



# EPV for z/OS Preparing Input for a Demo



Supporting  
**EPV for z/OS version V15**

**October 2019**



**All the trademarks mentioned belong to their respective companies.**

---

**EPV Technologies contact details:**

EPV Technologies  
Viale Angelico, 54  
00195 Roma  
Tel. 06 86210880  
Fax. 06 86387461  
E-mail: [epvtech@epvtech.com](mailto:epvtech@epvtech.com)  
WEB: <http://www.epvtech.com>

---



---

## Contents

1	Introduction.....	- 5 -
2	Mandatory input data .....	- 6 -
2.1	SMF 30 subtype 2, 3 records.....	- 6 -
2.2	SMF 70 and 72 records .....	- 6 -
3	Other needed input data .....	- 7 -
4	Optional input data.....	- 8 -
5	Preparing data for a demo .....	- 10 -
5.1	Collecting the data - Standard JCLs .....	- 10 -
5.1.1	SMF.....	- 10 -
5.1.2	DCOLLECT.....	- 11 -
5.1.3	IMS.....	- 12 -
5.1.4	ORACLE-STORAGETEK VTCS.....	- 13 -
5.1.5	IBM VTS.....	- 14 -
5.2	Compressing the data .....	- 16 -
5.3	Sending the data .....	- 16 -
6	Alternative JCLs .....	- 17 -
6.1	Improving performance by using the standard IBM FTP .....	- 17 -
6.1.1	SMF.....	- 17 -
6.1.2	DCOLLECT.....	- 18 -
6.1.3	IMS.....	- 18 -
6.2	Other supported IMS records .....	- 20 -
6.2.1	IMS – FA RECORDS SELECTION .....	- 20 -
6.2.2	IMS – 7,8 RECORDS SELECTION .....	- 20 -
6.3	IBM VTS – USING SMF RECORDS .....	- 22 -
7	Customer support.....	- 23 -
	Related documentation.....	- 24 -



## **About this manual**

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

## **Changes**

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



## 1 Introduction

The best way to evaluate the benefits provided by EPV for z/OS 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 z/OS 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 z/OS. 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 and 72 (RMF CPU and Workload activity).

### 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 and 72 records

SMF 70 and 72 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;
- WKLD, to produce Workload activity information.

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



### 3 Other needed input data

Using only the mandatory data will result in only a small subset of the EPV for z/OS views and analysis. So you are strongly advised to also provide the following SMF records:

- Record 0 (IPL); produced by default; **Warning:** this record is deprecated; by default EPV uses SMF 90 subtype 8 and 10; if you want to use SMF 0 you should customize the URIPLS user exit;
- Record 30 subtype 4,5,6 (Address Space activity); produced by default;
- Record 71 (Paging activity); produced by default; PAGING is normally explicitly specified in RMF Monitor I options;
- Record 73, (Channel activity); produced by default; CHAN is normally explicitly specified in RMF Monitor I options;
- Record 74 subtype 1, (Device activity); produced by default; DEVICE(DASD) is normally explicitly specified in RMF Monitor I options;
- Record 74 subtype 5, (Cache activity); produced by default; CACHE is normally explicitly specified in RMF Monitor I options;
- Record 75, (Page dataset activity); produced by default; PAGESP is normally explicitly specified in RMF Monitor I options;
- Record 78 subtype 2, (Virtual Storage activity); produced by default; VSTOR is normally explicitly specified in RMF Monitor I options;
- Record 78 subtype 3, (I/O Queuing activity); produced by default; IOQ(DASD) is normally explicitly specified in RMF Monitor I options;
- Record 90 subtype 8 and 10 (System status); produced by default; used for IPL data instead of SMF 0;
- Record 113 subtype 2, (hardware capacity, reporting, and statistics); produced only when activating Hardware Instrumentation Services (HIS) data collection.

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

EPV supports disk volume space statistics, produced using the IBM utility IDCAMS with the DCOLLECT input option. This data needs to be gathered daily for all the systems.<sup>1</sup>

The production step is very quick, and the added value is great because EPV for z/OS will then provide views showing both available and used space, for all SMS groups.

---

<sup>1</sup> If a system can access all disks then its enough to gather DCOLLECT data from this system.

---



## 4 Optional input data

If you want EPV for z/OS to produce subsystems activity views the following SMF records have to be provided:

- Record 101, (DB2 Accounting);
- Record 110, (CICS activity);
- Record 116, (Websphere MQ activity);
- Record 120, (Websphere activity).

The default is that these records are not produced. You have to enable their production by the appropriate subsystem settings.

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

IMS activity information is not provided by any SMF records. The information is generally gathered through 3 different sources:

- 7 and 8 record types from the IMS LOG<sup>2</sup>; to use this option you have to create a separate dataset for each IMS and set the first and second qualifiers of the dataset name to system and IMS id (see examples in Chapter 6); you should also communicate the IMS level to EPV support;
- FA record type produced by Mainview for IMS by BMC Software.

This information are gathered by EPV for IMS DC only.

EPV supports all these sources, but we suggest the FA or TAR record type, because it is more accurate and complete.

Other optional SMF record types are:

- record 42 subtype 6 (Data Set activity); produced by default; no parameter has to be specified;
- record 74 subtype 2 (XCF activity); produced by default by RMF Monitor III; no parameter has to be specified;
- record 74 subtype 3 (OMVS Kernel activity); produced by default by RMF Monitor III; OPD is normally explicitly specified in RMF Monitor III options;
- record 74 subtype 4 (Coupling Facility activity); produced by default by RMF Monitor III; CFDETAIL is normally explicitly specified in RMF Monitor III options;
- record 74 subtype 7 (Ficon Director activity); not produced by default; NOFCD is normally explicitly specified in RMF Monitor I options; FCD has to be set;
- record 74 subtype 8 (PPRC activity); not produced by default; NOESS is normally explicitly specified in RMF Monitor I options; ESS has to be set;

---

<sup>2</sup> Reports based on Input Queue time are not provided using this option.

---





- record 74 subtype 9 (PCIE activity); produced by default by RMF Monitor III; PCIE is normally explicitly specified in RMF Monitor III options;
- record 74 subtype 10 (SCM activity); produced by default by RMF Monitor III; SCM is normally explicitly specified in RMF Monitor III options;
- record 89 subtype 1 (Usage Data); produced by default; no parameter has to be specified.

IBM VTS (also called Hydra) provides an internal facility called Bulk Volume Information Retrieval (BVIR) which also allows to get Historical Statistics by using a two step JCL running the standard IEBGENER utility. In alternative a SMF user record can be produced.

EPV supports both input types.

ORACLE-STORAGETEK VTCS (also called VSM) produces a SMF user record with many subtypes. EPV supports 10, 11, 13, 14 and 20 subtype.

EPV also supports VSM tape volume statistics produced using the SWSADMIN utility with the QUMVCP NAME(ALL) option. These statistics must be gathered daily for all the systems.<sup>3</sup>

---

<sup>3</sup> If a system can access all ACS then it's enough to gather statistics from this system.

---



## 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. Alternative JCLs are discussed in Chapter 6.

#### 5.1.1 SMF

The following JCL will collect all the necessary SMF records. It will also select optional subsystems records for DB2, CICS, Websphere MQ and Websphere. The sss record type is the VTCS (VSM) user record. The iii record type is the VTS (Hydra) user record. If you are not interested in any of them simply remove the record type from the OUTDD line.

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(0,30(2,3,4,5,6),42,70:78,89,90,101,110,113,116,120, sss, iii))
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 DCOLLECT

The following JCL will collect the necessary DCOLLECT information.

It will also convert the DCOLLECT file to undefined format to avoid data corruption during the file transfer.

If a system can access all disks then it is sufficient to gather DCOLLECT data from this system.

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

- CHANGE *dcolpref* TO OUTPUT FILE PREFIX
- CHANGE *sysid* to the system id where you run the DCOLLECT job
- CHANGE FTP parameters (*your.ftp.address*, *user* and *password*) to appropriate values

```
//SELDCOL EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DCOUT DD DSN=dcolpref.sysid, DISP=(, CATLG, DELETE),
//      UNIT=SYSDA,
//      SPACE=(CYL, (10, 1), RLSE),
//      DCB=(RECFM=VB, LRECL=4092, BLKSIZE=27998)
//SYSIN DD *
DCOLLECT -
OUTFILE(DCOUT) -
VOLUMES( * ) -
NODATAINFO
/*
/* DO NOT CHANGE RECFM=U ON BOTH DD
//UNDCOL EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=dcolpref.sysid, DISP=SHR,
//      DCB=(RECFM=U)
//SYSUT2 DD DSN=dcolpref.sysid.UND, DISP=(, CATLG),
//      DCB=(RECFM=U),
//      UNIT=SYSDA, SPACE=(CYL, (100, 100), RLSE)
//SYSIN DD DUMMY
/*
```



```
//FTPCOL EXEC PGM=FTP, PARM='(EXIT'  
//SYSPRINT DD SYSOUT=*  
//OUTPUT DD SYSOUT=*  
//INPUT DD *  
your.ftp.address  
user password  
quote PASV  
bin  
put dcolpref.sysid.UND /dcolpref.sysid  
close  
quit  
/*
```

### 5.1.3 IMS

The following JCL will collect the 56FA records from the IMS log. You have to run this job only if you are interested in IMS transaction information.

It will also convert the IMS file to undefined format to avoid data corruption during the file transfer.

**WARNING: the name of the transferred file has to start with *sysid.imsid*.**

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

- CHANGE *reslib* TO YOUR IMS RESLIB FILE NAME
- CHANGE *sldsfile* TO SLDS FILE NAME
- CHANGE *imspref* TO OUTPUT FILE PREFIX
- CHANGE *imsid* to the IMS subsystem id
- CHANGE *sysid* to the system id where the IMS resides
- CHANGE FTP parameters (*your.ftp.address*, *user* and *password*) to appropriate values

```
//SELIMS EXEC PGM=DFSER10  
//STEPLIB DD DSN=reslib, DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=sldsfile,  
// DISP=(OLD, PASS), UNIT=TAPE  
//SYSUT4 DD DSN=imspref.sysid.imsid, DISP=(, CATLG, DELETE),  
// UNIT=SYSDA,  
// SPACE=(CYL, (100, 100), RLSE),  
// DCB=(RECFM=VB, LRECL=30970, BLKSIZE=30974)  
//SYSIN DD *  
CONTROL CNTL STOPAFT=EOF  
OPTION COPY OFFSET=5, FLDLEN=2, VALUE=56FA, COND=E  
END  
/*  
/* DO NOT CHANGE RECFM=U ON BOTH DD  
//UNDIMS EXEC PGM=IEBGGENER  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DSN=imspref.sysid.imsid, DISP=SHR,  
// DCB=(RECFM=U)  
//SYSUT2 DD DSN=imspref.sysid.imsid.UND, DISP=(, CATLG),  
// DCB=(RECFM=U),  
// UNIT=SYSDA, SPACE=(CYL, (100, 100), RLSE)  
//SYSIN DD DUMMY
```



```
/*
//FTPIMS EXEC PGM=FTP, PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//INPUT DD *
your.ftp.address
user password
quote PASV
bin
put 'imspref.sysid.imsid.UND' /sysid.imsid
close
quit
/*
```

### 5.1.4 ORACLE-STORAGETEK VTCS

The following JCL will produce the necessary VTCS pools information. You have to run this job only if you are interested in VTCS information.

If a system can access all ACS tape libraries then it's enough to gather statistics from this system. Cut and paste it in your JCL library, and do the following customizations:

- CHANGE *libpref* TO YOUR VTCS LIBRARY PREFIX
- CHANGE *vtcspref* TO OUTPUT FILE PREFIX
- CHANGE *sysid* to the system id where you run the VTCS job
- CHANGE FTP parameters (*your.ftp.address*, *user* and *password*) to appropriate values

```
//SELVSM EXEC PGM=SWSADMIN, PARM='MIXED'
//STEPLIB DD DSN=libpref.SLSLINK, DISP=SHR
//MVCOUT DD DUMMY
//SLSPRINT DD DSN=vtcspref.sysid, DISP=(, CATLG, DELETE),
// UNIT=SYSDA,
// SPACE=(TRK, (1, 1), RLSE),
// DCB=(LRECL=121, RECFM=FBA)
//SYSOUT DD SYSOUT=*
//SLSIN DD *
QU MVCP NAME(ALL)
/*
//FTPVSM EXEC PGM=FTP, PARM='(EXIT'
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//INPUT DD *
your.ftp.address
user password
quote PASV
ascii
put 'vtcspref.sysid' vtcspref.sysid
close
quit
/*
```



### 5.1.5 IBM VTS

The following JCL will produce VTS Historical Statistics. You have to run this job only if you are interested in VTS information.

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

- CHANGE *vtsunit* TO the esoteric name corresponding to the VTS
- CHANGE *vtspref* TO INPUT AND OUTPUT FILE PREFIX
- CHANGE *vtsid* to the VTS id
- CHANGE *jjj* to the Julian dates you want to collect
- CHANGE FTP parameters (*your.ftp.address*, *user* and *password*) to appropriate values

```
//VTSCOMM EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=vtspref.COMMAND,
// LABEL=(1,SL),DISP=(,CATLG),
// UNIT=vtsunit,DCB=(RECFM=F,BLKSIZE=80,LRECL=80,TRTCH=NOCOMP)
//SYSUT1 DD *
VTS BULK VOLUME DATA REQUEST
HISTORICAL STATISTICS FOR jjj-jjj
/*
/**
//SELVTS EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=vtspref.COMMAND,DISP=SHR,
// DCB=(RECFM=U,BLKSIZE=24000)
//SYSUT2 DD DSN=vtspref.vtsid.DATA,
// LABEL=(1,SL),DISP=(,CATLG),
// UNIT=vtsunit,DCB=(RECFM=U,BLKSIZE=24000)
//SYSIN DD DUMMY
/**
//FROMU2VB EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=vtspref.vtsid.DATA,DISP=SHR
//SYSUT2 DD DSN=vtspref.vtsid.VB,DISP=(,CATLG),
// DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
// UNIT=SYSDA,SPACE=(CYL,(10,5),RLSE)
//SYSIN DD DUMMY
/*
/**
//FROMVB2U EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=vtspref.vtsid.VB,DISP=SHR,
// DCB=(RECFM=U,BLKSIZE=32760)
//SYSUT2 DD DSN=vtspref.vtsid.VBUND,DISP=(,CATLG),
// DCB=(RECFM=U,BLKSIZE=32760),
// UNIT=SYSDA,SPACE=(CYL,(10,5),RLSE)
//SYSIN DD DUMMY
/*
//FTPVTS EXEC PGM=FTP,PARM='(EXIT)'
//SYSPRINT DD SYSOUT=*
```



```
//OUTPUT DD SYSOUT=*  
//INPUT DD *  
your.ftp.address  
user password  
quote PASV  
bin  
put `vtspref.vtsid.VBUND' /vtspref.vtsid  
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 JCLs

### 6.1 Improving performance by using the standard IBM FTP

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

#### 6.1.1 SMF

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(0,30(2,3,4,5,6),42,70:78,89,90,101,110,113,116,120,sss,iii))
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
/*
```



### 6.1.2 DCOLLECT

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 *dcolpref* TO OUTPUT FILE PREFIX
- CHANGE *sysid* to the system id where you run the DCOLLECT job
- CHANGE FTP parameters (*your.ftp.address, user and password*) to appropriate values

```
//SELDCOL EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DCOUT DD DSN=dcolpref.sysid, DISP=(,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(CYL,(10,1),RLSE),
// DCB=(RECFM=VB,LRECL=4092,BLKSIZE=27998)
//SYSIN DD *
DCOLLECT -
OUTFILE(DCOUT) -
VOLUMES(* ) -
NODATAINFO
/* DO NOT CHANGE RECFM=U ON //DDDCOL
//FTPDCOL EXEC PGM=FTP, PARM='(EXIT)'
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//DDDCOL DD DSN=dcolpref.sysid, DCB=RECFM=U, BLKSIZE=32760, DISP=SHR
//INPUT DD *
your.ftp.address
user password
quote PASV
bin
put //DD:DDDCOL /dcolpref.sysid
close
quit
/*
```

### 6.1.3 IMS

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 and BLKSIZE parameters. It's also required that the transfer is done in binary mode.

**WARNING: the name of the transferred file has to start with *sysid.imsid*.**

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

- CHANGE *reslib* TO YOUR IMS RESLIB FILE NAME
- CHANGE *sldsfile* TO SLDS FILE NAME
- CHANGE *imspref* TO OUTPUT FILE PREFIX
- CHANGE *imsid* to the IMS subsystem id



- CHANGE *sysid* to the system id where the IMS resides
- CHANGE FTP parameters (*your.ftp.address*, *user* and *password*) to appropriate values

```
//SELIMS EXEC PGM=DFSER10
//STEPLIB DD DSN=reslib,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=sldsfile,
// DISP=(OLD,PASS),UNIT=TAPE
//SYSUT4 DD DSN=imspref.sysid.imsid,DISP=(,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(CYL,(100,100),RLSE),
// DCB=(RECFM=VB,LRECL=30970,BLKSIZE=30974)
//SYSIN DD *
CONTROL CNTL STOPAFT=EOF
OPTION COPY OFFSET=5,FLDLLEN=2,VALUE=56FA,COND=E
END
/*
/* DO NOT CHANGE RECFM=U ON //DDIMEF
//FTPIMS EXEC PGM=FTP,PARM=(EXIT)
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//DDIMS DD DSN=imspref.sysid.imsid,DCB=RECFM=U,DISP=SHR
//INPUT DD *
your.ftp.address
user password
quote PASV
bin
put //DD:DDIMS /sysid.imsid
close
quit
/*
```



## 6.2 Other supported IMS records

As an alternative to 56FA you can use FA or 7 and 8 IMS log records.

Only the selection step is provided here.

### 6.2.1 IMS – FA RECORDS SELECTION

The following JCL step will select the FA records produced by the Mainview for IMS BMC product. You have to run this job only if you are interested in IMS transaction information.

**WARNING: the name of the transferred file has to start with *sysid.imsid*.**

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

- CHANGE *reslib* TO YOUR IMS RESLIB FILE NAME
- CHANGE *sldsfile* TO SLDS FILE NAME
- CHANGE *imspref* TO OUTPUT FILE PREFIX
- CHANGE *imsid* to the IMS subsystem id
- CHANGE *sysid* to the system id where the IMS resides

```
//SELIMS EXEC PGM=DFSERAI0
//STEPLIB DD DSN=reslib,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=sldsfile,
// DISP=(OLD,PASS),UNIT=TAPE
//SYSUT4 DD DSN=imspref.sysid.imsid,DISP=(,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(CYL,(100,100),RLSE),
// DCB=(RECFM=VB,LRECL=30970,BLKSIZE=30974)
//SYSIN DD *
CONTROL CNTL STOPAFT=EOF
OPTION COPY OFFSET=5,FLDLLEN=1,VALUE=FA,COND=E
END
/*
```

### 6.2.2 IMS – 7,8 RECORDS SELECTION

This is a deprecated option.

The following JCL will select the 7 and 8 records from the IMS log. You have to run this job only if you are interested in IMS transaction information.

**WARNING: the name of the transferred file has to start with *sysid.imsid*.**

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

- CHANGE *reslib* TO YOUR IMS RESLIB FILE NAME
- CHANGE *sldsfile* TO SLDS FILE NAME
- CHANGE *imspref* TO OUTPUT FILE PREFIX
- CHANGE *imsid* to the IMS subsystem id
- CHANGE *sysid* to the system id where the IMS resides



```
//SELIMS EXEC PGM=DFSER10
//STEPLIB DD DSN=reslib,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=sldsfile,
// DISP=(OLD,PASS),UNIT=TAPE
//SYSUT4 DD DSN=imspref.sysid.imsid,DISP=(,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(CYL,(100,100),RLSE),
// DCB=(RECFM=VB,LRECL=30970,BLKSIZE=30974)
//SYSIN DD *
CONTROL CNTL STOPAFT=EOF
OPTION COPY OFFSET=5,FLDLN=1,VALUE=07,COND=E
OPTION COPY OFFSET=5,FLDLN=1,VALUE=08,COND=E
END
/*
```



### 6.3 IBM VTS – USING SMF RECORDS

If you want to write VTS information in SMF, as an alternative to the standard JCL, you have to perform the following steps:

1) Go to the IBM web site:

<http://public.dhe.ibm.com/storage/tapetool/>

2) Download the following files:

- `ibmcntl.xmi` Execution JCL for current tape analysis tools.
- `ibmjcl.xmi` Parameters needed for job execution, but do NOT need to be modified by the user.
- `ibmload.xmi` Load library for executable load modules.
- `ibmpat.xmi` Data pattern library only needed if you will be running the QSAMDRVR utility.
- `ibmtools.txt` The txt file includes the instructions to load the libraries (.xmi files) on z/OS.

3) You need to customize and run the BVIRHSTx JCL



## 7 Customer support

For any technical problems or questions about EPV for z/OS please email:

[epv.support@epvtech.com](mailto:epv.support@epvtech.com)

For any other issue about EPV for z/OS please email:

[epv.info@epvtech.com](mailto:epv.info@epvtech.com)



## Related documentation

The following manuals complement the information provided in this manual:

- *EPV for z/OS V15 (SAS based) Installation and Customization Guide*
- *EPV for z/OS V15 Installation and Customization Guide*
- *EPV for z/OS V15 Database Layout*
- *EPV for z/OS V15 Release Notes*
- *EPV for z/OS V15 List of Views*
- *EPV for z/OS V15 Getting Started*
- *EPV V15 User Interface*
- *EPV V15 Messages and Codes*