



Loading Data in EPV for z/OS

*IT Cost
Under Control*



Supporting
EPV for z/OS Plus V11

May 2013



All the trademarks mentioned belong to their respective companies.



Contents

1	Overview	- 5 -
2	Preparing input data	- 6 -
2.1	Dumping SMF interval records.....	- 6 -
2.2	Dumping SMF event records	- 7 -
2.3	Dumping IMS FA records.....	- 8 -
	Related documentation.....	- 9 -



About this manual

The aim of this manual is to describe the steps to be taken whenever you need to load or reload old data into your EPV databases (e.g. to fill in gaps).

Changes

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



1 Overview

Most of the information stored inside the EPV databases is aggregated at the hourly level, so this may lead to data being overwritten whenever you load or reload old data.

Suppose you have no data from 8:00 to 16:40 on a specific date and you want to recover this data. If you take input data from 7:55 to 17:01 you will cover the gap between 8:00 and 17:00 but you will corrupt two hours at the borders. In fact hour 7 will contain measurements based only on the last 5 minutes (from 7:55 to 8:00). Hour 17 will be even worse: it will contain measurements based on one minute only (from 17:00 to 17:01).

In reality this issue could also happen in the normal daily process; this is the reason why EPV uses a specific database, by default called MDETA, during the loading phase.

MDETA is an intermediate database where detail data is stored directly after the parsing phase without any aggregation. Theoretically, if all the data types are present in the MDETA and the date you want to load is included in the retention period of the database, the MDETA database will automatically manage the merging of the data and will ensure there is no data corruption at the borders as described above.

Unfortunately, due to the large amount of data managed each day:

- the MDETA database may not contain all the data types (e.g. SMF record 101 for DDF and SMF record 110 for CICS by default are not stored there);
- the data retention period in the database may be limited to a few days (by default 3 days).

During normal daily processing all the data in MDETA is protected. Any recently collected data may not be accurate in any given hour due to the lack of precision in the collection periods. However after the merging of the retained data and subsequent collections then the data will be complete, and will be ready for processing.

When loading or reloading old data it is unlikely the MDETA will be able to protect your data so you need to follow some simple steps, described in the following chapters, to correctly build the input files in order to avoid the data corruption at the borders issue.



2 Preparing input data

For all the required records you need to make a dump that completely covers the hours you want to load. So if you have a gap inside your EPV databases between the 09:00 AM until the 10:30 AM on a specific date, you have to prepare an input file completely covering the hours 9 and 10.

The information stored in the SMF logs could be grouped in two different types: interval records and event records.

Due to the way EPV uses them, you need to dump your data using two separate steps:

- one for the interval records: 30 (subtype 2,3,6), 70, 71, 72, 73, 74, 75, 78, VSM (user record 252 by default), VTS (user record, 194 by default);
- one for the event records: 0, 30 (subtype 5), 101, 110, 116, 120.

In addition, if you need to load IMS FA records, you need to perform another input preparation step.

Examples of preparing input files to load the 1st January 2011 data from 09,00 to 11,00 are provided in the following sections.

2.1 Dumping SMF interval records

In this case, and only for the interval records, you need to make a dump filtering the data using the rule: "between 09:01 AM until 11:01 AM".

A filter of this type will take all the intervals related to the range of hours from the 9 to the 11 filling the holes and replacing the old data with a new set of data that anyway is complete.

You can use the following JCL as an example¹:

```
//JJJJJJJ JOB CLASS=J,CLASS=J,MSGCLASS=J,NOTIFY=JJJJJJ
//SELSMF EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//INDD1 DD DSN=SMF.DATASETS,DISP=SHR
//OUTDD1 DD DSN=SMF.DATASETS.SELECTION,DISP=(,CATLG),
// UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE),
// DCB=(LRECL=32756,BLKSIZE=27998,RECFM=VBS)
//SYSIN DD *
INDD(INDD1,OPTIONS(DUMP))
OUTDD(OUTDD1,TYPE(30(2,3,6),70:78,194,252))
DATE(2011001,2011001)
START(0901)
END(1101)
/*
```

¹ SMF 194 is the default user record for the IBM VTS: SMF 252 is the default user record for the ORACLE VSM.



2.2 Dumping SMF event records

In this case you need to make a dump filtering the data using the rule: "between 09:00 AM until 10:59 AM".

A filter of this type will take all the intervals related to the range of hours from 9 to 11 filling the gaps and replacing the old data with a new set of data that is complete.

You can use the following JCL as an example:

```
//JJJJJJJ JOB CLASS=J,CLASS=J,MSGCLASS=J,NOTIFY=JJJJJJ
//SELSMF EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//INDD1 DD DSN=SMF.DATASETS,DISP=SHR
//OUTDD1 DD DSN=SMF.DATASETS.SELECTION,DISP=(,CATLG),
// UNIT=SYSDA, SPACE=(CYL,(100,100),RLSE),
// DCB=(LRECL=32756,BLKSIZE=27998,RECFM=VBS)
//SYSIN DD *
INDD(INDD1,OPTIONS(DUMP))
OUTDD(OUTDD1,TYPE(0,30(5),101,110,116,120))
DATE(2011001,2011001)
START(0900)
END(1059)
/*
```



2.3 Dumping IMS FA records

IMF FA records are written by event so this step is conceptually identical to the previous one.

You can use the following JCL as an example²:

```
//JJJJJJJ JOB CLASS=J,CLASS=J,MSGCLASS=J,NOTIFY=JJJJJJ
//STEP1 EXEC PGM=SORT
//SORTIN DD DSN=IMS.LOGS,DISP=SHR
//SORTOUT DD DSN=IMS.LOGS.SELECTION,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(30,5))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(30,5))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(30,5))
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
SORT FIELDS=COPY
INCLUDE COND=(21,4,CH,EQ,C'IMS1',
              AND,105,4,PD,EQ,+2011001,
              AND,109,4,BI,GE,X'09000000',
              AND,109,4,BI,LT,X'11000000')
/*
```

² One JCL for each IMS has to be run; in this example the IMS1 subsystem is specified in the INCLUDE command.



Related documentation

The following manuals complement the information provided in this manual:

- *EPV for z/OS V11 Plus Installation and Customization Guide*
- *EPV for z/OS V11 Release Notes*
- *EPV for z/OS V11 List of Views*
- *EPV for z/OS V11 Getting Started*
- *EPV for z/OS V11 Preparing Input for a Demo*
- *EPV V11 User Interface*
- *EPV Plus Products Operations Guide*