DBARS

Installation Guide

Version 4.4

1. Tasks to perform before installing DBARS

The pre-installation tasks should be executed on the mainframe.

Software Prerequisites

DBARS requires:

- Current z/OS version

- Db2 Server for z/OS Version 9, 10, 11, 12 or 13

1.1. 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 DBARS Writer executes in the DBARS address space. It records all intercepted Db2 accesses into the DBARS Recorder. The Writer requires following Db2 privileges:

- access the Db2 system catalogs
- write to the DBARS tables
- execute the plans DBARS and DBARSW

The DBARS runtime requirements are met when the Writer has SYSADM authority.
If it is impossible to grant SYSADM, explicit grants should be issued.

If DBARS is requested to record Db2 BIND, Db2 UTIL or Db2 commands, the MONITOR1 privilege is also required for the DBARS Writer, since these requests are recorded using an IFI trace.

1.2. Define the DBARS Libraries

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

Please note that the default library names may be prefixed with a high level qualifier, if your naming conventions require it. The high level qualifier can be specified during FTP of the DBARS components to the mainframe libraries.

Library Name	Record Format and Length	Block Size	Primary Extent	Secondary Extent (in 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

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

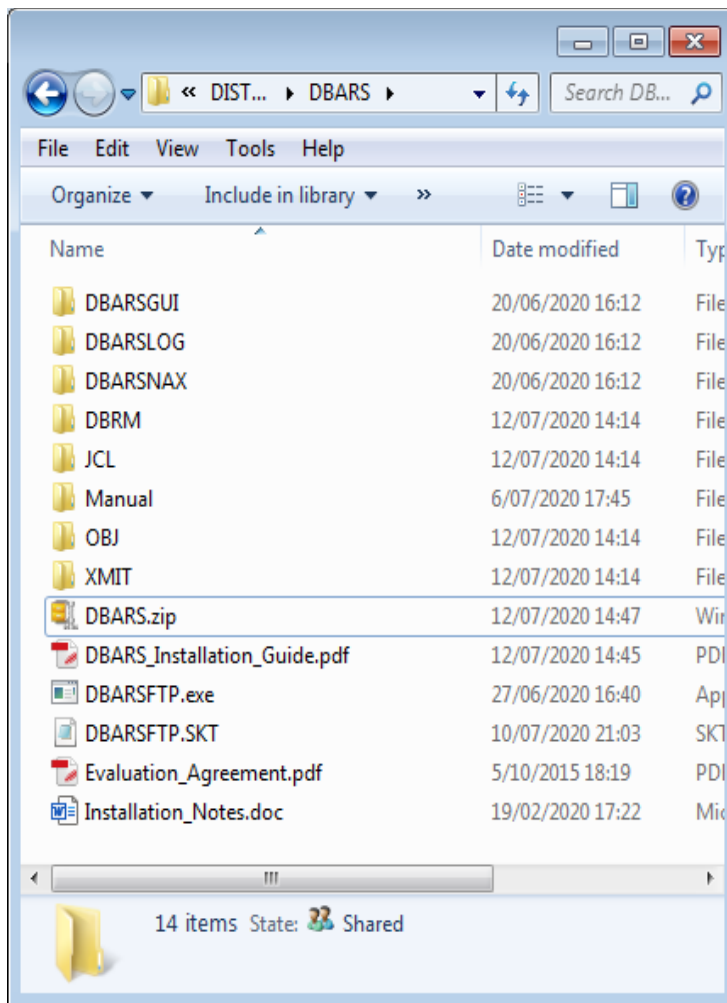Following z/OS command may be used to activate PROGxx before IPL:

    SETPROG APF,ADD,DSNAME=<your_hlq.>DBARS.LOADLIB,VOLUME=XXXXXX

Note

If you must add a high level qualifier to the default DBARS library names, note that you can specify the HLQ when uploading DBARS with the DBARSFTP utility.

See .

2. Uploading DBARS to z/OS

The DBARS product is electronically delivered as the file DBARS.ZIP. Process the ZIP as follows:

- Store the ZIP to a PC folder, for example C:\DBARS
- Extract the ZIP file to the same folder.
- The DBARS folder should now have the structure shown above.

The folders DBRM, JCL and OBJ contain the DBARS components for transfer to the mainframe.

The XMIT folder contains the DBARS libraries in TRANSMIT format.

The Manual folder provides the DBARS documentation files.

The DBARSGUI, DBARSLOG and DBARSNAX folders contain optional components that partly execute on Windows.

DBARSGUI provides a modern GUI interface to inspect the DBARS Recorder dataset. Installation and use of DBARSGUI is described in the manuals DBARSGUI Installation Guide and DBARSGUI Users Guide.

The DBARSNAX feature allows to send emails when access exceptions occur. The feature is described in chapter 7 of the DBARS Users Guide.

The DBARSLOG feature allows to write to the Windows Event Log when an access exception is detected. The feature is described in chapter 10 of the DBARS Users Guide.

DBARS can be uploaded to z/OS:

- using the DBARSFTP utility described in Uploading DBARS using DBARSFTP
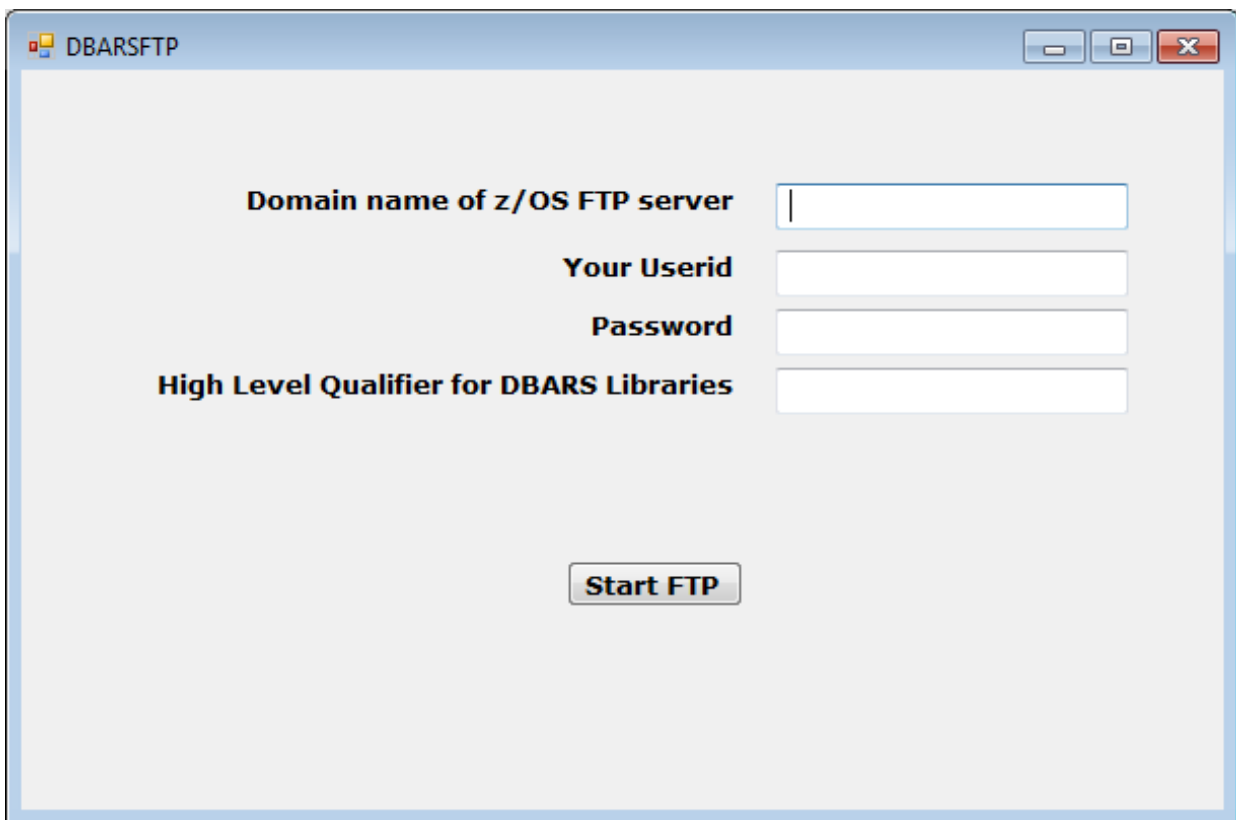- using TSO TRANSMIT as described in Uploading DBARS using TSO RECEIVE

3. Uploading DBARS using DBARSFTP

Click on the **DBARSFTP.exe** to start FTP.

DBARSFTP will transfer the DBARS components from the PC folders to following, previously created z/OS libraries:

- DBARS.OBJLIB
- DBARS.DBRMLIB
- DBARS.JCLLIB

If the naming conventions of your organization do not allow these default library names, enter the high level qualifier for the DBARS libraries on the screen below.

On the DBARSFTP screen enter:

- the IP address of the target z/OS system
- your FTP userid
- your FTP password
- optionally, the high level qualifier for the DBARS library names (without trailing dot)

Eventual errors during the FTP session will be stored in the DBARSFTP.LOG file and shown on the screen.

4. Uploading DBARS using TSO RECEIVE

The XMIT folder contains the DBARS components to transmit, namely:

DBRMLIB.XMIT   Db2 database request modules
JCLLIB.XMIT   DBARS JCL members for installation and samples
OBJLIB.XMIT   DBARS object members

These files are fixed blocked with recordlength 80 and blocksize 8000.

Transfer them to z/OS in binary mode.

RECEIVE each of them in TSO.

If a high level qualifier is used for the DBARS libraries, use the 'DA' TSO RECEIVE parameter.

5. Installing DBARS on the Mainframe

5.1. The DBARS Recorder

DBARS logs the intercepted Db2 accesses into its **Recorder**, using dual logging. When the primary Recorder is full, the secondary Recorder is automatically activated. Recorder switching also occurs when the Recorder is archived. Dual recording ensures that no intercepted accesses are lost.

The DBARS Recorder can be defined as:

- a VSAM cluster, which is the recommended option
- a BSAM dataset
- a Db2 table, an option that should be used on very lightly loaded Db2 systems only

5.2. Building DBARS

From a TSO session, submit following jobstreams from the DBARS JCLLIB to z/OS **in the specified order**:

DBARSTAB to create the DBARS tables
DBARSLNK to link the DBARS modules into the DBARS.LOADLIB
DBARSBND to bind the DBARS plans

- Before submitting, complete the JOB statement in each JCL stream.
- Verify that the SET statements at the begin of each JCL refer to the correct Db2 and DBARS libraries.
- The DD statements for Db2 are preset in the JCL's to Db2 Version 12. Modify if needed.

5.2.1 DBARSTAB

Must be executed in each Db2 subsystem. Complete the JOB statement, if not already done during FTP.

Verify the following Db2 statements and modify them to meet your installation requirements and standards:

```
CREATE DATABASE DBARS
CREATE TABLESPACE DBARS          for the DBARS.RECORDER table
CREATE TABLESPACE DBARSAUX       for the DBARS.AUX_RECORDER table
CREATE TABLESPACE EXCEPT         for the DBARS.EXCEPTIONS table
CREATE TABLESPACE TRANSFER       for the DBARS.TRANSFER table
CREATE TABLESPACE ARCHIVE        for the DBARS.ARCHIVE_CATALOG table
```

DBARS.RECORDER tables

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 buffer pools, 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.

If the DBARS Recorder will be defined as a VSAM or BSAM dataset, create the DBARS.RECORDER and DBARS.AUX_RECORDER tables with minimal space allocations.

Please note that using a Db2 table for the Recorder is not recommended, unless the monitored Db2 system is lightly loaded.

DBARS.EXCEPTION table

Stores Db2 accesses that are blocked or alerted. Since the SQL text is also stored in the table, BUFFERPOOL BP32K is recommended.

DBARS.TRANSFER table

Is used only when FTP-ing recorded Db2 accesses to an external security system. Create the tablespace with minimal space, if the Transfer table will not be used.

DBARS.ARCHIVE_CATALOG table

Whenever the DBARS Recorder is archived, an entry is made into the ARCHIVE_CATALOG table. The catalog row contains the dataset-name of the archive volume and the data range for the DBARS audit data on this volume.

The Archive Catalog is used when an archive must be accessed interactively on TSO or using the DBARSGUI interface.

The length of a catalog row is about 100 bytes. There will be a row for each DBARS Archive dataset newly allocated. Using DISP=MOD during archiving will update the row for the existing archive volume.

5.2.2 DBARSBND

Must be executed in each z/OS system that will run DBARS. Complete the JOB statement and verify the Db2 and DBARS library definitions.

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.

5.2.3 DBARSLNK

Must be executed in each z/OS system that will run DBARS. Complete the JOB statement and verify the Db2 and DBARS library definitions.

5.3. Create a VSAM Recorder

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

- Complete the JOB statement.
- If required by your installation, apply a high level qualifier to the cluster names
  - DBARS.RECORDER.A
  - DBARS.RECORDER.B
  - DBARS.RECORDER.A.INDEX
  - DBARS.RECORDER.B.INDEX
- Specify the VOLUME parameter in the DEFINE CLUSTER statements.
- Verify space defined by the RECORDS parameter.

The DBARSREC job also defines the Recorder Index clusters. These KSDS clusters are provided to allow faster scan of the Recorder.

Note

DBARS performs its own record blocking before writing to the Recorder.

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.

5.4. Create a sequential (BSAM) Recorder

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 SAM_PREFIX as shown in the DBARS
startup parameters.

○ 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.

Note

DBARS performs its own record blocking before writing to the Recorder.

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.

5.5. 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

5.6. Db2 version or release migration

When a new Db2 version or release has been installed, DBARS will automatically recognize it and adapt its processing accordingly.

The DBARS plans should be rebound on the newly installed Db2. This is achieved by submitting the DBARSBND JCL job, which is part of the DBARS distribution material.

6. Starting DBARS

DBARS retrieves its execution parameters from the EXECPARM member in DBARS.JCLLIB.

A default EXECPARM is provided with the distribution. The default should be sufficient to initially start DBARS. You should update the DB2_SUBSYS parameter and specify the Db2 tables to be monitored after "AUDITNAMES". For details, please refer to the DBARS User Guide chapter 2 "DBARS setup".

DBARS EXECPARM

```
DB2_SUBSYS   DBxG                 /* DB2 subsystem to audit        */
Q2_SIZE      4096                 /* Size of audit queue in MB     */
SEQREC                            /* Use a VSAM Recorder           */
OPTIONS      NOMASK               /* MASK sets hostvar masking off */
RECORD       DDL UTIL BIND CMND   /* Additional recording options  */
AUDITNAMES                        /* Names of tables to be audited */
<names of audited tables>
```

Although DBARS can be started using normal JCL, a started task is normally used to bring DBARS up.

A sample startup task is included here.

```
//DBARS     PROC
//MAIN      EXEC PGM=DBARS,MEMLIMIT=4224M,PARM='DBARS.JCLLIB(EXECPARM)'
//STEPLIB  DD DSN=DBARS.LOADLIB,DISP=SHR
//         DD DSN=DSNx10.DBxG.SDSNEXIT,DISP=SHR
//         DD DSN=DSNx10.SDSNLOAD,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
//SYSUDUMP DD SYSOUT=A
//SYSTSPRT DD SYSOUT=A
```

For details regarding the MEMLIMIT parameter, please refer to chapter 3 of the DBARS User Guide under "DBARS Storage requirements".

7. JCL samples provided in the DBARS distribution

DBARS.SAMPJCL   DBARS started task
DBARSA.SAMPJCL   Archive the DBARS Recorder manually
DBARSAR.SAMPJCL   Report archived accesses
DBARSARC.SAMPJCL   Automated DBARS archiving
DBARSBND.SAMPJCL   Bind DBARS plans
DBARSCAT.SAMPJCL   Create DBARS archive as a Db2 table
DBARSCRS.SAMPJCL   Start the centralized Recorder service
DBARSCSR.SAMPJCL   Create a BSAM Recorder
DBARSEXT.SAMP   Sample DBARS user exit
DBARSFTP.SAMPJCL   FTP the Recorder to an external security system
DBARSGUI.SAMPJCL   Start the DBARS Graphical User Interface
DBARSLNK.SAMPJCL   Link the DBARS objects into the DBARS LOADLIB
DBARSLOG.SAMPJCL   Send access exceptions to the Windows Event Log
DBARSNAX.SAMPJCL   Send access exceptions as an email email
DBARSREC.SAMPJCL   Create a VSAM Recorder
DBARSRR.SAMPJCL   Print selected Recorder entries
DBARSRRS.EXEC   Scan a BSAM Recorder in TSO
DBARSRRX.EXEC   Scan a VSAM Recorder in TSO
DBARSTAB.SAMPJCL   Create the DBARS tables
EXECPARM.SAMP   Sample DBARS startup parameters
FTPPARMS.SAMP   Sample FTP parameters for DBARSFTP
REDUCEP.SAMP   Sample parameters for the SQL Reduction facility