EPV Technologies

EPV for Db2
Preparing Input for a Demo

IT Cost
Under Control

Supporting

EPV for Db2 V15
EPV for Db2 V15 (SAS based)

April 2021

Contents

About this manual

This manual is intended to help anyone who wants to provide the data needed to prepare an EPV for Db2 demo.

Changes

Technical changes or additions to the text are indicated by a vertical line to the left of the change.

Terminology

A “view” is an EPV report presented in an HTML page.

1 Introduction

The best way to evaluate the benefits provided by EPV for Db2 for customers is to have a demo based on their data in their own environment.
Providing the data needed to prepare a demo is a quick and easy task to perform.

In this manual, after a short description of EPV for Db2 input data, a simple four step process to do that is presented.
Sample JCLs are also provided.

2 Mandatory input data

Some SMF records data are mandatory in order to run EPV for Db2. If you don't provide them EPV will not produce any usable output.
They are:

- Record 30 subtype 2,3 (Address Space Interval activity);
- Record 70 (RMF CPU activity);
- Record 100 (Db2 Statistics);
- Record 101 (Db2 Accounting).

2.1 SMF 30 subtype 2, 3 records

SMF 30 subtype 2 and 3 records are not produced by default.

To activate SMF interval accounting using the global recording interval the following parameters have to be set in the SMFPRMxx member of the SYS1.PARMLIB library:

- INTVAL(mm) where mm is the interval duration; suggested values are 10 or 15 minutes;
- SYNCVAL(nn) where nn is the minute in the hour that starts the interval; suggested value is 00;

In addition the following parameter have to be set under SYS and SUBSYS sections:

- INTERVAL(SMF,SYNC).

Writing of these records has to be allowed in SMFPRMxx (under the TYPE sub parameter).

It's very important you synchronise SMF and RMF data; to do that you must set the following parameter in the ERBRMFxx member, used by RMF Monitor I, of your SYS1.PARMLIB library:

- SYNC(SMF).

2.2 SMF 70 records

SMF 70 records are produced by default.
However the following parameters are normally explicitly specified in RMF monitor I (ERBRMFxx member of the SYS1.PARMLIB library):

- CPU, to produce CPU activity information.

Writing of these records also has to be allowed in SMFPRMxx (under the TYPE sub parameter).

2.3 SMF 100 records

To produce SMF 100 the Db2 Statistic Trace, Class 1 has to be activated.

Writing of these records also has to be allowed in SMFPRMxx (under the TYPE sub parameter).

2.4 SMF 101 records

To produce SMF 101 the Db2 Accounting Trace, Class 1 has to be activated.

Writing of these records also has to be allowed in SMFPRMxx (under the TYPE sub parameter).

3 Other suggested input data

Using only the mandatory data will result in a subset of the EPV for Db2 views and analysis. So you are strongly advised to also provide additional info in SMF 100 and 101 records by activating also the following trace classes:

- Record 100, Db2 Statistic Trace, Class 3,5.
- Record 101, Db2 Accounting Trace, Class 2,3.

To get package CPU usage, accounting trace class 7 is needed.
To get information about package waits, accounting trace class 8 is needed.
Additional package information (such as the number of getpages) are available if trace class 10 is also activated.

4 Optional input data

EPV for Db2 can collect deadlock, timeout, FTB usage and BP dataset I/O statistics detail data by reading the 105, 172, 196, 199[1] and 389[2] IFCIDs from SMF 102 records.

EPV for Db2 can also collect information about:

- Db2 levels,
- table, index and feedback statistics,
- DSNZPARM parameter settings by running the DSNWZP or ADMIN_INFO_SYSPARM stored procedure provided by IBM.

The above information should be gathered daily from each Db2 subsystem.

[1] Statistics trace class 8 has to be activated.

[2] Statistics trace class 8 has to be activated.

5 Preparing data for a demo

To have a good demo, a few hours worth of data are enough. If you have more systems sharing resources the result will be better. If you had a bad day, with lot of problems, the EPV demo will probably help you understand what happened.
The following steps have to be performed in order to prepare input data for an EPV demo.

5.1 Collecting the data – Standard JCLs

When transferring variable data (VB or VBS) from the mainframe to other platforms it is obviously important to do that without corrupting the logical structure of the records.
There are different possibilities to reach this goal. In this chapter the standard JCLs to be used are provided. An alternative JCL to collect SMF data is provided in Chapter 6.

5.1.1 SMF

The following JCL will collect all the necessary SMF records.
It will also convert the SMF file to undefined format to avoid data corruption during the file transfer.

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE *smfinput* TO YOUR SMF INPUT FILE NAME
- CHANGE *smfpref* TO OUTPUT FILE PREFIX
- CHANGE *yyyyxxx* to the starting and ending Julian date you want to select
- CHANGE *hhmm* to the starting and ending hours you want to select
- CHANGE FTP parameters (*your.ftp.address, user and password*) to appropriate values

```
//SELSMF   EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//INDD1    DD DSN=smfinput,DISP=SHR
//OUTDD1   DD DSN=smfpref.VBS,DISP=(,CATLG),
//            UNIT=SYSDA, SPACE=(CYL,(100,100),RLSE),
//            DCB=(LRECL=32760,BLKSIZE=27998,RECFM=VBS)
//SYSIN    DD *
 INDD(INDD1,OPTIONS(DUMP))
 OUTDD(OUTDD1,TYPE(30(2,3),70,100,101,102))
 DATE(yyyyxxx,yyyyxxx)
 START(hhmm)
 END(hhmm)
/*
//* DO NOT CHANGE RECFM=U ON BOTH DD
//UNDSMF   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=smfpref.VBS,DISP=SHR,
//            DCB=(RECFM=U)
//SYSUT2   DD DSN=smfpref.UND,DISP=(,CATLG),
//            DCB=(RECFM=U),
//          UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE)
//SYSIN     DD DUMMY
/*
//FTPSMF    EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT    DD SYSOUT=*
//INPUT DD *
your.ftp.address
user password
quote PASV
bin
put ‘smfpref.UND’ /smfpref.smf
close
quit
/*
```

5.1.2 DSNZPARM settings

The following JCL will produce files containing DSNZPARM parameter settings by calling the IBM provided DSNWZP stored procedure.

WARNING: in order to use the ADMIN_INFO_SYSPARM stored procedure, please use the JCL in Attachment B

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE *prefix* TO YOUR Db2 LIB PREFIX
- CHANGE *db2id* TO YOUR Db2 ID
- CHANGE *seqpref* TO OUTPUT FILE PREFIX
- CHANGE FTP parameters (*your.ftp.address, user and password*) to appropriate values

```
//STEP01    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN     DD *
 DELETE seqpref.db2id.DSNZPARM
/*
//*
//STEP02    EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN     DD DUMMY
//SYSUT1    DD DATA,DLM=EE
/* REXX */
PARSE ARG SSID
IF SSID = '' THEN DO
   SAY '*** ERROR - SSID NOT SET'
   EXIT (12)
END
"SUBCOM DSNREXX"
IF RC = 1
THEN S_RC = RXSUBCOM('ADD','DSNREXX','DSNREXX')
ADDRESS DSNREXX "CONNECT " SSID

IF SQLCODE <> 0 THEN DO
```

```
   SAY '*** ERROR - UNEXPECTED SQLCODE CONNECTING TO' SSID
   SAY '*** ERROR - SQLCODE  = ' SQLCODE
   EXIT (12)
END
/* SET UP SQLDA FOR CALL TO DSNWZP */
INSQLDA.SQLD = 1
INSQLDA.1.SQLTYPE = 449        /* VARCHAR */
INSQLDA.1.SQLLEN  = 32000
INSQLDA.1.SQLIND  = 0
INSQLDA.1.SQLDATA = ' '

/* CALL PROCEDURE DSNWZP */
ADDRESS DSNREXX
"EXECSQL CALL :DSNWZP USING DESCRIPTOR :INSQLDA  "
IF SQLCODE = 0 THEN DO
    RESULT = INSQLDA.1.SQLDATA
    PARMCOUNT = 1
    I = POS(X2C(404025),RESULT)
    DO WHILE ( I <> 0)
      RESULT = SUBSTR(RESULT,4)
      NI = POS(X2C(404025),RESULT)
      IF NI = 0 THEN NI = LENGTH(RESULT) + 1
      ZPARM.PARMCOUNT = SUBSTR(RESULT,1,NI-1)
      ZPARM.0 = PARMCOUNT
      PARMCOUNT = PARMCOUNT + 1
      RESULT = SUBSTR(RESULT,NI)
      I = POS(X2C(404025),RESULT)
    END

    DO I = 1 TO ZPARM.0
      ROWVALU = ZPARM.I
      OROWVALU = ROWVALU
      L = POS('/',ROWVALU)
      W1 = SUBSTR(LEFT(ROWVALU,L-1),1,8)
      WR = SUBSTR(ROWVALU,L+1)
      L = POS('/',WR)
      W2 = SUBSTR(LEFT(WR,L-1),1,8)
      WR = SUBSTR(WR,L+1)
      L = POS('/',WR)
      W3 = SUBSTR(LEFT(WR,L-1),1,24)
      WR = SUBSTR(WR,L+1)
      L = POS('/',WR)
      W4 = SUBSTR(LEFT(WR,L-1),1,8)
      WR = SUBSTR(WR,L+1)
      L = POS('/',WR)
      W5 = SUBSTR(LEFT(WR,L-1),1,8)
      WR = SUBSTR(WR,L+1)
      L = POS('/',WR)
      W6 = SUBSTR(LEFT(WR,L-1),1,40)
      WR = SUBSTR(WR,L+1,40)
      IF I = ZPARM.0 THEN DO
         WW = '';
         DO K=1 TO 40
            XX=SUBSTR(WR,K,1)
            IF C2X(XX) = '00' THEN DO
```

```
               XX = ' '
            END
            SAY WW
            WW = WW!!XX
         END
         WR = WW
      END
      OUT.I = W2 W3 WR W6 W4 W5
    END
    ADDRESS
    "EXECIO * DISKW PARMOUT (STEM OUT. FINIS"
    IF RC <> 0 THEN DO
       SAY '*** ERROR - CAN NOT WRITE OUTPUT FILE'
       EXIT (12)
       END
    EXIT (0)
END
ELSE DO
   SAY '*** ERROR - UNEXPECTED SQLCODE CALLING DSNWZP'
   SAY '*** ERROR - SQL CODE = ' SQLCODE
   EXIT (12)
END
EE
//*
//SYSUT2    DD DSN=&&PDS(REXXDSNZ),UNIT=SYSDA,
//          DISP=(NEW,PASS,DELETE),
//          SPACE=(TRK,(1,1,1)),
//          DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
//*
//STEP03    EXEC PGM=IRXJCL,PARM='REXXDSNZ db2id'
//STEPLIB   DD DSN=prefix.SDSNLOAD,DISP=SHR
//SYSEXEC   DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
//SYSTSPRT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//PARMOUT   DD DISP=(,CATLG),DSN=seqpref.db2id.DSNZPARM,
//          UNIT=SYSDA,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=255,RECFM=FB,BLKSIZE=0)
//SYSTSIN   DD DUMMY
/*
//*
//STEP04    EXEC PGM=IEBGENER
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
//SYSUT1    DD DATA,DLM=EE
/* REXX */
TRACE 'O'
CVTADDR = C2D(STORAGE(10,4))   /* CVTADDR FROM PSA + X'10'   */
AMB = STORAGE(D2X(CVTADDR+X2D('154')),8)
DD = DATE()
R.0=1
R.1='EPV ==> SYSNAME' AMB DD
"EXECIO * DISKW OUT1 (STEM R. FINIS"
IF RC <> 0 THEN EXIT 8
EXIT 0
```

```
EE
//*
//SYSUT2    DD DSN=&&PDS(SYS),UNIT=SYSDA,
//            DISP=(NEW,PASS,DELETE),
//            SPACE=(TRK,(1,1,1)),
//            DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
//*
//STEP05      EXEC PGM=IRXJCL,PARM='SYS'
//SYSEXEC     DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
//SYSTSPRT    DD SYSOUT=*
//SYSPRINT    DD SYSOUT=*
//SYSUDUMP    DD SYSOUT=*
//OUT1        DD DISP=MOD,DSN=seqpref.db2id.DSNZPARM
//SYSTSIN     DD DUMMY
/*
//FTPSTAT EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
your.ftp.address
user password
quote PASV
ascii
put ‘seqpref.db2id.DSNZPARM’ db2id.DSNZPARM.TXT
close
quit
/*
```

5.1.3 Table space statistics

The following JCL will produce files containing table space statistics collected by querying the Db2 catalog.
WARNING: it can be used only for Db2 V9 or above; for Db2 V8 please use the JCL in Attachment A

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE prefix TO YOUR Db2 LIB PREFIX
- CHANGE db2id TO YOUR Db2 ID
- CHANGE seqpref TO OUTPUT FILE PREFIX
- CHANGE aulplan TO DSNTIAUL PLAN (default is DSNTIB81 in 8.1, DSNTIB91 in 9.1, DSNTIBA1 in 10.1 and DSNTIBB1 in 11.1)

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE seqpref.db2id.DBCOUNT
/*
//*
//STEP02   EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=db2id.DB2.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

```
//SYSPUNCH DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(db2id)
  RUN  PROGRAM(DSNTIAUL) PLAN(aulplan)  PARMS('SQL') -
       LIB('prefix.RUNLIB.LOAD')
/*
//SYSREC00 DD DISP=(,CATLG),DSN=seqpref.db2id.DBCOUNT,
//            UNIT=SYSDA,SPACE=(CYL,(1,1)),
//            DCB=(LRECL=133,RECFM=FB,BLKSIZE=0)
//SYSUDUMP DD DUMMY
//SYSIN    DD *
   SELECT * FROM (
   SELECT SUBSTR(CURRENT SERVER,1,4)          ,
          SUBSTR(';',1,1)                     ,
          SUBSTR(CURRENT MEMBER,1,4)          ,
          SUBSTR(';',1,1)                     ,
          SUBSTR(DBNAME,1,24)                 ,
          SUBSTR(';',1,1)                     ,
          SUBSTR(TSNAME,1,24)                 ,
          SUBSTR(';',1,1)                     ,
          DIGITS(PARTITION)                   ,
          SUBSTR(';',1,1)                     ,
          DIGITS(DECIMAL((CARDF)))            ,
          SUBSTR(';',1,1)                     ,
          DIGITS(DECIMAL((SPACEF)/(1024),10,2)),
          SUBSTR(';',1,1)                     ,
          (DATE(STATSTIME))
          FROM SYSIBM.SYSTABLEPART
          ) AS TBDUM;
/*
//*
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DATA,DLM=EE
/* REXX */
TRACE 'O'
CVTADDR = C2D(STORAGE(10,4))   /* CVTADDR FROM PSA + X'10'  */
AMB = STORAGE(D2X(CVTADDR+X2D('154')),8)
DD = DATE()
R.0=1
R.1='EPV ==> SYSNAME' AMB DD
"EXECIO * DISKW OUT2 (STEM R. FINIS"
IF RC <> 0 THEN EXIT 8
EXIT 0
EE
//*
//SYSUT2   DD  DSN=&&PDS(SYS),UNIT=SYSDA,
//            DISP=(NEW,PASS,DELETE),
//            SPACE=(TRK,(1,1,1)),
//            DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
//*
//STEP04   EXEC PGM=IRXJCL,PARM='SYS'
//SYSEXEC  DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
//SYSTSPRT DD SYSOUT=*
```

```
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//OUT2     DD DISP=MOD,DSN=seqpref.db2id.DBCOUNT
//SYSTSIN  DD DUMMY
/*
//FTPSTAT EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
your.ftp.address
user password
quote PASV
ascii
put ‘seqpref.db2id.DBCOUNT’  db2id.DBCOUNT.TXT
close
quit
/*
```

5.1.4 Index space statistics

The following JCL will produce files containing index space statistics collected by querying the Db2 catalog.
WARNING: it can be used only for Db2 V9 or above

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE prefix TO YOUR Db2 LIB PREFIX
- CHANGE db2id TO YOUR Db2 ID
- CHANGE seqpref TO OUTPUT FILE PREFIX
- CHANGE aulplan TO DSNTIAUL PLAN (default is DSNTIB81 in 8.1, DSNTIB91 in 9.1, DSNTIBA1 in 10.1 and DSNTIBB1 in 11.1)

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE seqpref.db2id.IXCOUNT
/*
//*
//STEP02   EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=db2id.DB2.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(db2id)
  RUN  PROGRAM(DSNTIAUL) PLAN(aulplan)  PARMS('SQL') -
       LIB('prefix.RUNLIB.LOAD')
/*
//SYSREC00 DD DISP=(,CATLG),DSN=seqpref.db2id.IXCOUNT,
//         UNIT=SYSDA,SPACE=(CYL,(1,1)),
//         DCB=(LRECL=133,RECFM=FB,BLKSIZE=0)
//SYSUDUMP DD DUMMY
//SYSIN    DD *
```

```
   SELECT * FROM (
   SELECT SUBSTR(CURRENT SERVER,1,4)         ,
          SUBSTR(';',1,1)                    ,
          SUBSTR(CURRENT MEMBER,1,4)         ,
          SUBSTR(';',1,1)                    ,
          SUBSTR(DBNAME,1,24)                ,
          SUBSTR(';',1,1)                    ,
          SUBSTR(INDEXSPACE,1,24)            ,
          SUBSTR(';',1,1)                    ,
          DIGITS(PARTITION)                  ,
          SUBSTR(';',1,1)                    ,
          DIGITS(DECIMAL((CARDF)))           ,
          SUBSTR(';',1,1)                    ,
          DIGITS(DECIMAL((SPACEF)/(1024),10,2)),
          SUBSTR(';',1,1)                    ,
          (DATE(STATSTIME))                  ,
          SUBSTR(';',1,1)                    ,
          (DATE(LASTUSED))
     FROM SYSIBM.SYSINDEXPART ,
    (SELECT DBNAME,TBNAME,INDEXSPACE,NAME AS XNAME,CREATOR AS XCREATOR
     FROM SYSIBM.SYSINDEXES ) AS TBX ,
    (SELECT DBNAME AS SDBNAME,NAME AS SNAME,CREATOR AS SCREATOR
     ,PARTITION AS SPARTITION,LASTUSED
     FROM SYSIBM.SYSINDEXSPACESTATS ) AS TBXS
     WHERE IXNAME    = XNAME
     AND   IXCREATOR = XCREATOR
     AND   IXNAME    = SNAME
     AND   IXCREATOR = SCREATOR
     AND   PARTITION = SPARTITION
     ) AS TBDUM;
/*
//*
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DATA,DLM=EE
/* REXX */
TRACE 'O'
CVTADDR = C2D(STORAGE(10,4))   /* CVTADDR FROM PSA + X'10'   */
AMB = STORAGE(D2X(CVTADDR+X2D('154')),8)
DD = DATE()
R.0=1
R.1='EPV ==> SYSNAME' AMB DD
"EXECIO * DISKW OUT3 (STEM R. FINIS"
IF RC <> 0 THEN EXIT 8
EXIT 0
EE
//*
//SYSUT2   DD  DSN=&&PDS(SYS),UNIT=SYSDA,
//         DISP=(NEW,PASS,DELETE),
//         SPACE=(TRK,(1,1,1)),
//         DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
//*
//STEP04   EXEC PGM=IRXJCL,PARM='SYS'
//SYSEXEC  DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
```

```
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//OUT3     DD DISP=MOD,DSN=seqpref.db2id.IXCOUNT
//SYSTSIN  DD DUMMY
/*
//FTPSTAT EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
your.ftp.address
user password
quote PASV
ascii
put ‘seqpref.db2id.IXCOUNT’  db2id.IXCOUNT.TXT
close
quit
/*
```

5.1.5 Statistics feedback

The following JCL will produce files containing missing or conflicting statistics for tables and indexes collected by querying the Db2 catalog.
WARNING: it can be used only for Db2 V11 or above

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE prefix TO YOUR Db2 LIB PREFIX
- CHANGE db2id TO YOUR Db2 ID
- CHANGE seqpref TO OUTPUT FILE PREFIX
- CHANGE aulplan TO DSNTIAUL PLAN (default is DSNTIBB1 in 11.1)

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE seqpref.db2id.DBSTATFB
/*
//*
//STEP02   EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=db2id.DB2.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(db2id)
  RUN  PROGRAM(DSNTIAUL) PLAN(aulplan)  PARMS('SQL') -
       LIB('prefix.RUNLIB.LOAD')
/*
//SYSREC00 DD DISP=(,CATLG),DSN=SEQPREF.DB2ID.DBSTATFB,
//          UNIT=SYSDA,SPACE=(CYL,(1,1)),VOL=SER=VOLSER,
//          DCB=(LRECL=255,RECFM=FB,BLKSIZE=0)
//SYSUDUMP DD DUMMY
```

```
//SYSIN    DD *
   SELECT * FROM (
   SELECT SUBSTR(CURRENT SERVER,1,4)   ,
   SUBSTR(';',1,1)                     ,
   SUBSTR(CURRENT MEMBER,1,4)          ,
   SUBSTR(';',1,1)                     ,
   SUBSTR(DBNAME,1,24)                 ,
   SUBSTR(';',1,1)                     ,
   SUBSTR(TSNAME,1,24)                 ,
   SUBSTR(';',1,1)                     ,
   SUBSTR(TBNAME,1,44)                 ,
   SUBSTR(';',1,1)                     ,
   SUBSTR(IXNAME,1,44)                 ,
   SUBSTR(';',1,1)                     ,
   SUBSTR(COLNAME,1,44)                ,
   SUBSTR(';',1,1)                     ,
   DIGITS(DECIMAL((NUMCOLUMNS)))       ,
   SUBSTR(';',1,1)                     ,
   TYPE                                ,
   SUBSTR(';',1,1)                     ,
   REASON                              ,
   SUBSTR(';',1,1)                     ,
   BLOCK_RUNSTATS                      ,
   SUBSTR(';',1,1)                     ,
   LASTDATE
   FROM SYSIBM.SYSSTATFEEDBACK
   ORDER BY DBNAME
   ) AS TBDUM;
/*
//*
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DATA,DLM=EE
/* REXX */
TRACE 'O'
CVTADDR = C2D(STORAGE(10,4))   /* CVTADDR FROM PSA + X'10'  */
AMB = STORAGE(D2X(CVTADDR+X2D('154')),8)
DD = DATE()
R.0=1
R.1='EPV ==> SYSNAME' AMB DD
"EXECIO * DISKW OUT4 (STEM R. FINIS"
IF RC <> 0 THEN EXIT 8
EXIT 0
EE
//*
//SYSUT2   DD  DSN=&&PDS(SYS),UNIT=SYSDA,
//             DISP=(NEW,PASS,DELETE),
//             SPACE=(TRK,(1,1,1)),
//             DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
//*
//STEP04   EXEC PGM=IRXJCL,PARM='SYS'
//SYSEXEC  DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

```
//SYSUDUMP DD SYSOUT=*
//OUT4     DD DISP=MOD,DSN=seqpref.db2id.DBSTATFB
//SYSTSIN  DD DUMMY
/*
//FTPSTAT EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
your.ftp.address
user password
quote PASV
ascii
put 'seqpref.db2id.DBSTATFB' db2id.DBSTATFB.TXT
close
quit
/*
```

5.1.6 Db2 levels

The following JCL will produce files containing information about Db2 code, catalog and function levels.
WARNING: it can be used only for Db2 V12 or above

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE prefix TO YOUR Db2 LIB PREFIX
- CHANGE db2id TO YOUR Db2 ID
- CHANGE seqpref TO OUTPUT FILE PREFIX
- CHANGE aulplan TO DSNTIAUL PLAN (default is DSNTIBB1 in 11.1)

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE seqpref.db2id.DB2LEVEL
/*
//*
//STEP02   EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=db2id.DB2.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(db2id)
  -DISPLAY GROUP
  END
/*
//SYSTSPRT DD DISP=(,CATLG),DSN=seqpref.db2id.DB2LEVEL,
//         UNIT=SYSDA,SPACE=(CYL,(1,1)),
//         DCB=(LRECL=133,RECFM=FB,BLKSIZE=0)
//SYSUDUMP DD DUMMY
//SYSIN    DD DUMMY
/*
//*
```

```
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DATA,DLM=EE
/* REXX */
TRACE 'O'
CVTADDR = C2D(STORAGE(10,4))   /* CVTADDR FROM PSA + X'10'  */
AMB = STORAGE(D2X(CVTADDR+X2D('154')),8)
DD = DATE()
R.0=1
R.1='EPV ==> SYSNAME' AMB DD
"EXECIO * DISKW OUT5 (STEM R. FINIS"
IF RC <> 0 THEN EXIT 8
EXIT 0
EE
//*
//SYSUT2   DD  DSN=&&PDS(SYS),UNIT=SYSDA,
//             DISP=(NEW,PASS,DELETE),
//             SPACE=(TRK,(1,1,1)),
//             DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
//*
//STEP04   EXEC PGM=IRXJCL,PARM='SYS'
//SYSEXEC  DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//OUT5     DD DISP=MOD,DSN=seqpref.db2id.DB2LEVEL
//SYSTSIN  DD DUMMY
/*
//FTPSTAT EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
your.ftp.address
user password
quote PASV
ascii
put ‘seqpref.db2id.DB2LEVEL’ db2id.DB2LEVEL.TXT
close
quit
/*
```

5.2 Compressing the data

When the data is on PC you should compress it (the compression factor is usually very high).
Please be aware that compression tools may have limitations on the size of the file they can compress.

5.3 Sending the data

You can send data to EPV Technologies in two main ways:

- Uploading the data to the EPV FTP server;
- Creating a CD/DVD and sending it to our local distributor or directly to EPV Technologies via a courier service.

It's always better before sending the data to send a small file with only one SMF record type (e.g. SMF 70) by FTP or e-Mail, so we can confirm everything is correct before spending time sending large amounts of data.

6 Alternative JCL to collect SMF data

If you use the standard IBM FTP you can improve processing performance by eliminating the step performing the conversion to undefined by using the JCL provided in this chapter.

In the FTP step, records are read as if in undefined format in order to avoid FTP eliminating the VB and VBS headers and so corrupting the records. As stated in the comments it is essential not to change the RECFM parameter. It’s also required that the transfer is done in binary mode.

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE *smfinput* TO YOUR SMF INPUT FILE NAME
- CHANGE *smfpref* TO OUTPUT FILE PREFIX
- CHANGE *yyyyxxx* to the starting and ending Julian date you want to select
- CHANGE *hhmm* to the starting and ending hours you want to select
- CHANGE FTP parameters (*your.ftp.address, user and password*) to appropriate values

```
//SELSMF   EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//INDD1    DD DSN=smfinput,DISP=SHR
//OUTDD1   DD DSN=smfpref.VBS,DISP=(,CATLG),
//         UNIT=SYSDA, SPACE=(CYL,(100,100),RLSE),
//         DCB=(LRECL=32760,BLKSIZE=27998,RECFM=VBS)
//SYSIN    DD *
 INDD(INDD1,OPTIONS(DUMP))
 OUTDD(OUTDD1,TYPE(30(2,3),70,100,101,102))
 DATE(yyyyxxx,yyyyxxx)
 START(hhmm)
 END(hhmm)
/*
//* DO NOT CHANGE RECFM=U ON //DDSMF
//FTPSMF   EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//DDSMF    DD DSN=smfpref.VBS,RECFM=U,BLKSIZE=32760,DISP=SHR
//INPUT DD *
your.ftp.address
user password
quote PASV
bin
put //DD:DDSMF /smfpref.smf
close
quit
/*
```

7 Customer support

For any technical problem with or question about EPV for Db2 please write an email to:

epv.support@epvtech.com

For any other issue about EPV for Db2 please write an email to:

epv.info@epvtech.com

Attachment A

The following JCL will produce files containing table space statistics collected querying the Db2 catalog only for Db2 V8.

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE prefix TO YOUR Db2 LIB PREFIX
- CHANGE db2id TO YOUR Db2 ID
- CHANGE seqpref TO OUTPUT FILE PREFIX
- CHANGE aulplan TO DSNTIAUL PLAN (default is DSNTIB81 in 8.1, DSNTIB91 in 9.1, DSNTIBA1 in 10.1 and DSNTIBB1 in 11.1)

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE seqpref.db2id.DBCOUNT
/*
//*
//STEP02   EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD DISP=SHR,DSN=db2id.DB2.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(db2id)
  RUN  PROGRAM(DSNTIAUL) PLAN(aulplan)  PARMS('SQL') -
       LIB('prefix.RUNLIB.LOAD')
/*
//SYSREC00 DD DISP=(,CATLG),DSN=seqpref.db2id.DBCOUNT,
//         UNIT=SYSDA,SPACE=(CYL,(1,1)),
//         DCB=(LRECL=133,RECFM=FB,BLKSIZE=0)
//SYSUDUMP DD DUMMY
//SYSIN    DD *
   SELECT * FROM (
   SELECT SUBSTR(CURRENT SERVER,1,4)           ,
          SUBSTR(';',1,1)                      ,
          SUBSTR(CURRENT MEMBER,1,4)           ,
          SUBSTR(';',1,1)                      ,
          SUBSTR(DBNAME,1,24)                  ,
          SUBSTR(';',1,1)                      ,
          SUBSTR(TSNAME,1,24)                  ,
          SUBSTR(';',1,1)                      ,
          DIGITS(PARTITION)                    ,
          SUBSTR(';',1,1)                      ,
          DIGITS(DECIMAL((CARDF))              ,
          SUBSTR(';',1,1)                      ,
          DIGITS(DECIMAL((SPACEF)/(1024),10,2)),
          SUBSTR(';',1,1)                      ,
          MIN(DATE(STATSTIME))
          FROM SYSIBM.SYSTABLEPART
          GROUP BY DBNAME, TSNAME, PARTITION) AS TBDUM;
/*
//*
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DATA,DLM=EE
/* REXX */
TRACE 'O'
CVTADDR = C2D(STORAGE(10,4))   /* CVTADDR FROM PSA + X'10'  */
AMB = STORAGE(D2X(CVTADDR+X2D('154')),8)
DD = DATE()
R.0=1
R.1='EPV ==> SYSNAME' AMB DD
"EXECIO * DISKW OUT2 (STEM R. FINIS"
IF RC <> 0 THEN EXIT 8
EXIT 0
EE
//*
//SYSUT2   DD  DSN=&&PDS(SYS),UNIT=SYSDA,
//         DISP=(NEW,PASS,DELETE),
//         SPACE=(TRK,(1,1,1)),
//         DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
//*
//STEP04   EXEC PGM=IRXJCL,PARM='SYS'
//SYSEXEC  DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//OUT2     DD DISP=MOD,DSN=seqpref.db2id.DBCOUNT
//SYSTSIN  DD DUMMY
/*
//FTPSTAT EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
your.ftp.address
user password
quote PASV
ascii
put ‘seqpref.db2id.DBCOUNT’  db2id.DBCOUNT.TXT
close
quit
/*
```

Attachment B

The following JCL will produce files containing DSNZPARM parameter settings by calling the IBM provided ADMIN_INFO_SYSPARM stored procedure.

Cut and paste it in your JCL library, and do the following customizations:

- CHANGE *prefix* TO YOUR Db2 LIB PREFIX
- CHANGE *db2id* TO YOUR Db2 ID
- CHANGE *seqpref* TO OUTPUT FILE PREFIX
- CHANGE FTP parameters (*your.ftp.address, user and password*) to appropriate values

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DELETE seqpref.db2id.DSNZPARM
/*
//*
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DATA,DLM=EE
/* REXX */
PARSE ARG SSID
IF SSID = '' THEN DO
   SAY '*** ERROR - SSID NOT SET'
   EXIT (12)
END
"SUBCOM DSNREXX"
IF RC = 1
THEN S_RC = RXSUBCOM('ADD','DSNREXX','DSNREXX')
ADDRESS DSNREXX "CONNECT " SSID

IF SQLCODE <> 0 THEN DO
   SAY '*** ERROR - NO CONNECTION TO DB2'
   CALL SQLCAERR
   EXIT (12)
END
SSID_IND = -1
RETCD_IND=0
RETCD=0
MSSG_IND=0
MSSG=LEFT(' ',1331,' ')

INSQLDA.SQLD = 1
INSQLDA.1.SQLTYPE = 449        /* VARCHAR */
INSQLDA.1.SQLLEN  = 32000
INSQLDA.1.SQLIND  = 0
INSQLDA.1.SQLDATA = ' '

PROC = 'SYSPROC.ADMIN_INFO_SYSPARM'
ADDRESS DSNREXX 'EXECSQL CALL ' PROC,
                              '(:SSID  :SSID_IND',
                              ',:RETCD :RETCD_IND',
                              ',:MSSG  :MSSG_IND)'
IF SQLCODE < 0 THEN DO
   CALL SQLCAERR
   EXIT (20)
END
ELSE
SAY 'PROC CALL IS OK'
DO
   ADDRESS DSNREXX 'EXECSQL DESCRIBE PROCEDURE :PROC INTO :SQLDA'
   IF SQLCODE <> 0 THEN DO
      CALL SQLCAERR
      EXIT (20)
   END
   SAY 'PROC DESCRIBE IS OK'

   /****************************************************************/
   /* SET UP A CURSOR                                              */
   /****************************************************************/
   ADDRESS DSNREXX 'EXECSQL ASSOCIATE LOCATOR (:MSSG)',
                   'WITH PROCEDURE :PROC'
   IF SQLCODE <> 0 THEN DO
      CALL SQLCAERR
      EXIT (20)
   END
   SAY 'PROC LOCATE IS OK'

   ADDRESS DSNREXX "EXECSQL ALLOCATE C101 CURSOR FOR RESULT SET :MSSG"
   IF SQLCODE <> 0 THEN DO
      CALL SQLCAERR
      EXIT (20)
   END
   SAY 'ALLOCATE CURSOR IS OK'

   CURSOR = 'C101'
   ADDRESS DSNREXX "EXECSQL DESCRIBE CURSOR :CURSOR INTO :SQLDA"
   IF SQLCODE <> 0 THEN DO
      CALL SQLCAERR
      EXIT (20)
   END
   SAY 'DESCRIBE CURSOR IS OK'

  /****************************************************************/
  /* RETRIEVE THE ROWS FROM THE RESULT SET                        */
  /****************************************************************/
  I=0
  DO UNTIL(SQLCODE <> 0)
      ADDRESS DSNREXX 'EXECSQL FETCH C101 ',
                'INTO :ROWNUM, :MACRO, :PARAMETER,',
                ':INSTALL_PANEL       :IND_INSTALL_PANEL,',
                ':INSTALL_FIELD       :IND_INSTALL_FIELD,',
                ':INSTALL_LOCATION    :IND_INSTALL_LOCATION,',
                ':VALUE,',
                 ':ADDITIONAL_INFO     :IND_ADDITIONAL_INFO'
      IF SQLCODE = 0 THEN
      DO
        I = I + 1
        PRM.I = SUBSTR(MACRO,1,8)
        PRM.I = PRM.I SUBSTR(PARAMETER,1,32)
        PRM.I = PRM.I SUBSTR(STRIP(VALUE),1,40)
        PRM.I = PRM.I SUBSTR(INSTALL_FIELD,1,40)
        PRM.I = PRM.I SUBSTR(INSTALL_PANEL,1,8)
        PRM.I = PRM.I SUBSTR(INSTALL_LOCATION,1,5)
        PRM.I = PRM.I SUBSTR(STRIP(ADDITIONAL_INFO),1,20)
      END
  END
  PRM.0=I
  IF SQLCODE = 100 & I = 0 THEN DO
     SAY 'ERROR NO PARMS RETURNED'
     EXIT (12)
  END
  "EXECIO *  DISKW PARMOUT (STEM PRM. FINIS"
  IF RC <> 0 THEN DO
     SAY 'ERROR IN WRITING TO OUTPUT FILE'
     EXIT (12)
  END
  ELSE SAY 'PARMS WRITTEN = ' I
END

ADDRESS DSNREXX "EXECSQL CLOSE C101"
IF SQLCODE <> 0 THEN CALL SQLCAERR
ADDRESS DSNREXX "EXECSQL COMMIT"
IF SQLCODE <> 0 THEN CALL SQLCAERR
ADDRESS DSNREXX 'DISCONNECT'
IF SQLCODE <> 0 THEN CALL SQLCAERR

S_RC = RXSUBCOM('DELETE','DSNREXX','DSNREXX')

EXIT

SQLCAERR:
SAY 'SQLCODE   = ' SQLCODE
SAY 'SQLSTATE  = ' SQLSTATE
SAY 'SQLCAERRD.1 = ' SQLCAERRD.1
SAY 'SQLCAERRD.2 = ' SQLCAERRD.2
SAY 'SQLCAERRD.3 = ' SQLCAERRD.3
SAY 'SQLCAERRD.4 = ' SQLCAERRD.4
SAY 'SQLCAERRD.5 = ' SQLCAERRD.5
SAY 'SQLCAERRD.6 = ' SQLCAERRD.6
SAY 'SQLCAERRMC  = ' SQLCAERRMC
RETURN
EE
//*

//SYSUT2    DD DSN=&&PDS(REXXDSNZ),UNIT=SYSDA,
//          DISP=(NEW,PASS,DELETE),
//          SPACE=(TRK,(1,1,1)),
//          DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
```

```
//*
//STEP03     EXEC PGM=IRXJCL,PARM='REXXDSNZ db2id'
//STEPLIB    DD DSN=prefix.SDSNLOAD,DISP=SHR
//SYSEXEC    DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
//SYSTSPRT   DD SYSOUT=*
//SYSPRINT   DD SYSOUT=*
//SYSUDUMP   DD SYSOUT=*
//PARMOUT    DD DISP=(,CATLG),DSN=seqpref.db2id.DSNZPARM,
//           UNIT=SYSDA,SPACE=(CYL,(1,1)),
//           DCB=(LRECL=255,RECFM=FB,BLKSIZE=0)
//SYSTSIN    DD DUMMY
/*
//*
//STEP04     EXEC PGM=IEBGENER
//SYSPRINT   DD SYSOUT=*
//SYSIN      DD DUMMY
//SYSUT1     DD DATA,DLM=EE
/* REXX */
TRACE 'O'
CVTADDR = C2D(STORAGE(10,4))   /* CVTADDR FROM PSA + X'10'  */
AMB = STORAGE(D2X(CVTADDR+X2D('154')),8)
DD = DATE()
R.0=1
R.1='EPV ==> SYSNAME' AMB DD
"EXECIO * DISKW OUT1 (STEM R. FINIS"
IF RC <> 0 THEN EXIT 8
EXIT 0
EE
//*
//SYSUT2     DD DSN=&&PDS(SYS),UNIT=SYSDA,
//           DISP=(NEW,PASS,DELETE),
//           SPACE=(TRK,(1,1,1)),
//           DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB,DSORG=PO)
//*
//STEP05     EXEC PGM=IRXJCL,PARM='SYS'
//SYSEXEC    DD DSN=&&PDS,DISP=(OLD,DELETE,DELETE)
//SYSTSPRT   DD SYSOUT=*
//SYSPRINT   DD SYSOUT=*
//SYSUDUMP   DD SYSOUT=*
//OUT1       DD DISP=MOD,DSN=seqpref.db2id.DSNZPARM
//SYSTSIN    DD DUMMY
/*
//FTPSTAT EXEC PGM=FTP,PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
your.ftp.address
user password
quote PASV
ascii
put ‘seqpref.db2id.DSNZPARM’ db2id.DSNZPARM.TXT
close
quit
/*
```

Related documentation

The following manuals complement the information provided in this manual:

- *EPV for Db2 V15 Installation and Customization Guide*
- *EPV for Db2 V15 (SAS based) Installation and Customization Guide*
- *EPV for Db2 V15 List of Views*
- *EPV for Db2 V15 Release Notes*
- *EPV for Db2 V15 DataBase Layout*
- *EPV V15 User Interface*
- *EPV V15 Messages and Codes*