is used to scan a Recorder that has been defined as a sequential dataset or a VSAM cluster. DBARSRRI can be invoked only if the TSO terminal has at least 32 lines.

For a sequential Recorder, DBARSRRI is called from a TSO session using the TSO command:

EXEC 'DBARS.JCLLIB(DBARSRRS)' ['SSID=xxxx'] or using an installation defined ISPF shortcut. The SAM_PREFIX argument in the DBARSRRS member should be set to the SAM_PREFIX as defined in section SAM_PREFIX.

For a VSAM Recorder, DBARSRRI is called from a TSO session using the TSO command:

EXEC 'DBARS.JCLLIB(DBARSRRX)' ['SSID=xxxx'] or using an installation defined ISPF shortcut.

8.1 Entering Report Criteria

At entry into the DBARSRRI program, the report selection criteria are requested. Report criteria can be entered for the columns of the Recorder and for expressions in the recorded SQL statement text.

8.1.1 Recorder Column Criteria

Enter selection values for one or more of the following Recorder columns:

Recording date   Date of the recorded access[4]

Recording time   Time of the access[5]

Table creator   Creator of the table accessed

Table name   Name of the table accessed

Appl server   Name of the Db2 system where the SQL statement was executed

Db2 authid   Db2 userid issuing the recorded SQL statement

z/OS userid   z/OS userid issuing the recorded SQL statement

---

[4] Recording date can be entered as a valid Db2 date expression, with the following restrictions:
- The IN operator should not be used
- If the BETWEEN operator is used, the arguments should be absolute date/time values, for example 2008-10-02 or 10.20.22. Special registers are not allowed as BETWEEN arguments.

[5] Recording time can be entered as a valid Db2 time expression, with the same restrictions mentioned above for date expressions.

Correlation name   Name of the process issuing the recorded SQL statement (for example the z/OS job name)

Connection type   Type of the users Db2 connection ('batch', 'tso', ...)

Appl requester   Name of the Db2 system sending the SQL statement

Ext_application   Name of the external application issuing the SQL statement (distributed data access only)

Ext_workstation   Name of the workstation issuing the SQL statement (distributed data access only)

Program name   Name of the program containing the recorded SQL statement

Statement type   Type of the recorded SQL statement as SELECT, INSERT, DELETE or UPDATE

SQLCODE   SQLCODE resulting from statement execution

SQLROWS   Number of rows modified by the statement

Location   Name of the Db2 subsystem where the recorded statement was executed

Dynamic SQL

- Y will report SQL statements executed in dynamic mode
- N will report SQL statements executed in static mode
- blank will report both static and dynamic statements

DBARS RegionName

If the jobname of the DBARS address space is different from "DBARS", specify the jobname.

If multiple Db2 systems are recorded, you should specify the name of the DBARS region that records the intended Db2 system.

Criteria syntax rules:

- Criteria can be entered as a simple value, for example:
  TABLE NAME : **CUSTOMER**

- A generic value may be supplied using a trailing % sign, for example:
  PROGRAM NAME : **DSQ%**

- Selection criteria may be entered as a Db2 expression, for example:
  STATEMENT TYPE : **<> 'SELECT'**
  or
  RECORDING DATE : **> CURRENT DATE - 2 MONTHS**

- Db2 expressions must obey Db2 syntax rules.
- When multiple selection values are entered, the Recorder table rows must satisfy all criteria before being selected.

8.1.2 Accelerating the Recorder scan

Since the Recorder is a VSAM ESDS, scanning is essentially a sequential process. However, DBARS maintains an index on the ESDS to allow for faster scans. The Recorder index is on date and time.

To speed up a scan, specify the Recording date column as a non-generic value (for example CURRENT DATE or 2008-12-12). When a date is supplied, the scan may be further accelerated by supplying a time value (which may be generic).

8.1.3 SQL text expressions

Recorded accesses may be selected by examining the text of the recorded SQL statement.

An SQL text expression consists of a **column_name** with an optional **column_value**. When a column_value is supplied, it is connected to the column_name by an **operator**.

When a column_name is specified alone, recorded statements will be reported as soon as they contain a reference to the column_name.

When a column_name with an operator and a column_value are specified, recorded statements are reported when they reference the column_name with the specified value.

The supplied expression is checked against:

- the INSERT VALUES clause
- the UPDATE SET clause
- the WHERE clause when present

Expression Syntax Rules

- The column expression has the format:

  column_name   operator   column_value

- The operator should be entered as:

=	to test equal
<	to test lower than
>	to test higher than
<=	to test lower or equal than
>=	to test higher or equal than
<>	to test not equal
LIKE	to perform a generic test using a trailing % sign

- Column_values may, but need not, be enclosed in quotes.

- Leading zeroes need not be supplied for a column that is logically numerical, even if it has been defined as character to Db2.

- The elements of the expression may, but need not, be separated by one or more blanks.

- Up to 3 column expressions can be specified on the criteria panel.

- When multiple column expressions are specified, a recorded statement will be reported only when it satisfies all the expressions.

Examples

(1)   Table-Name   EMP_MED_HIST
        Statement-Type   SELECT
        Text-expression-1   EMPNO = 100

        Reports all recorded SELECT's on the medical history of employee 100.

(2)   Table-Name   EMPLOYEE
        Statement-Type   UPDATE
        Text-expression-1   EMPNO = 100
        Text-expression-2   SALARY

        Reports all recorded UPDATE's on the SALARY of employee number 100.

(3)   Table-Name   CUSTOMERS
        Text expression-1   CUSTNAME LIKE FREI%

        Reports all recorded accesses to customer names matching the generic specification.

8.2 PFkey Assignments on the Report Criteria Screen

PF1 Displays a help file.

PF3 Terminates the report program.

PF5 Recalls the previous report selection parameters (*)

PF6 Recalls the next report selection parameters (*)

(*) Note

All report selection parameters are saved into a Db2 table, for the active TSO userid. Up to 32 sets of selection parameters are retained. Using the PF5 and PF6 keys, the user can browse these saved parameter sets.

8.3　Processing the Access Report

When the report criteria have been entered, DBARSRRI selects the requested rows from the Recorder table a list and displays the first page of that list on the terminal.

Program Function keys (PFkeys) are used to process the report list. Some of the PFkeys operate on the "current row", that is the row on the line pointed to by the cursor. To execute these functions, move the cursor to the line showing the object and press the corresponding PF key.

Only the PF keys labelled with a function name on the bottom of the screen are enabled.

PF1　Requests help.

PF2　Shows the SQL statement text in formatted mode.

PF3　Terminates list display.

PF4　If in list mode, takes a hardcopy of the entire list.
　　　If in page mode, takes a hardcopy of the current screen.

PF5　Invokes the "format" function.

　　　The format function displays all column names in the list, preceded by a + sign if the column is displayed or by a - sign if the column has been hidden previously. You may override the + or - sign in the following manner:

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

　　　-　　　Hides a column so that it is no longer displayed, although it remains in the list.

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

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

　　　2-9　　Entering a number from 2 to 9 will move the column to the corresponding position in the list. The column previously on that position will take the position of the column moved.

PF6　Places the current line on the first line of the list.

PF7　Displays the previous page in the list.

PF8　Displays the next page in the list.

PF9　Makes the first line of the last page in the list current.

PF10 When displaying the list in pagemode, calls the "Search" function which allows to search list columns on their value. Search can be done using the logical operators = ^= > >. Functions are also provided to locate the highest and lowest column value in the list.

When in listmode, moves the display window to the left.

PF11 Moves the display window to the right if the screen capacity does not allow to display the entire list line.

PF12 Swaps the listformat between column and page mode.

In column mode, several table rows are displayed on one screen. In page mode, only one row is displayed.

ENTER Will make the object on the line pointed to by the cursor to the current object and highlight it. Other functions may operate on the "current object". Pressing ENTER is not required : it is sufficient to place the cursor on the required line and press the function PFkey. On return from the function, the object will be shown as current automatically.

9 Batch Access Reporting from a sequential file or a VSAM Recorder

The current DBARS Recorder dataset is processed by the batch DBARSRR program, to generate a printed report or a result table. Using a result table allows to import from the VSAM Recorder into the Recorder table.

The result table can be:

- the DBARS.RECORDER table created using the DBARSTAB installation job
- a table created previously with a table structure identical to that of the DBARS Recorder (created "LIKE")

After the reporting run, the result table can be displayed using interactive DBARS reporting, as shown above.

The DBARSRR program determines which of the Recorders (RECORDA or RECORDB) is the current one. Therefore, DD statements for both recorders must be supplied, except for a sequential Recorder, which is dynamically allocated by the program.

Report control and report selection statements are supplied after a DD SYSIN.

9.1   Report Control Statements

SSID xxxx
>Specify the name of the DB2 subsystem where the DBARS Recorder resides. Alternatively, use a xxxx.SDSNEXIT DD to identify that subsystem. SSID should be the first statement in the SYSIN stream.

DBARS_JN *nnn*
>If the jobname of the DBARS address space is different from "DBARS", specify the jobname in the DBARS_JN statement. If multiple Db2 systems are recorded, you should specify the name of the DBARS region that records the intended Db2 system.

SAM_PREFIX *nnn*
>If the Recorder is a sequential dataset, include the SAM_PREFIX defined in section SAM_PREFIX

OUTTABLE *nnn* [APPEND]
>The OUTTABLE statement requests report output to the result table named *nnn*. That table should have been created previously by the user. The APPEND option allows to extend an existing report table with a new report. Without APPEND, DBARSRR will delete all rows, if any, from the target result table.

OUTLINES {*nn*|66}
>The OUTLINES statement requests a printed report and specifies the number of lines per page.

>OUTTABLE and OUTLINES are mutually exclusive. If neither OUTTABLE or OUTLINES is specified, the report will be printed with a linecount of 66.

DB2_REC
>Specify when the DBARS Recorder is a DB2 table.

SEQREC
>Specify the keyword for a VSAM Recorder.

SEQRECS

>Specify the keyword for a sequential Recorder. An XML Recorder cannot be scanned using DBARSRR.

9.2 Selection Statement Syntax

Recorder selection criteria are coded on SYSIN. They have the following syntax:
variable operator value

- **VARIABLE** names the Recorder field to be tested as follows:

DATE	Date of recorded access (a valid Db2 date expression)
TIME	Time of recorded access (a valid Db2 time expression)
TCREATOR	Creator of the recorded table
TNAME	Name of the recorded table
DB2ID	Db2 userid issuing the recorded SQL statement
ZOSID	z/OS userid issuing the recorded SQL statement
CORRELATION	Name of the process issuing the recorded SQL statement
CONNECTION	Type of the users Db2 connection ('batch', 'tso', ...)
PROGRAM	Name of the program containing the recorded SQL statement
STMTYPE	SQL statement type as SELECT, INSERT, DELETE, UPDATE
SQLCODE	SQLCODE resulting from statement execution
SQLROWS	Number of rows modified by the statement
SERVER	Db2 subsystem where the recorded statement was executed
DYNAMIC	**Y** will report SQL statements executed in dynamic mode
	N will report SQL statements executed in static mode
	blank will report both static and dynamic statements
EXT_SERVER	Name of the Db2 system sending the SQL statement(distributed data access only)
EXT_APPL	Name of the external application sending the SQL statement (distributed data access only)
EXT_WS	Name of the workstation from which the SQL statement was sent (distributed data access only)

- **OPERATOR** requests the test operation to be performed between "variable" and "value", as follows:

=	to test equal
<	to test lower than
>	to test higher than
<=	to test lower or equal than
>=	to test higher or equal than
<>	to test not equal
LIKE	to perform a generic test using trailing % sign

- **VALUE**

  - Alpha values may, but need not, be enclosed in quotes.
  - A trailing % sign used with the = operator will act as a LIKE operator.
  - Leading zeroes need not be supplied for a column that is logically numerical.
  - The elements of the expression may, but need not, be separated by one or more blanks.
  - If multiple column expressions are specified, a recorded statement will be reported only when it satisfies all the expressions.
  - To test whether a column value is present, use the expression <> ''.
  - For example: EXT_SERVER <> ''
  - Db2 date/time expressions may be entered for the DATE and TIME columns. For example : DATE >= CURRENT DATE – 1 MONTH

9.3 Sample JCL

```
//DBARSRR   JOB (...),'...',REGION=8M,CLASS=A,MSGCLASS=A,
//          USER=...,PASSWORD=...
//DBARSRR   EXEC PGM=DBARSRR
//STEPLIB   DD DSN=DBARS.LOADLIB,DISP=SHR
//RECORDA   DD DSN=DBARS.RECORDER.A,DISP=SHR
//RECORDB   DD DSN=DBARS.RECORDER.B,DISP=SHR
//SYSPRINT  DD SYSOUT=A
//SYSUDUMP  DD SYSOUT=A
//SYSIN     DD *
DATE = CURRENT DATE
TIME = 14.05%
/*
```

The above JCL will scan the active Recorder and print all accesses at 14.05 of the current date.

10 Exception Reporting

The online Exception Reporting program DBARSXR is used to scan the DBARS EXCEPTIONS table.

DBARSXR is called from a TSO session using the TSO command

CALL 'DBARS.LOADLIB(DBARSXR)' ['SSID=xxxx']

or using an installation defined ISPF shortcut.

The SSID parameter on the above CALL statement can be used if the TSO session is not implicitly connected to the Db2 subsystem where the DBARS tables reside, that is, if the TSO login procedure does not contain a DD statement for the corresponding SDSNEXIT dataset.

DBARSXR can be invoked only if the TSO terminal has at least 32 lines.

10.1 Entering Report Criteria

At entry into the DBARSXR program, the report selection criteria are requested. Report criteria can be entered for the columns of the Exceptions table.

Enter selection values for one or more of the following table columns:

Recording date	Date of the recorded access (as a valid Db2 date expression)
Recording time	Time of the access (as a valid Db2 time expression)
Exception Code	Exception code stored by the Exception user exit
Table creator	Creator of the table accessed
Table name	Name of the table accessed
Appl server	Name of the Db2 system where the SQL statement was executed
Db2 authid	Db2 userid issuing the recorded SQL statement
z/OS userid	z/OS userid issuing the recorded SQL statement
Correlation name	Name of the process issuing the recorded SQL statement (for example the z/OS job name)
Connection type	Type of the users Db2 connection ('batch', 'tso', ...)
Appl requester	Name of the Db2 system sending the SQL statement
Ext_application	Name of the external application issuing the SQL statement (distributed data access only)
Ext_workstation	Name of the workstation issuing the SQL statement (distributed data access only)

Program name   Name of the program containing the recorded SQL statement

Statement type   Type of the recorded SQL statement
as SELECT, INSERT, DELETE or UPDATE

Exception   The reason that caused the SQL access to be stored in the exceptions table:

- **"Alerted"** when DBARS issued an alert, based on a DBARS access rule.
- **"Blocked"** when DBARS has blocked statement execution, based on a DBARS access rule.
- The numerical value of the **returncode** passed by a DBARS user exit.

SQLROWS   Number of rows modified by the statement

Location   Name of the Db2 subsystem where the recorded statement was executed

Dynamic SQL

- Y will report SQL statements executed in dynamic mode
- N will report SQL statements executed in static mode
- blank will report both static and dynamic statements

Order by   Insert the column-names or column-numbers for sorting the report. If multiple columns are specified, they should be separated by a comma. The keywords ASC and DESC request an ascending or descending order.

For example:

ORDER BY Db2_ID, ACCESS_STAMP DESC

or

ORDER BY 5, 1 DESC

Criteria syntax rules:

- Criteria can be entered as a simple value, for example:
  TABLE NAME : **CUSTOMER**

- A generic value may be supplied using a trailing % sign, for example:
  PROGRAM NAME : **DSQ%**

- Selection criteria may be entered as a Db2 expression, for example:
  STATEMENT TYPE : **<> 'SELECT'**
  or
  RECORDING DATE : **> CURRENT DATE - 2 MONTHS**

- Db2 expressions on recorder table columns are executed using Db2 calls. Therefore, these expressions must obey Db2 syntax rules. All columns of the Recorder table, except SQLCODE and SQLROWS, have the CHARACTER format. Search values entered for CHARACTER columns must be enclosed in quotes.

- When multiple selection values are entered, the Recorder table rows must satisfy all criteria before being selected.

ORDER BY column-names and column-numbers

Column-nr	Column-name
1	ACCESS_STAMP
2	DATE(ACCESS_STAMP)
3	TIME(ACCESS_STAMP)
4	EXCEPTION CODE
5	LOCATION
6	DB2_ID
7	MVS_ID
8	CORRELATION
11	TCREATOR
12	TNAME
13	PROGRAM
14	STMNTNR
15	Exception Reason
16	SQLROWS
17	OPCODE
18	DYNAMIC
19	CONNECT_ID
20	LUWID
21	EXT_SERVER
22	EXT_APPL
23	EXT_STATION

10.2 PFkey Assignments on the Report Criteria Screen

PF1   Displays a help file.

PF3   Terminates the report program.

PF5   Recalls the previous report selection parameters[6]

PF6   Recalls the next report selection parameters

---

[6] All report selection parameters are saved into a Db2 table, for the active TSO userid. Up to 32 sets of selection parameters are retained. Using the PF5 and PF6 keys, the user can browse these saved parameter sets.

10.3 Processing the Access Report

When the report criteria have been entered, DBARSXR extracts the requested rows from the Exceptions table into a list and displays the first page of that list on the terminal.

Program Function keys (PFkeys) are used to process the report list. Some of the PFkeys operate on the "current row", that is the row on the line pointed to by the cursor. To execute these functions, move the cursor to the line showing the object and press the corresponding PF key.

Only the PF keys labelled with a function name on the bottom of the screen are enabled.

PF1 Requests help.

PF2 Shows the SQL statement text in formatted mode.

PF3 Terminates list display.

PF4 If in list mode, takes a hardcopy of the entire list.
     If in page mode, takes a hardcopy of the current screen.

PF5 Invokes the "format" function.

The format function displays all column names in the list, preceded by a + sign if the column is displayed or by a - sign if the column has been hidden previously. You may override the + or - sign in the following manner:

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

- Hides a column so that it is no longer displayed, although it remains in the list.

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

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

2-9 Entering a number from 2 to 9 will move the column to the corresponding position in the list. The column previously on that position will take the position of the column moved.

PF6 Places the current line on the first line of the list.

PF7 Displays the previous page in the list.

PF8 Displays the next page in the list.

PF9 Makes the first line of the last page in the list current.

PF10   When displaying the list in pagemode, calls the "Search"
       function which allows to search list columns on their value.
       Search can be done using the logical operators = ^= > >.
       Functions are also provided to locate the highest and lowest
       column value in the list.

        When in listmode, moves the display window to the left.

PF11   Moves the display window to the right if the screen capacity does
       not allow to display the entire list line.

PF12   Swaps the listformat between column and page mode.

       In column mode, several table rows are displayed on one screen.
       In page mode, only one row is displayed.

ENTER Will make the object on the line pointed to by the cursor to the
       current object and highlight it. Other functions may operate on
       the "current object". Pressing ENTER is not required : it is
       sufficient to place the cursor on the required line and press the
       function PFkey. On return from the function, the object will be
       shown as current automatically.

11 Archiving the DBARS Recorder

The DBARSA program transfers a selected number of rows from the Recorder to a sequential dataset (identified by the ARCHIVE DD) or to a Db2 table.

11.1 KEEP Statement

The KEEP statement is used only with a Recorder on a Db2 table. KEEP is entered in the PARM field of the EXEC DBARSA statement.

If the KEEP statement is omitted, all Recorder rows are transferred to the archive. This is the default archiving strategy.

Syntax:

KEEP n {DAY(S)|MONTH(S)|YEAR(S)} [NOPURGE]

The KEEP operand states how many recorder rows must be kept in the Recorder table, for example: 6 MONTHS, 1 YEAR. The remaining rows are exported to the Archive.

The KEEP specification is used to build the WHERE clause on the Recorder SELECT and DELETE statements issued by DBARSA:

WHERE DATE(ACCESS_STAMP) <= CURRENT DATE – (KEEP_specification)

The NOPURGE operand states that the Recorder rows should be written to the Archive without being deleted from the Recorder table.

11.2 JOBNAME Statement

The statement is used only with a Recorder on a Db2 table. JOBNAME is entered in the PARM field of the EXEC DBARSA statement.

Syntax:

JOBNAME <jobname of the DBARS address space>

If JOBNAME is omitted, it defaults to DBARS.

11.3 SEQREC[S] statement

The SEQREC statement indicates that the Recorder is a VSAM cluster. The SEQRECS statement indicates that the Recorder is a sequential dataset. SEQREC[S] is entered in the PARM field of the EXEC DBARSA statement.

Syntax:

```
SEQREC      [Jobname] [RESTART|NORESTART] [TABLE <owner.name>]
            [RECID A|B]

SEQRECS     [Jobname] SAM_PREFIX <prefix> [RESTART|NORESTART]
            [TABLE <owner.name>] [RECID A|B]
```

- The optional Jobname should be entered only when the z/OS JOB name for the main DBARS address space is different from "DBARS".
- For a sequential Recorder (SEQRECS) specify the SAM_PREFIX defined in section SAM_PREFIX.
- When the previous archive run did not complete normally, the RESTART or the NORESTART argument must be supplied.
  - RESTART should be specified to continue archiving the recorder which was being processed when the abend occurred.
  - NORESTART should be specified only when it is not possible to resume the archive.

  The restart arguments are ignored when the previous archive run was successful.
- The TABLE keyword allows to archive the Recorder to a named Db2 table. This table should have been created previously LIKE the DBARS Recorder table. The archive table can be processed using the DBARSIR program, as described earlier in section 7.2.
- The RECID keyword allows to explicitly archive a named Recorder. This facility should only be used in an automated archive procedure, as described in Automated Recorder Archiving. In all other cases, DBARSA will determine the current Recorder and archive it.

Notes:

1. Unless archiving is from a Db2 table, DD statements must be provided for both RECORDA and RECORDB.

2. Unless archiving is to a Db2 table, a DD statement must be provided for the ARCHIVE dataset.

3. The DBARS distribution file provides the member DBARS\JCL\DBARSCAT with a sample JCL stream to create a DBARS archive table.

4. The archive program automatically determines the current recorder cluster to be archived. During archiving, recording continues on the alternate cluster.

11.4 Coding the ARCHIVE DD

The disposition of the ARCHIVE dataset and the physical volume specifications are the responsibility of the user.

If a recorder archive is kept for a given period (a quarter for example), a DISP=(NEW,CATLG) should be specified at the begin of the period. Within the archiving period, a DISP=MOD could be used to extend the current archive volume.

11.5 Archive a Db2 Recorder to disk

```
//DBARSA      JOB (xxxx),'xxx',REGION=4M,CLASS=A,MSGCLASS=A,
//            USER=xxxx,PASSWORD=xxxx
//DBARSA      EXEC PGM=DBARSA
//ARCHIVE     DD DSN=DBARS.ARCHIVE,DISP=(MOD),UNIT=SYSDA
//STEPLIB     DD DSN=DBARS.LOADLIB,DISP=SHARE
//SYSPRINT    DD SYSOUT=A
//SYSUDUMP    DD SYSOUT=A
```

11.6 Archive a VSAM Recorder to disk

```
//DBARSA      JOB (xxxx),'xxx',REGION=4M,CLASS=A,MSGCLASS=A,
//            USER=xxxx,PASSWORD=xxxx
//DBARSA      EXEC PGM=DBARSA,PARM='SEQREC'
//ARCHIVE     DD DSN=DBARS.ARCHIVE,DISP=(MOD),UNIT=SYSDA
//RECORDA     DD DSN=DBARS.RECORDER.A,DISP=SHR
//RECORDB     DD DSN=DBARS.RECORDER.B,DISP=SHR
//STEPLIB     DD DSN=DBARS.LOADLIB,DISP=SHARE
//SYSPRINT    DD SYSOUT=A
//SYSUDUMP    DD SYSOUT=A
```

11.7 Archive a VSAM Recorder to a Db2 table

```
//DBARSA      JOB (xxxx),'xxx',REGION=4M,CLASS=A,MSGCLASS=A,
//            USER=xxxx,PASSWORD=xxxx
//DBARSA      EXEC PGM=DBARSA,PARM='SEQREC TABLE DBARS.ARCHIVE'
//RECORDA     DD DSN=DBARS.RECORDER.A,DISP=SHR
//RECORDB     DD DSN=DBARS.RECORDER.B,DISP=SHR
//STEPLIB     DD DSN=DBARS.LOADLIB,DISP=SHARE
//SYSPRINT    DD SYSOUT=A
//SYSUDUMP    DD SYSOUT=A
```

11.8   Combined Recorder Archiving

When the Recorder has been defined as a VSAM cluster, it can be archived frequently (perhaps daily) to the DBARS.RECORDER table, using the SEQREC TABLE option.

The DBARS.RECORDER table can then be used for fast Recorder access, using the DBARSIR program.

The DBARS.RECORDER can be archived to a sequential dataset using the KEEP option of the DBARSA program.

12 Access Reporting from an Archive

The DBARS archive dataset, identified by the DD ARCHIVE, is processed by the DBARSAR program, to generate a printed report or a result table. Using a result table allows to import from an archive into the Recorder table.

The result table should be a Db2 table created previously with a table structure identical to that of the DBARS Recorder (created "LIKE"). After the reporting run, the result table can be displayed using interactive DBARS reporting. It is possible to use the DBARS.RECORDER table as OUTTABLE.

Report control and report selection statements are supplied after a DD SYSIN.

12.1 Report Control Statements

SSID xxxx

> Specify the name of the DB2 subsystem where the DBARS Recorder resides. Alternatively, use a xxxx.SDSNEXIT DD to identify that subsystem. SSID should be the first statement in the SYSIN stream.

OUTTABLE *nnn* [APPEND]

> The OUTTABLE statement requests archive report output to the result table named *nnn*. That table should have been created previously by the user. The APPEND option allows to extend an existing report table with a new report. Without APPEND, DBARSAR will delete all rows, if any, from the target result table.

OUTLINES {nn|66}

> The OUTLINES statement requests a printed archive report and specifies the number of lines per page.

OUTTABLE and OUTLINES are mutually exclusive. If neither OUTTABLE or OUTLINES is specified, the archive report will be printed with a linecount of 66.

12.2 Selection Statement Syntax

Archive selection criteria are coded on SYSIN. They have the following syntax:
variable operator value

- **VARIABLE** names the Recorder field to be tested as follows:

DATE	Date of recorded access (a valid Db2 date expression)
TIME	Time of recorded access (a valid Db2 time expression)
TCREATOR	Creator of the recorded table
TNAME	Name of the recorded table
DB2ID	Db2 userid issuing the recorded SQL statement
ZOSID	z/OS userid issuing the recorded SQL statement
CORRELATION	Name of the process issuing the recorded SQL statement
CONNECTION	Type of the users Db2 connection ('batch', 'tso', ...)
PROGRAM	Name of the program containing the recorded SQL statement
STMTYPE	SQL statement type as SELECT, INSERT, DELETE, UPDATE
SQLCODE	SQLCODE resulting from statement execution
SQLROWS	Number of rows modified by the statement
SERVER	Db2 subsystem where the recorded statement was executed
DYNAMIC	**Y** will report SQL statements executed in dynamic mode
	N will report SQL statements executed in static mode
	blank will report both static and dynamic statements
EXT_SERVER	Name of the Db2 system sending the SQL statement(distributed data access only)
EXT_APPL	Name of the external application sending the SQL statement (distributed data access only)
EXT_WS	Name of the workstation from which the SQL statement was sent (distributed data access only)

- **OPERATOR** requests the test operation to be performed between "variable" and "value", as follows:

=	to test equal
<	to test lower than
>	to test higher than
<=	to test lower or equal than
>=	to test higher or equal than
<>	to test not equal
LIKE	to perform a generic test using trailing % sign

- **VALUE**

  - Alpha values may, but need not, be enclosed in quotes.
  - Db2 date/time expressions may be entered for the DATE and TIME columns. For example : DATE >= CURRENT DATE – 1 MONTH
  - A trailing % sign used with the = operator will act as a LIKE operator.
  - Leading zeroes need not be supplied for a column that is logically numerical.
  - The elements of the expression may, but need not, be separated by one or more blanks.
  - If multiple column expressions are specified, a recorded statement will be reported only when it satisfies all the expressions.
  - To test whether a column value is present, use the expression <> ``.

12.3 Sample JCL

```
//DBARSAR   JOB (....),'...',REGION=4M,CLASS=A,MSGCLASS=A,
//          USER=....,PASSWORD=....
//DBARSAR   EXEC PGM=DBARSAR
//STEPLIB   DD DSN=DBARS.LOADLIB,DISP=SHARE
//ARCHIVE   DD DSN=DBARS.ARCHIVE.Y2008.Q4,DISP=OLD
//SYSPRINT  DD SYSOUT=A
//SYSUDUMP  DD SYSOUT=A
//SYSIN     DD *
DATE = 2008-10-11
DB2ID = U1234
TNAME = CUSTOMERS
STMTYPE = UPDATE
/*
```

The above JCL will scan the archive and print all updates of the customer table by the named user on the specified date.

13   Writing a DBARS User Exit

A DBARS user exit is declared with an EXIT statement in the EXECPARM member of the DBARS JCLLIB. The user exit is written in REXX and called by the DBARS Writer task before the access is stored in the Recorder table.

The exit runs under the DBARS userid. At entry, there is a connection to the Db2 system that holds the DBARS Recorder table. The exit may perform any necessary action including SQL access. However, if SQL statements are issued, a COMMIT should be used before returning to DBARS.

Following arguments are passed to the exit:

Argument Number	Argument Content	Argument Length
1	Access timestamp	26
2	TCB address	8
3	Name of Db2 application server	16
4	Db2 userid	8
5	z/OS userid	8
6	Correlation name	12
7	Connection type	8
8	LUW_id	12
9	External Server Name	16
10	External Application Name	32
11	External Workstation Name	18
12	RDI Call Type	4
13	RDI Statement Type	4
14	Table Creator	8
15	Table Name	128
16	Program Name	8
17	Program Section number	4
18	SQLCODE	8
19	Number of rows processed	16
20	Operation Code	8
21	Dynamic access flag	1
22	Length of SQL Statement_text	8
23	Pointer to SQL Statement_text	8

The exit may pass a returncode 8 to bypass the recording of the SQL statement. Exiting with a returncode other than 8, will record the statement.

DBARS Sample User Exit

```
/*      Sample DBARS User Exit      */

Arg Stamp ,
TCB ,
Db2_server ,
Db2_userid ,
zOS_userid ,
Correlation ,
Connection ,
LUWid ,
External_Server ,
External_Application ,
External_Workstation ,
RDI_Call_Type ,
RDI_Statement_Type ,
Table_Creator ,
Table_Name ,
Program_Name ,
Program_Section ,
SQLCODE ,
Rows_modified ,
Operation ,
Dynamic_access ,
Length_SQL_Statement ,
SQL_Statement ,
.

/*      Display all input arguments, except statement text */

Say Stamp TCB Db2_server Db2_userid zOS_userid Correlation Connection
Say LUWid External_Server External_Application External_Workstation
Say RDI_Call_Type RDI_Statement_Type Table_Creator Table_Name
Say Program_Name Program_Section SQLCODE Rows_processed Operation

/*      Display the SQL statement text */

Say STORAGE(SQL_Statement,Length_SQL_Statement)

Exit 0
```

14   Writing a compiled DBARS User Exit

A compiled DBARS user exit is declared with an EXITMOD statement in the EXECPARM member of the DBARS JCLLIB, as described in section 3.4.

The exit load module is invoked by the DBARS Writer before storing the audit record to the DBARS Recorder. It is assumed that the exit is written in Assembler.

On entry to the exit, following general registers have been setup:

Register 0	address a workarea, described later in this section
Register 1	address a parameter list, described later in this section
Register 13	address of a 72-byte register savearea, for use by the exit
Register 14	return address to the DBARS Writer
Register 15	entrypoint address of the exit

When DBARS is stopped, the exit is invoked with binary zeroes in Register 1.

When the exit returns to DBARS, a returncode should be specified in register 15.

If register 15 is non-zero, the current audit record is not written to the Recorder.
If register 15 is zero, the current audit record is written to the Recorder.

Format of the parameterlist

- Address of the current audit record
- Up to 8 addresses of optional parameters specified on the EXITMOD startup statement

Format of the exit workarea

- 4 double-words for use by the exit. The area is set to binary zeroes when the exit is called for the first time.
- The 8-byte current DBARS version number as $V_{version\_nr}R_{release\_nr}M_{modification\_nr}$
- A fullword containing the current session number, if session recording has been enabled.

Format of the audit record passed

```
IN_RECORD        DSECT
STAMP            DS  CL26       ;EXECUTION TIMESTAMP
INSERT_STAMP     DS  CL26       ;RECORDING TIMESTAMP
TCB              DS  F          ;ALLIED TCB ADDRESS
                 ORG TCB        ;IF SESSION CONTROL ACTIVE,
SESSION_NR       DS  F          ;SESSION NUMBER ASSIGNED BY DBARS
LOCATION         DS  CL16       ;Db2 SUBSYSTEM
X_DB2ID          DS  CL8        ;USER Db2 ID
X_MVSID          DS  CL8        ;USER Z/OS ID
TABLE_CREATOR    DS  CL8        ;CREATOR OF TABLE ACCESSED
TABLE_NAME       DS  CL64       ;NAME OF TABLE ACCESSED
CORRELATION      DS  CL12       ;CORRELATION (EG JOBNAME)
X_RDIPROGN       DS  CL8        ;PROGRAM NAME
X_RDISECTN       DS  H          ;PROGRAM SECTION NUMBER
X_RDICTYPE       DS  H          ;Db2 CALL TYPE
X_RDISTYPE       DS  H          ;Db2 STATEMENT TYPE
X_SQLCODE        DS  F          ;SQLCODE
X_SQLROWS        DS  F          ;SQLROWS
STMTYPE          DS  CL8        ;EXTERNALIZED STATEMENT TYPE
DYNAMIC          DS  CL1        ;Y IF DYNAMIC STATEMENT
CONNECTION       DS  CL8        ;CONNECTION (CICS, TSO, DDF, BATCH)
LUWID            DS  CL12       ;LUWID
EXT_SERVER       DS  CL16       ;EXTERNAL SERVER IP-ADDRESS
EXT_APPL         DS  CL32       ;EXTERNAL APPLICATION
EXT_WS           DS  CL18       ;EXTERNAL WORKSTATION NAME
                 DC  CL1' '     ;ALIGNMENT FOR STMNT_TEXT
STMNT_TEXT       DS  H,CL30720  ;STATEMENT TEXT
```

Sample EXITMOD

```
EXITMOD               START 0
                      STM   R14,R12,12(R13)     ;SAVE DBARS REGISTERS
                      LR    R3,R15
                      USING EXITMOD,R3          ;ADDRESSABILITY FOR EXITMOD
                      LA    R15,SAVEAREA        ;GET A LOCAL SAVEAREA
                      ST    R13,4(,R15)         ;CHAIN TO DBARS SAVEAREA
                      LR    R13,R15
                      LTR   R1,R1
                      JZ    DBARS_RETURN        ;BR IF DBARS IS BEING STOPPED
*
                      L     R4,0(,R1)           ;ADDRESS AUDIT RECORD
                      USING INREC,R4
                      LR    R10,R0              ;ADDRESS WORKAREA
                      USING WORK,R10
*
*                                               ;PROCESS INREC DATA
*
DBARS_RETURN          DS    0H
                      L     R13,4(,R13)         ;BACKCHAIN TO DBARS SAVEAREA
                      LM    R14,R12,12(R13)     ;RELOAD DBARS REGISTERS
                      XR    R15,R15             ;RC = WRITE AUDIT RECORD
                      BR    R14                 ;RETURN TO DBARS
*
SAVEAREA              DS    9D
R0                    EQU   0                   ;REGISTER EQUATES
R1                    EQU   1
R2                    EQU   2
R3                    EQU   3
R4                    EQU   4
R5                    EQU   5
R6                    EQU   6
R7                    EQU   7
R8                    EQU   8
R9                    EQU   9
R10                   EQU   10
R11                   EQU   11
R12                   EQU   12
R13                   EQU   13
R14                   EQU   14
R15                   EQU   15
*
```

```
INREC              DSECT
STAMP              DS   CL26        ;EXECUTION TIMESTAMP
INSERT_STAMP       DS   CL26        ;RECORDING TIMESTAMP
TCB                DS   F           ;ALLIED TCB ADDRESS
                   ORG  TCB         ;IF SESSION CONTROL ACTIVE,
SESSION_NR         DS   F           ;SESSION NUMBER ASSIGNED BY DBARS
LOCATION           DS   CL16        ;Db2 SUBSYSTEM
X_DB2ID            DS   CL8         ;USER Db2 ID
X_MVSID            DS   CL8         ;USER Z/OS ID
TABLE_CREATOR      DS   CL8         ;CREATOR OF TABLE ACCESSED
TABLE_NAME         DS   CL64        ;NAME OF TABLE ACCESSED
CORRELATION        DS   CL12        ;CORRELATION (EG JOBNAME)
X_RDIPROGN         DS   CL8         ;PROGRAM NAME
X_RDISECTN         DS   H           ;PROGRAM SECTION NUMBER
X_RDICTYPE         DS   H           ;Db2 CALL TYPE
X_RDISTYPE         DS   H           ;Db2 STATEMENT TYPE
X_SQLCODE          DS   F           ;SQLCODE
X_SQLROWS          DS   F           ;SQLROWS
STMTYPE            DS   CL8         ;EXTERNALIZED STATEMENT TYPE
DYNAMIC            DS   CL1         ;Y IF DYNAMIC STATEMENT
CONNECTION         DS   CL8         ;CONNECTION (CICS, TSO, DDF, BATCH)
LUWID              DS   CL12        ;LUWID
EXT_SERVER         DS   CL16        ;EXTERNAL SERVER IP-ADDRESS
EXT_APPL           DS   CL32        ;EXTERNAL APPLICATION
EXT_WS             DS   CL18        ;EXTERNAL WORKSTATION NAME
                   DC   CL1' '      ;ALIGNMENT FOR STMNT_TEXT
STMNT_TEXT         DS   H,CL30720   ;STATEMENT TEXTINREC
*
*
WORK               DSECT
USER_WORK          DS   4D          ;FOR USE BY EXIT
DBARS_VERSION      DS   CL8         ;DBARS VERSION NUMBER
*
*
                   END
```

15   Reporting Access Exceptions

An installation may wish to screen all data accesses to audited tables by examining, within a DBARS User Exit, the data captured by DBARS and presented to the Exit as entry arguments.

15.1   Detecting an access exception

Access exceptions may be detected:

- By checking combined values in the exit entry arguments, using REXX statements.

  The Sample Exception Exit uses the arguments **Table_Name** and **Db2_server** to detect remote access to the Customer table.

- By scanning the SQL statement text for occurrence of unusual values in specified table columns. To facilitate SQL text scanning, DBARS provides the DBARSRXI function.

  The function is called as follows:

  address linkmvs "DBARSXRI SCAN statement expression"

  where:

  statement
      Is the text of the SQL statement received as input.

  expression
      Is a logical expression specified in Db2 syntax. The expression states a column name and a column value. DBARSXRI will check the INSERT VALUES and the WHERE clauses to detect whether the statement accesses the named column with the named value. If it does, DBARSXRI returns an rc=1. Else rc=0 is returned.

15.2   Signalling an access exception

- An access exception will be stored into the DBARS EXCEPTIONS table, when the exit returns to DBARS with a returncode greater than 12. The returncode is then considered as an exception code and stored in the Exceptions table, along with all other access informations passed to the exit as input arguments.

- Furthermore, when an exception is detected, the exit can send a message to the z/OS console or to a designated z/OS userid. This can be done by calling the DBARSXI function as follows:

  address linkmvs "DBARSRXI SEND {userid|CONSOLE} messagetext"

15.3 Signalling a row fetch exception

- Row fetching is reported for tables defined with the argument **COUNT_ROWS** in the startup AUDITNAMES section.

- When a row fetch is reported, the variable "Operation" is set to FETCH and the number of rows already fetched is stored in the variable "Rows_processed".

- To signal a row fetch exception, the exit may use a DBARSRXI function and / or store the access in the DBARS Exception table.

- The exit should terminate with a returncode, as follows:

  - Returncode 0 : continue fetch monitoring
  - Returncode 8 : stop fetch monitoring
  - Returncode > 12 : stop fetch monitoring and store the access in the DBARS Exception table. The returncode will be stored in the EXCEPTION column of the Exception table.

15.4 Notes

- The Exceptions table is always a Db2 table, even when the Recorder is a VSAM cluster or a sequential dataset.
- Storing the access in the Exceptions table does **not** prevent the actual data access. In fact, the access will be recorded in both the Recorder and in the Exceptions table.
- DBARS provides the DBARSXR utility to inspect the Exceptions table in TSO.

15.5 Sample Exception Exit

```rexx
/* REXX */

/*
        If an access is from a remote server, the exit considers as exceptions:

        -   update attempts to the customer table
        -   accesses to "sensitive" customer numbers

        All exceptions are stored in the DBARS EXCEPTIONS table with the specific
        exception code.

*/

Arg Stamp ,
TCB ,
Db2_server ,
Db2_userid ,
zOS_userid ,
Correlation ,
Connection ,
LUWid ,
External_Server ,
External_Application ,
External_Workstation ,
RDI_Call_Type ,
RDI_Statement_Type ,
Table_Creator ,
Table_Name ,
Program_Name ,
Program_Section ,
SQLCODE ,
Rows_modified ,
Operation ,
Dynamic_access ,
Length_SQL_Statement ,
SQL_Statement ,
.

exception_code = 0
local_Db2 = "LOCDBVA"
statement = STORAGE(SQL_Statement,Length_SQL_Statement)
if Table_Name = "CUSTOMER_TABLE" then do
  if Db2_server <> local_Db2 then do
    if Operation <> "SELECT" then exception_code = 2040
    else do
        expression = "SENSITIVE <> 0"
        address linkmvs "DBARSRXI SCAN statement expression"
        if rc = 1 then exception_code = 2041
    end
  end
end
exit exception_code
```

15.6 Sample Row Fetch Exit

```rexx
/* REXX */

/* Parameters passed to the exit */

Arg Stamp ,
TCB ,
Db2_server ,
Db2_userid ,
zOS_userid ,
Correlation ,
Connection ,
LUWid ,
External_Server ,
External_Application ,
External_Workstation ,
RDI_Call_Type ,
RDI_Statement_Type ,
Table_Creator ,
Table_Name ,
Program_Name ,
Program_Section ,
SQLCODE ,
Rows_modified ,
Operation ,
Dynamic_access ,
Length_SQL_Statement ,
SQL_Statement ,
.
/*      Signal a fetch exception when more than 100 rows are fetched
        from the ACCOUNTS table */

 Limit = 100

 If Table_Name = 'ACCOUNTS' & Operation = 'FETCH' then do

    Rows = format(Rows_Processed)
    If Rows > Limit then do

      Table = strip(Table_Creator)"."strip(Table_Name)
      Msg = Rows "rows fetched from" Table
      address linkmvs "DBARSRXI SEND CONSOLE Msg"

      Msg = "by user" Db2_userid "and program" Program_Name
      address linkmvs "DBARSRXI SEND CONSOLE Msg"

      Msg = "Access stored in DBARS Exception table."
      address linkmvs "DBARSRXI SEND CONSOLE Msg"

      /* Write to DBARS Exception table exception type 1100 */

      Exit 1100
    End
 End
 Exit 0
```

16 DBARSGUI

16.1 Installing DBARSGUI

16.1.1 Installing on z/OS

The standard DBARS installation also installs the mainframe DBARSGUI components.

16.1.2 Installing on a workstation

The DBARS distribution file contains a folder, named DBARSGUI. The entire folder should be moved to a Windows folder accessible to the users that will invoke DBARSGUI from their workstation.

16.2 Configuring DBARSGUI

16.2.1 Configuring on z/OS

The DBARS distribution file contains in its JCL folder, the DBARSGUI SAMPJCL file, for starting the DBARSGUI server on z/OS.

EXEC PGM=DBARSGUI,PARM='TCPIP=...,PORT=...,MAXCLIENT=...'

Following optional server configuration options may be coded on the DBARSGUI PARM operand :

TCPIP
> Specify the name of your TCP/IP address space. **Default is TCPIP**.

PORT
> Specify the TCP/IP port to be used by DBARSGUI for communication with the workstations. **Default is 1946.**

MAXCLIENT
> Specify the maximum **concurrent** number of DBARSGUI clients allowed. **Default is 16.**

The distributed DBARSGUI JCL includes the DD statements for a VSAM Recorder. If your installation does not use a VSAM Recorder, these DD statements should be omitted.

16.2.2 Configuring on the workstation

The DBARS distribution folder DBARSGUI contains the DBARS.BAT which starts the DBARS application as follows:

start "DBARS" /min java -classpath . DBARS <IP_address> 1946 A

In the above command, replace:

- <IP_address> with the IP address of your z/OS system
- the default port 1946 (only if the default port was modified in the DBARSGUI PARM described above)
- replace **A** by **B** if DBARS uses a VSAM Recorder or by **C** if a sequential (BSAM) Recorder is used

16.3 Starting DBARSGUI

16.3.1 Starting the DBARSGUI server on z/OS

DBARSGUI can be started by submitting the customized DBARSGUI JCL.

Alternatively, DBARSGUI can be invoked as a started task. In this case, issue the following RACF commands to assign privileges to the started task:

```
RDEFINE STARTED DBARSGUI.DBARSGUI STDATA(USER(xxxx) GROUP(yyy))
SETROPTS RACLIST(STARTED) REFRESH
```

DBARSGUI issues TCP/IP Listen commands to wait for workstation requests. Note that there is no command provided to shutdown the DBARSGUI server. Use the z/OS cancel command instead.

16.3.2 Starting the DBARSGUI server on the workstation

Start the DBARS.BAT configured above. You will probably want to create a shortcut to DBARS.BAT on the Windows desktop.

16.4 Using DBARSGUI

16.4.1 Logon

When the DBARS BAT has been started, the user is requested to logon to the DBARSGUI server, using his Db2 Userid and Password.

Click the **Accesses** button to view selected audit events.
Click the **Exceptions** button to view audit events that were alerted or blocked.

16.4.2 Recorder Search Criteria

The next screen requests the user to enter the Recorder Search Criteria.

DBARS Report Criteria	
Quit	Submit	NextParm	PrevParm	Help

Recording date	
Recording time	
Table Creator	
Table Name	
Application Server	
DB2 Authorization ID	
z/OS User ID	
Correlation Name	
Connection Type	
Application Requester	
External Application	
External Workstation	
Program Name	
SQL Statement Type	
SQLCODE	
SQLROWS	
Dynamic SQL	
DBARS Region	DBARS

SQL Statement Expressions:

How to enter the search is described in the [Entering Report Criteria](#) section of this manual.

Alternatively, the user can press the Help button for a description of the columns and buttons to be used. The Help file is an HTML document.

Also, when moving the mouse over a button or a criteria field, a short descriptive text is displayed for the item.
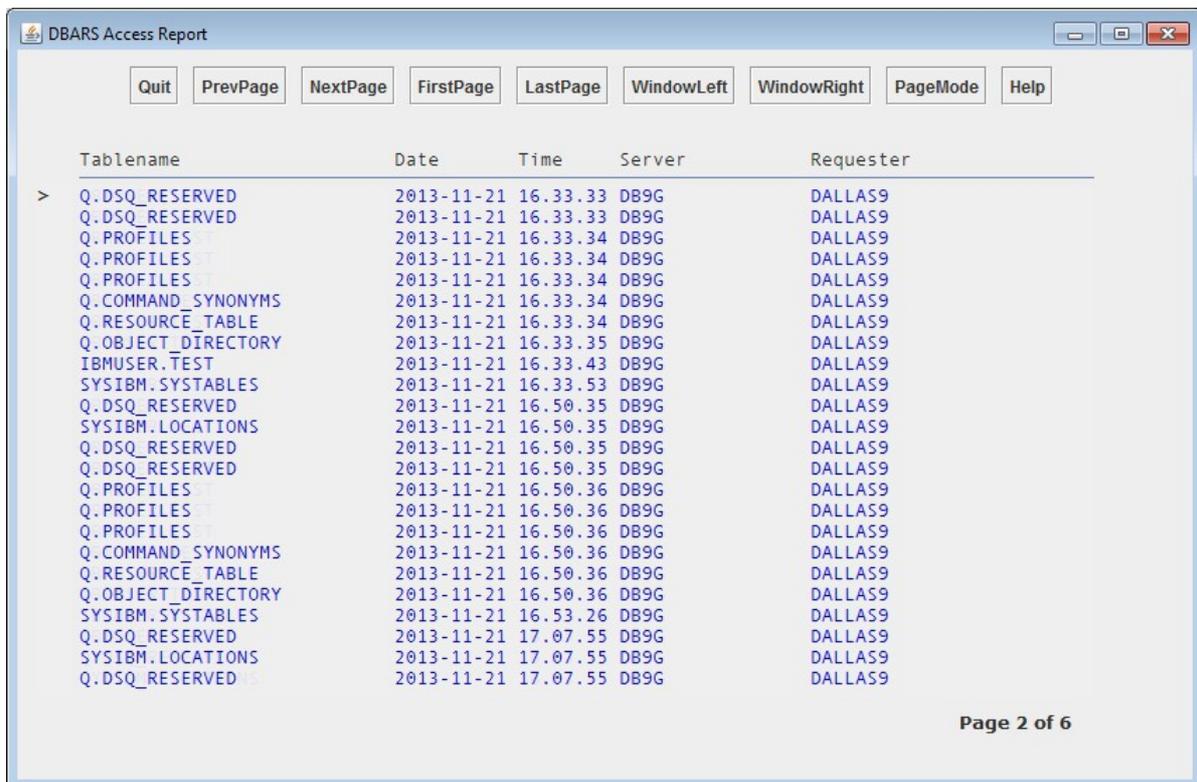
16.4.3 Exception Search Criteria

When an exception list has been requested at logon, enter the selection parameters.

DBARS Report Criteria

Quit	Submit	NextParm	PrevParm	Help

Field	Value
Recording date	
Recording time	
Table Creator	
Table Name	
Application Server	
DB2 Authorization ID	
z/OS User ID	
Correlation Name	
Connection Type	
Application Requester	
External Application	
External Workstation	
Program Name	
SQL Statement Type	
Exception	blocked
SQLROWS	
Dynamic SQL	
Order by	ACCESS_STAMP

How to enter the search is described in Chapter [10](#) of this manual.

Alternatively, the user can press the Help button for a description of the columns and buttons to be used. The Help file is an HTML document.

Also, when moving the mouse over a button or a criteria field, a short descriptive text is displayed for the item.

16.4.4 ListMode Access Report

The initial result of the Recorder Scan is a ListMode report shown above. The report shows all Recorder rows that meet the supplied Search Criteria.

The user can press the Help button for a description of the buttons to be used. The Help file is an HTML document.

Moving the mouse over a button will show the action initiated by the button. A number of keyboard keys can be used as a shortcut for the button. The PageDown and PageUp keys for instance can be used instead of the PrevPage and NextPage buttons.

The ListMode screen is too small to display all columns for a given Recorder row. The WindowLeft and WindowRight buttons are provided for sliding the screen to the left or the right.

16.4.5 PageMode Access Report

The PageMode Report will show all columns for a selected Recorder row on one screen. The PageMode report is invoked:

- By left clicking the mouse on a ListMode screenline.
- By selecting a line with the cursor up and down keys and pressing Enter or the PageMode button. (The "current report line" is indicated by a > sign at the begin of the line.)
- The NextPage, PrevPage, FirstPage and LastPage keys allow to browse the Access Report in page mode.
- The SQLText button provides a formatted display of the current SQL statement.
- The Copy button writes the current screen to the workstation file C:\DBARS_Hardcopy.doc.
- The Search key allows to search the current report as shown below.
- The Help button displays the function help file.

DBARS PageMode Access Report

| Quit | PrevPage | NextPage | FirstPage | LastPage | SQLText | Copy | Search | Help |

TableName	Q.DSQ_RESERVED
Date	2013-11-21
Server	DB9G
DB2 Userid	IBMUSER
Correlation	IBMUSER
Program	DSQFSDB7
Access	SELECT
SQLCODE	0
LUW_Id	CC4C57ED86C5
External Appl	

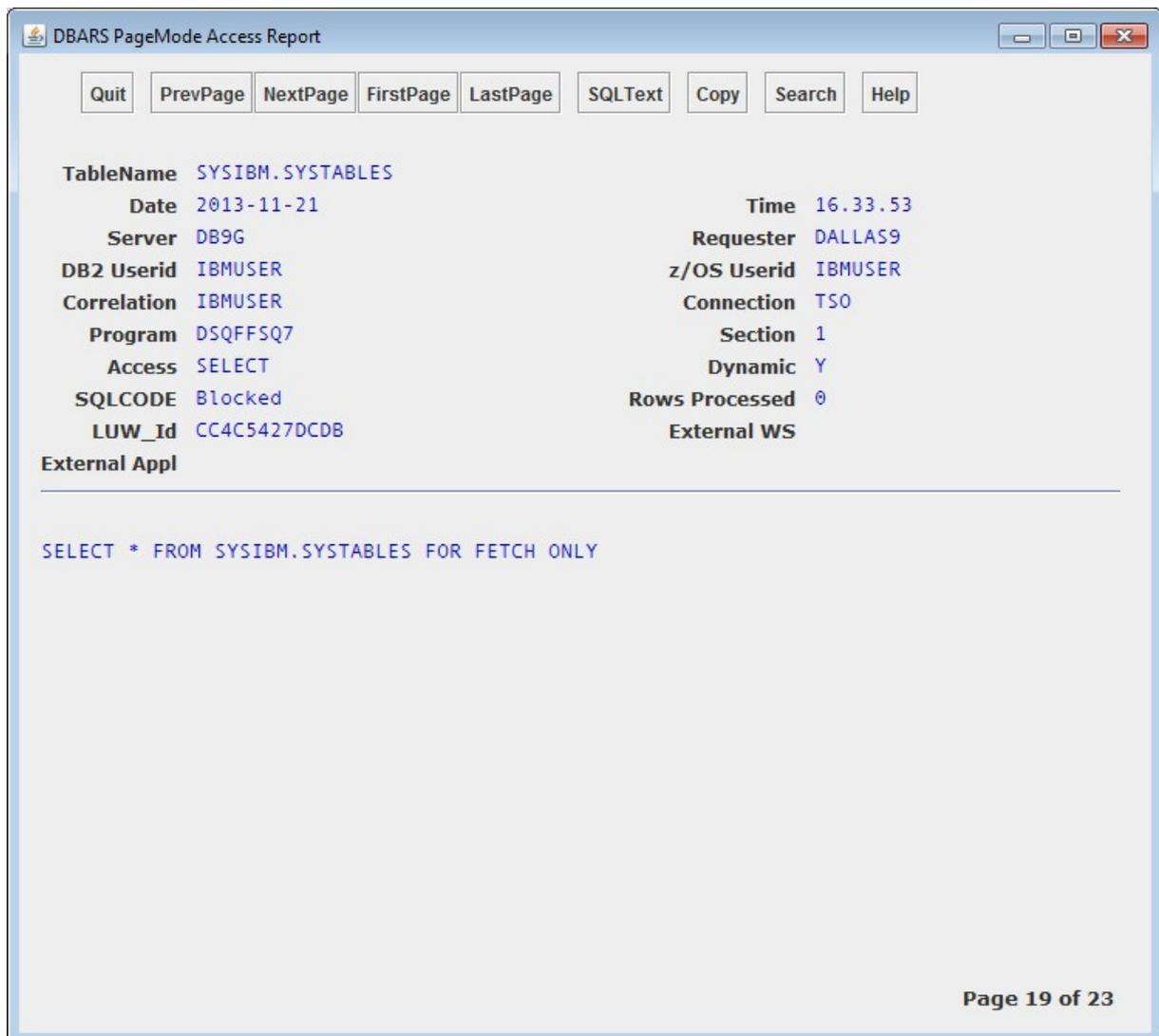|	
Time	16.50.35
Requester	DALLAS9
z/OS Userid	IBMUSER
Connection	TSO
Section	1
Dynamic	N
Rows Processed	0
External WS	

```
SELECT CCSID_ENCODING ( CURRENT APPLICATION ENCODING SCHEME ) , COUNT ( * ) ,
CURRENT OPTIMIZATION HINT , CURRENT FUNCTION PATH , CURRENT RULES , CURRENT
DEGREE , CURRENT LOCALE LC_CTYPE , CURRENT APPLICATION ENCODING SCHEME ,
CURRENT PRECISION , CURRENT SQLID INTO : H , : H , : H , : H , : H , : H , : H
, : H , : H , : H FROM Q . DSQ_RESERVED WHERE QMFREQ2 = 'L'
```

Page 37 of 129

16.4.6 Exception Pagemode Report

The Exception PageMode Report will show all columns for a selected row on one screen. The PageMode report is invoked:

- By left clicking the mouse on a ListMode screenline.
- By selecting a line with the cursor up and down keys and pressing Enter or the PageMode button. (The "current report line" is indicated by a > sign at the begin of the line.)
- The NextPage, PrevPage, FirstPage and LastPage keys allow to browse the Access Report in page mode.
- The SQLText button provides a formatted display of the current SQL statement.
- The Copy button writes the current screen to the workstation file C:\DBARS_Hardcopy.doc.
- The Search key allows to search the current report as shown below.
- The Help button displays the function help file.

16.4.7 Access Report Search

Search is performed by supplying:

- the name of a Recorder column as it appears on the PageMode report screen in **Search Column** (for example PROGRAM)
- the column value in **Search Value** (for example EMP01)

The Search Up and Search Down radio buttons provide for forward and backward Report search.

17 Transferring the DBARS Recorder to an external security system

The DBARSFTP component transfers a sequential Recorder to an external security system such as the Oracle AUDIT Vault and Database Firewall©, using FTP.

The DBARSFTP member in the DBARS JCLLIB provides a model for the DBARSFTP execution JCL.

```
//DBARSFTP JOB (...),'...',CLASS=A,MSGCLASS=A,REGION=0M,
//             USER=....,PASSWORD=....
//*
//FTP      EXEC PGM=DBARSFTP,PARM='ASN=DBARS,INTERVAL=30'
//STEPLIB  DD   DSN=DBARS.LOADLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=A
//SYSTSPRT DD   SYSOUT=A
//SYSUDUMP DD   SYSOUT=A
//FTPPARMS DD   DSN=DBARS.JCLLIB(FTPPARMS),DISP=SHR
```

17.1 DBARSFTP PARM Keywords

ASN=

Specify the name of the DBARS address space that owns the Recorder to be FTP-ed. DBARSFTP uses the address space name to compose the datasetname of the Recorder and to dynamically allocate it.

INTERVAL=

Specify the FTP interval in seconds. In the model JCL, an FTP run would be scheduled every 30 seconds.

SAM_PREFIX=

Specify the SAM_PREFIX as defined in section SAM_PREFIX

EXT=

If the file extension of the remote file is not specified in the FTPPARMS dataset, it can be specified here.

Specify EXT=XML if the transmitted DBARS recorder is in XML format. Alternatively, append .XML to the output filename on the PUT statement in FTPPARMS.

17.2   The **FTPPARMS member**

The member provides the FTP commands to be used by DBARSFTP.

- The first line specifies the IP-address of the primary FTP server. Multiple IP-addresses can be specified, separated by a blank. The secondary IP-addresses are used when the primary FTP server cannot be reached.
- The second line provides the login userid.
- The third line contains the user password. If the password is preceded by the string PASSWORD=xxx, DBARSFTP will encode the password on its first execution and store the encoded password in the FTPPARMS member.
- The following lines specify the remaining FTP commands. An ASCII - command is usually required.
- The PUT statement should provide the remote filename only. The name of the FTP-ed Recorder will be inserted by DBARSFTP. The remote filename may contain the variable symbols &SDATE and &STIME. These will be - replaced with the current date and time during FTP.
- The QUIT command should be the last line in the member.
- Lines starting with an asterisk are handled as comment lines.

  Example

  ```
  102.148.6.5
  USER01
  PASSWORD=xxxxx
  ASCII
  PUT DBARS.AUDIT.RECORDS.D&SDATE.T&STIME
  QUIT
  ```

17.3   **DBARSFTP Operation**

- DBARSFTP uses the z/OS FTP API. Consequently an OMVS segment must be defined for the application. Superuser privileges are not required.
- When an FTP run starts, DBARSFTP requests a Recorder switch from the DBARS address space. The closed Recorder is then sent to the FTP server.
- DBARSFTP uses the z/OS resource control services to prevent multiple concurrent accesses to the same FTP server.
- To stop DBARSFTP, issue the z/OS command F DBARSFTP,STOP.

17.4   **DBARSFTP Error handling**

When an error occurs during the FTP session, for instance because none of the FTP servers can be reached, DBARSFTP issues a DBARSFTP90 message and retries the FTP transfer until it is successful. The installation's automation procedures should intercept the DBARSFTP90 message to request a solution for the error situation.

18   Recorder Data Items

ACCESS_STAMP	TIMESTAMP	Timestamp of statement execution
INSERT_STAMP	TIMESTAMP	Timestamp of statement recording
TCB	INTEGER	User's TCB address
LOCATION	CHAR(16)	Name of the Db2 subsystem accessed
DB2_ID	CHAR(8)	Db2 userid performing the statement
MVS_ID	CHAR(8)	z/OS userid performing the statement
CORRELATION	CHAR(12)	Access association (such as jobname)
CONNECT_ID	CHAR(8)	TSO | BATCH | CICS | DDF
LUWID	CHAR(12)	LUW identification
EXT_SERVER	CHAR(16)	Name of external Db2 server (if DDF)
EXT_APPL	CHAR(32)	Name of external application (if DDF)
EXT_STATION	CHAR(18)	Name of workstation (if DDF)
TCREATOR	CHAR(8)	Table creator
TNAME	VARCHAR(128)	Tablename accessed
PROGRAM	CHAR(8)	Name of program issuing the SQL statement
STMNTNR	SMALLINT	Statement number within PROGRAM
CALL_TYPE	SMALLINT	Internal Db2 request identifier[7]
STMT_TYPE	SMALLINT	Internal Db2 request identifier
OPCODE	CHAR(8)	SELECT | UPDATE | DELETE | INSERT
SQLCODE	INTEGER	SQLCODE of the recorded statement
SQLROWS	INTEGER	Number of rows modified by the statement
DYNAMIC	CHAR(1)	Y if a dynamic, N if a static statement
STMNT_TEXT	LONG VARCHAR	Text of the recorded SQL statement

---

[7] These identifiers are described in the DSNXRDI member of the Db2 library DSNxxx.SDSNMACS.

19   Exceptions Table Structure

Column	Type	Description
ACCESS_STAMP	TIMESTAMP	Timestamp of statement execution
INSERT_STAMP	TIMESTAMP	Timestamp of statement recording
EXCEPTION	INTEGER	Exception code passed by the User Exit
LOCATION	CHAR(16)	Name of the Db2 subsystem accessed
DB2_ID	CHAR(8)	Db2 userid performing the statement
MVS_ID	CHAR(8)	z/OS userid performing the statement
CORRELATION	CHAR(12)	Access association (such as jobname)
CONNECT_ID	CHAR(8)	TSO | BATCH | CICS | DDF
LUWID	CHAR(12)	LUW identification
EXT_SERVER	CHAR(16)	Name of external Db2 server (if DDF)
EXT_APPL	CHAR(32)	Name of external application (if DDF)
EXT_STATION	CHAR(18)	Name of workstation (if DDF)
TCREATOR	CHAR(8)	Table creator
TNAME	VARCHAR(128)	Tablename accessed
PROGRAM	CHAR(8)	Name of program issuing the SQL statement
STMNTNR	SMALLINT	Statement number within the PROGRAM
CALL_TYPE	SMALLINT	Internal Db2 request identifier
STMT_TYPE	SMALLINT	Internal Db2 request identifier
OPCODE	CHAR(8)	SELECT | UPDATE | DELETE | INSERT
SQLCODE	INTEGER	SQLCODE of the recorded statement
SQLROWS	INTEGER	Number of rows modified by the statement
DYNAMIC	CHAR(1)	Y if a dynamic, N if a static statement
STMNT_TEXT	LONG VARCHAR	Text of the recorded SQL statement

20   Recorder Record Structure on the DBARS Archive

ACCESS_STAMP	CHAR(26)	Timestamp of statement execution
INSERT_STAMP	CHAR(26)	Timestamp of statement recording
TCB	INTEGER	User's TCB address
LOCATION	CHAR(16)	Name of the Db2 subsystem accessed
DB2_ID	CHAR(8)	Db2 userid performing the statement
MVS_ID	CHAR(8)	z/OS userid performing the statement
TCREATOR	CHAR(8)	Table creator
TNAME	CHAR(64)	Tablename accessed
CORRELATION	CHAR(12)	Access association (such as jobname)
PROGRAM	CHAR(8)	Name of program issuing the SQL statement
STMNTNR	SMALLINT	Statement number within PROGRAM
CALL_TYPE	SMALLINT	Internal Db2 request identifier
STMT_TYPE	SMALLINT	Internal Db2 request identifier
FILLER	CHAR(2)	
SQLCODE	INTEGER	SQLCODE of the recorded statement
SQLROWS	INTEGER	Number of rows modified by the statement
OPCODE	CHAR(8)	SELECT | UPDATE | DELETE | INSERT
DYNAMIC	CHAR(1)	Y if a dynamic, N if a static statement
CONNECT_ID	CHAR(8)	TSO | BATCH | CICS | DDF
LUWID	CHAR(12)	LUW identification
EXT_SERVER	CHAR(16)	Name of external Db2 server (if DDF)
EXT_APPL	CHAR(32)	Name of external application (if DDF)
EXT_STATION	CHAR(18)	Name of workstation (if DDF)
FILLER	CHAR(1)	
L_STMNT_TEXT	SMALLINT	Length of the recorded SQL statement
STMNT_TEXT	VARCHAR(30720)	Text of the recorded SQL statement

21   Recorder Record Structure for FTP

Field	Type	Description
ACCESS_STAMP	CHAR(26)	Timestamp of statement execution
INSERT_STAMP	CHAR(26)	Timestamp of statement recording
FILLER	CHAR(4)	
LOCATION	CHAR(16)	Name of the Db2 subsystem accessed
DB2_ID	CHAR(8)	Db2 userid performing the statement
MVS_ID	CHAR(8)	z/OS userid performing the statement
TCREATOR	CHAR(8)	Table creator
TNAME	CHAR(64)	Tablename accessed
CORRELATION	CHAR(12)	Access association (such as jobname)
PROGRAM	CHAR(8)	Name of program issuing the SQL statement
SQLCODE	CHAR(8)	SQLCODE of the recorded statement (left justified)
SQLROWS	CHAR(8)	Number of rows modified by the statement (left justified)
OPCODE	CHAR(8)	SELECT | UPDATE | DELETE | INSERT
DYNAMIC	CHAR(1)	Y if a dynamic, N if a static statement
CONNECT_ID	CHAR(8)	TSO | BATCH | CICS | DDF
LUWID	CHAR(12)	LUW identification
EXT_SERVER	CHAR(16)	Name of external Db2 server (if DDF)
EXT_APPL	CHAR(32)	Name of external application (if DDF)
EXT_STATION	CHAR(18)	Name of workstation (if DDF)
FILLER	CHAR(3)	
STMNT_TEXT	VARCHAR	Text of the recorded SQL statement

22 Recorder Record XML Structure

Tag	Name	Description
A	ACCESS_STAMP	Canonical timestamp of statement execution [8]
B	INSERT_STAMP	Canonical timestamp of statement recording
C	TCB	z/OS TCB address or zero
D	LOCATION	Name of the Db2 subsystem accessed
E	Db2_ID	Db2 userid performing the statement
F	MVS_ID	z/OS userid performing the statement
G	CORRELATION	Access association (such as jobname)
H	CONNECT_ID	TSO | BATCH | CICS | DDF
I	LUWID	LUW identification
J	EXT_SERVER	IP-address of external Db2 server if DDF Database name if local access
K	EXT_APPL	Name of external application (if DDF)
L	EXT_STATION	Name of workstation (if DDF)
M	TCREATOR	Table creator
N	TNAME	Tablename accessed
O	PROGRAM	Name of program issuing the SQL statement
P	SECTION	Statement number within program
Q	CTYPE	Internal Db2 command type
R	STYPE	Internal Db2 statement type
S	OPCODE	SELECT | UPDATE | DELETE | INSERT
T	SQLCODE	SQLCODE of the recorded statement
U	SQLROWS	Number of rows modified by the statement
V	DYNAMIC	Y if a dynamic, N if a static statement
W	STMNT_TEXT	Text of the recorded SQL statement

---

[8] The format of the canonical timestamp is as follows :
yyyy-mm-ddThh:mm:ss.xxxTZ
where   xxx = the number of milliseconds
TZ = signed timezone in hhmm format

23 DBARS Messages

23.1 Messages issued by the DBARS Main task

DBARS00I   Default Db2 subsystem is *nnnn*

The current DSNHDECP member points to the named Db2 subsystem or a Db2_SUBSYS parameter has been specified. DBARS will connect to this subsystem.

DBARS01I   Recording enabled for Server *nnn*

The message indicates that all audited accesses to the named Db2 server will be recorded.

DBARS02I   Waiting on Db2 subsystem

The message indicates that the Db2 server whose accesses must be recorded, is not active. DBARS waits until the Db2 server has been started. This message is more likely to appear when DBARS is started during z/OS IPL.

DBARS03I   *n* Db2 objects selected for recording.

The message shows the number of objects that will be recorded because they relate to audited tables.

DBARS04I   Current startup parameters using member xxxxxxxx:

The message shows the DBARS execution parameters obtained using the system defaults and the specifications of the DBARS startup member. This member resides in the DBARS.JCLLIB. The name of the member defaults to EXECPARM, but can be specified in a PARM operand of the EXEC DBARS JCL statement.

DBARS05I   Initiator loaded at nnnnnnnn

The message shows the load address of the DBARS Initiator function.

DBARS06I   Writer task active

The message indicates that the DBARS Writer function has been started.

DBARS07I   STOP request has been queued

The message is issued after a DBARS STOP command has been submitted. DBARS shutdown will start as soon as all recorded information has been written to the RECORDER table.

DBARS08I   DBARS shutdown started

The message is issued when DBARS shutdown starts.

DBARS10I	DBARS Version v.r ModLevel *nnnn*

Version number, release number and modification level of the DBARS software being loaded.

DBARS11A	*nnn* Address Space not found

The named Db2 database address space is not active. The name of this address space is supplied in the Db2_ADDRSPACE parameter of the DBARS EXECPARM, or is defaulted to <subsystem>DBM1.

Action:
Find the name of the Db2 database address space and supply it in the EXECPARM.

DBARS12A	RC=*nn* during license check

An error occurred when verifying the DBARS software license key. Another message will explain the license problem.

Action:
Contact SPR Software Support to obtain a valid license key.

DBARS13A	DBARS License key has expired

The DBARS evaluation period has expired.

DBARS14A	DBARS License key not found

The license key (member SCF in DBARS.JCLLIB) was not found.

DBARS15A	DBARS not licensed for CPU-id *nnnnn*

The license key was not built for the named CPU-id.

DBARS16A	DBARS License key has no valid content

The license key seems corrupted.

DBARS17I	DBARS License Info:
DBARS18I	Machine-type *nnnn* CPU-id *nnnnn*
DBARS19I	Customer Name *nnn*
DBARS20I	Software key expires on *yyyy-mm*-01

Messages DBARS17I to DBARS20I are informational and describe the current DBARS license key. Message DBARS20I is issued for evaluation licenses only.

DBARS21I	Maxlength of recorded SQL statement is nK

The message shows the maximum length of the SQL statements in the RECORDER table. This maximum depends on the bufferpool where the DBARS tablespace is located.

DBARS90A   Startup parameter syntax error

An syntax error has been detected in the DBARS startup
parameters. DBARS startup is terminated.

DBARS91A   Parameter no longer required

The specified startup parameter was valid in a previous DBARS
version, but is no longer required in the current release. The
parameter is accepted for compatibility reasons but effectively
ignored. DBARS startup continues.

DBARS92A   DBARS92A GETSHARED Failure RC=x Reason=xxxxxxxx
DBARS92A SHAREMEMOBJ Failure RC=x Reason=xxxxxxxx
DBARS92A DETACH Failure RC=x Reason=xxxxxxxx

Ensure that MEMLIMIT has been specified in the DBARS startup JCL.
If MEMLIMIT is specified, call for software support.
A description of the Reason code can be found in the IBM Manual
"System Codes" under system code DC2.

DBARS97A   DB2 component not recognized

An error occurred when installing the DBARS to Db2 interface.

Contact SPR software support.

DBARS98A   DBARS already active

An attempt was made to start DBARS more than once in the same
z/OS system.

DBARS99A   APF Authorization failure

The DBARS.LOADLIB is not an APF-authorized library.

23.2 Messages issued by the DBARS Initiator

DBARSI001I   DBARS Initiator is initializing

Initialization of the DBARS Initiator in the DBM1 address space is in progress.

DBARSI997A   DBARS Initiator recovery has been entered

A cancel condition has been detected during DBARS processing. Problem determination information has been stored to the DBARS trace buffer. DBARS processing continues.

To print the trace buffers, stop DBARS and contact DBARS software support.

DBARSI998A   RC xx requesting Recorder storage

Returncode xx was returned by the DBARS storage manager when attempting to obtain storage from the 64-bit buffer area.

If the RC equals 8, increase the MEMLIMIT value and restart DBARS.

DBARSI999A   DBARS Initiator disabled.

A severe error condition exists which does not allow DBARS recording to continue. Another message will precede and explain the error condition.

23.3 Messages issued by the DBARS Queue Manager

DBARSM01A   Recorder queue space full. Recording disabled until queue space becomes available..

The recorder queue space in the DBARS address space has no more free space after executing the defined number of queue extensions. DBARS recording is temporarily disabled until queue space becomes available again. Queue space is released when queue entries are written to the DBARS Recorder table or cluster.

Action
Enlarge the Q2_SIZE startup parameter.

DBARSM03I   Recorder Queue Statistics

Recorder queue statistics are displayed at DBARS shutdown or as a reply to the DBARS STATS command.

DBARSM04I   Requests queued: *n*

Total number of SQL statements stored into the recorder queue. The counter is reset at begin of day.

DBARSM05I   Requests processed: *n*

Total number of SQL statements retrieved from the recorder queue and written to the DBARS table. The counter is reset at begin of day.

DBARSM06I   Times recorder space full: *n*

Total number of times no space was found in the recorder queue. If this value is greater than zero, recorded events have been lost.

DBARSM07I   Pages in free queue: *n*

Number of 128 K pages currently in the free queue of the recorder space.

DBARSM08I   Pages in request queue: *n*

Number of 128 K pages currently in the "in-use" queue of the recorder space.

DBARSM09I   Max pages in request queue: *n*

Maximum number of 128 K pages ever in the "in-use" queue of the recorder space.

DBARSM14I   Recorder Statistics

The following messages are produced in reply to the RECSTATS command. They show statistical information about the DBARS Recorder.

DBARSM15I   Primary | Secondary Recorder currently in use

Shows the Recorder that is currently receiving output.

DBARSM16I	Recorder table statistics:

The following statistics are provided when the Recorder is a Db2 table.

- Recorder rows inserted in session
- Average SQL INSERT duration in milliseconds
- Maximum SQL INSERT duration in milliseconds

Each Recorder record is inserted as a single Db2 row.

DBARSM17I	Recorder cluster statistics:

The following statistics are provided when the Recorder is a VSAM cluster.

Notes

- Recorder blocks written   (1)
- Recorder blocks written in session   (2)
- Length of a Recorder block   (3)
- Average records in Recorder block   (4)
- Allocated VSAM cluster extents
- Ending VSAM cluster RBA   (5)
- High-allocated VSAM cluster RBA   (6)
- Available cluster space in bytes   (7)
- Available cluster space percentage
- Average IO duration in milliseconds   (8)
- Maximum IO duration in milliseconds

Notes

1. The total number of VSAM records (Recorder blocks) in the cluster.
2. The number of VSAM records (Recorder blocks) written during the current DBARS session.
3. The length of a VSAM record (Recorder block).
4. Individual Recorder records are blocked by DBARS before being written to the cluster as a block. The length of a Recorder record depends on the length of the recorded SQL statement.
5. The highest RBA defined for the Recorder cluster.
6. The highest RBA used for the Recorder cluster.
7. The available Recorder cluster space, in bytes and as a percentage.
8. The duration of writing the Recorder blocks to the VSAM cluster. Writing a block stores multiple Recorder records, as shown under (4).

DBARSM18I	Recorder cluster not opened yet

No records have been written yet to the Recorder. Statistics are not available.

23.4 Messages issued by the DBARS Writer task

DBARSW00A   SQL error during CONNECT to *server*

An SQLCODE was returned when DBARS connects to *server*. The SQLCA is formatted upon the console and DBARS terminates.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support .

DBARSW01A   SQL error getting audited table list

An SQLCODE was returned when DBARS obtains the list of audited tables. The SQLCA is formatted upon the console and DBARS terminates.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support .

DBARSW02A   SQL error getting audited view list

An SQLCODE was returned when DBARS obtains the list of views depending on audited tables. The SQLCA is formatted upon the console and DBARS terminates.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW03A   SQL error inserting into DBARS.RECORDER

An SQLCODE was returned when DBARS inserts an access row into the RECORDER table. The SQLCA is formatted upon the console.

If the SQLCODE shows a recoverable error (such as lack of space) on the primary recorder, DBARS automatically switches to the auxiliary recorder table AUX_RECORDER. If the SQLCODE persists on the AUX_RECORDER as well, DBARS abends.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW04I   Switching to auxiliary recorder

A recoverable error has occurred on the RECORDER table and the auxiliary recorder has been activated.

Action
Resolve the error on the primary recorder and issue the SWITCH_REC command to activate the primary recorder again.

DBARSW05I	Switched to alternate recorder

The alternate Recorder is active.

DBARSW06I	Switched to primary recorder

The primary Recorder is active.

DBARSW07A	SQL error deleting from DBARS.RECORDER

A SWITCH_REC command was issued and an SQLCODE was returned when deleting rows from the primary recorder. The SQLCA is formatted upon the console and DBARS terminates.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW08A	SQL error deleting from DBARS.AUX_RECORDER

A SWITCH_REC command was issued and an SQLCODE was returned when deleting the rows from the auxiliary recorder. The SQLCA is formatted upon the console and DBARS terminates.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW09A	SQL error getting audited object_ids

An SQLCODE was returned when DBARS obtains the list of audited object ID's. The SQLCA is formatted upon the console and DBARS terminates.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW10A	SQLCODE xxx selecting from DBARS_RECID

An SQLCODE was returned when DBARS obtains the ID of the current recorder. The SQLCA is formatted upon the console and DBARS terminates.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW11A	SQLCODE xxx updating DBARS_RECID

An SQLCODE was returned when DBARS updates the ID of the current recorder. The SQLCA is formatted upon the console and DBARS terminates.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW12A	VSAMRC xxxx writing to RECORDx

A VSAM error occurred when writing to the primary (RECORDA) or the auxiliary (RECORDB) recorder.

Action
Examine the VSAM PUT returncode and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW13I	VSAM Recorder cluster x opened

The active VSAM Recorder cluster (A or B) is shown under "x".

DBARSW14A	Error xxxx opening Recorder y

The Recorder (A or B) could not be opened. The decimal open error code is shown in "xxxx".

DBARS abends.

DBARSW15A	VSAM error xxxx opening Recorder Index y

The VSAM Recorder cluster (A or B) could not be opened. The decimal open error code is shown in "xxxx".

DBARS abends.

DBARSW16A	VSAMRC xxxx writing to RECORDBx

A VSAM error occurred when writing to the primary (RECORDAX) or the auxiliary (RECORDBX) recorder index.

Action
Examine the VSAM PUT returncode and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW17A	SQL error inserting into DBARS.EXCEPTIONS

An SQLCODE was returned when DBARS inserts a row into the EXCEPTIONS table. The SQLCA is formatted upon the console.

Action
Examine the SQLCODE and perform the necessary correction. If needed, contact SPR Software Support.

DBARSW18A	Starting Recorder Archive procedure xxxx
	A full condition has been detected for the current Recorder and the ARCHIVE_PROC statement was specified in the DBARS startup parameters. The user provided procedure named on the ARCHIVE_PROC statement, is now started.
DBARSW19A	SQLCODE xxx creating temporary tables
	A CREATE GLOBAL TEMPORARY TABLE statement failed. DBARS is terminated.
DBARSW20A	Syntax error in AUDIT statement
	A syntax error was detected in an AUDITNAMES table specification in the DBARS EXECPARM dataset. DBARS is terminated.
DBARSW21I	Tablenames recorded by DBARS:
	Output of the AUDITTAB command.
DBARSW22I	tablecreator.tablename
	Output of the AUDITTAB command.
DBARSW23I	Emptying Recorder queue before shutdown ...
	A DBARS STOP command has been submitted. Before terminating, the DBARS Writer task will write the remaining Recorder queue entries to the Recorder table or cluster.
DBARSW24I	Abend recovery completed.
	A program check interruption has occurred while the Write is processing a recording request.
	The Writer abend routines display diagnostic information on the console and create a formatted dump in the z/OS spool. Processing is then resumed.
	Action Forward the system log and the formatted dump to SPR software support.
DBARSW25A	Invalid EXCLUDE statement ignored.
	An invalid EXCLUDE statement was found following the ALLTABLES specification in the EXECPARMS dataset. The statement is ignored. Processing continues with the next statement.
DBARSW26I	Connecting the Initiator in Subsystem xxxx
	The DBARS Writer connects to the Initiator in the Db2 subsystem.
DBARSW27I	Connection to the Initiator established.
	The DBARS Writer is now connected to the Initiator.

DBARSW28I   Q2 space map re-ordered. RC=n

                   To speed up processing, the Q2 space map has been re-ordered.
                   The RC (1, 2, 3) indicates the amount of space re-ordered.

DBARSW29I   End-of-extent processed by DBARS

                   All extents of the current sequential Recorder were exhausted and a
                   B37 z/OS abend occurred. The abend was intercepted by DBARS
                   and an automatic Recorder switch is performed.

23.5 Messages issued during Interactive Access Reporting

DBARSIR01A   A terminal with at least 32 lines is required

The DBARSI program requires a terminal with 32 lines at least.

DBARSIR02A   SQLCODE *nnn* evaluating "*xxx*"'

When evaluating the selection expression *xxx*, SQLCODE *nnn* was returned.

Action
Ensure that the selection expression obeys Db2 syntax.

DBARSIR03A   PF key unassigned or function not available

You pressed a program function key other than the ones listed in the screen footer.

DBARSIR04A   Column reformat code is invalid

When reformatting the screen, a character other than + - < > or 2-9 was entered. You may want to display the help screen for the reformat function.

DBARSIR05A   No more than one ordering column allowed

When reformatting the screen, more than one sort column was indicated.

DBARSIR06A   Sort error code *nnn*

An error occurred during the sort requested from the reformat function.

Action
Retry the sort with a larger virtual storage. If the error persists, contact SPR Software Support.

23.6 Messages issued during Exception Reporting

DBARSXR01A   A terminal with at least 32 lines is required

The DBARSXR program requires a terminal with 32 lines at least.

DBARSXR02A   SQLCODE *nnn* evaluating "*xxx*"'

When evaluating the selection expression *xxx*, SQLCODE *nnn* was returned.

Action
Ensure that the selection expression obeys Db2 syntax.

DBARSXR03A   PF key unassigned or function not available

You pressed a program function key other than the ones listed in the screen footer.

DBARSXR04A   Column reformat code is invalid

When reformatting the screen, a character other than + - < > or 2-9 was entered. You may want to display the help screen for the reformat function.

DBARSXR05A   No more than one ordering column allowed

When reformatting the screen, more than one sort column was indicated.

DBARSXR06A   Sort error code *nnn*

An error occurred during the sort requested from the reformat function.

Action
Retry the sort with a larger virtual storage. If the error persists, contact SPR Software Support.

23.7 Messages issued by the DBARS Archiving program

DBARSA01A   Missing RESTART parameter

The Archive program detected that a previous archive run did not complete normally. In this case, the user must supply the RESTART or the NORESTART PARM argument.

Action
Refer to the description of the archive control parameters.

DBARSA02A   Invalid PARM argument *nnn*

Invalid syntax on the PARM operand of the EXEC DBARSA JCL statement.

Action
Refer to the description of the archive control parameters.

DBARSA03I   Archiving from DBARS.RECORDER *nnn*

Informatory message that shows the SQL statement built by the DBARSA program to select the rows to be archived.

DBARSA04I   *n* rows deleted from DBARS.RECORDER

Informatory message showing the number of rows written to the archive and deleted from the Recorder table.

DBARSA05A   DBARS_RECID table cannot be accessed.

The SEQREC archiving option was specified and an attempt was made to retrieve the current recorder cluster ID (A or B) from the DBARS_RECID table. This attempt failed.

DBARSA06A   VSAMRC x reading from RECORDA/RECORDB

A VSAM read failure occurred when reading from the recorder cluster. The VSAM RPL feedback field is displayed under "x".

Action
Examine the VSAM GET returncode and perform the necessary correction. If needed, contact SPR Software Support.

DBARSA07I   Archiving cluster xxx

The SEQREC archiving option has been specified. The message shows the name of the recorder cluster being archived as DBARS.RECORDER.A or DBARS.RECORDER.B

DBARSA08I   Submitting SWITCH_REC command to DBARS.

The SEQREC archiving option has been specified. Before starting the archive, a request is sent to the DBARS address space for switching to the alternate recorder cluster.

23.8 Messages issued during Archive Reporting

DBARSAR01A   RC *nn* opening ARCHIVE dataset

Returncode nn was received from z/OS when opening the ARCHIVE dataset.

Action
Review the ARCHIVE DD statement.

DBARSAR91A	Parameter name too long
DBARSAR92A	Parameter name invalid
DBARSAR93A	Operation code invalid
DBARSAR94A	Invalid alpha expression
DBARSAR95A	Invalid numeric expression
DBARSAR96A	Expression has no valid syntax
DBARSAR97A	OUTTABLE statement incorrect
DBARSAR98A	OUTLINES statement incorrect

Messages DBARSAR91A through DBARSAR98A signal a syntax error in the selection criteria supplied to the Archive Reporting program DBARSAR.

Action
Refer to Access Reporting from an Archive for a description of the DBARSAR parameter syntax.

23.9 Messages issued during batch Reporting from a VSAM Recorder

DBARSRR97A   OUTTABLE statement incorrect
DBARSRR98A   OUTLINES statement incorrect
DBARSRR99A   DBARS_JN statement incorrect

23.10 Messages issued during a Recorder FTP session

DBARSFTP00 DBARSFTP ended.

> The FTP program has terminated, usually after a Modify STOP request.

DBARSFTP01 Sending <Recorder_name>

> Message issued when an FTP run starts.

DBARSFTP02 <Recorder_name> sent

> Message issued when an FTP run ends normally.

DBARSFTP03 nnn bytes sent at nnn KB/SEC

> Statistical message issued at the normal end of an FTP run.

DBARSFTP04 FTP transfer failed

> The FTP transfer was not successful. One or more DBARSFTP90 messages should preceed.

DBARSFTP05 DBARS_RECID table cannot be accessed
DBARSFTP05 SQLCODE is xxx

> A Db2 error occurred when DBARSFTP attempts to access the DBARS_RECID table.

DBARSFTP07 Target FTP-server is xxxxxxxxx

> FTP transfer to the designated FTP server is now starting.

DBARSFTP08 Locking FTP-server xxxxxxxxx

> DBARSFTP issues a cross-system z/OS enqueue for the designated FTP server and waits, if the server is in use by another DBARSFTP instance.

DBARSFTP09 FTP-server unlocked

> A z/OS dequeue has been issued for the designated FTP server.

DBARSFTP80 RDJFCB failed. RC=xxx

> An internal DBARSFTP error has occurred. Request software support.

DBARSFTP90 FTP OPEN failure. RC=xxx, Reason=xxx

> The FTP OPEN command is invalid. Verify that the FTP server is operational.

DBARSFTP90 FTP USER failure. RC=xxx, Reason=xxx

> The FTP USER command is invalid. Check the FTPPARMS library member.

DBARSFTP90 FTP PASS failure. RC=xxx, Reason=xxx

> The FTP PASS command is invalid. Check the FTPPARMS library member.

DBARSFTP90 FTP xxx command failure. RC=xxx, Reason=xxx

> The named FTP command is invalid. Check the FTPPARMS library member.

DBARSFTP90 API INIT failure. RC=xxx, Reason=xxx
DBARSFTP90 API GETL failure. RC=xxx
DBARSFTP90 API GETL failure. RC=xxx

> The above messages indicate an internal DBARSFTP error. Contact software support.

DBARSFTP98 Invalid PARM syntax

> Check the PARM keywords on the EXEC PGM=DBARSFTP statement.

DBARSFTP99 FTPPARMS open failure <RC>

> The FTPPARMS library member cannot be opened. Check the DBARSFTP JCL.