



EPV Real Time Installation and Customization



Supporting
EPV zParser V15

October 2021



All the mentioned trademarks 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
WEB: <http://www.epvtech.com>



Contents

1	Introduction.....	- 5 -
2	Pre-requisites.....	- 6 -
3	Installation.....	- 7 -
3.1	z/OS component.....	- 7 -
3.2	Distributed component.....	- 8 -
4	Customization	- 9 -
4.1	Customization on z/OS	- 9 -
4.2	Customization on distributed platforms	- 10 -
6	Running.....	- 15 -
7	Customer support	- 17 -
	Related documentation.....	- 18 -



About this manual

This manual is intended to help anyone who wants to use EPV Real Time to retrieve SMF data directly from memory.

Terminology

The string \$EPVROOT used in this manual represents the path to your EPV installation folder.
The string \$ProfileName represents the name of your EPV User Profile.



1 Introduction

The goal of this manual is to explain how to install and customize EPV Real Time.

EPV Real Time is a new product designed to access SMF data directly in the z/OS memory and send it to an external server for data collection and processing.

EPV Real Time has three main programs, the first one is running on z/OS and the other two on a Windows, Linux or Unix server:

- the EPVSMFS started task, in z/OS, reads SMF records in-memory and sends them to the distributed environment;
- data is being received by the EPV Real Time Socket Server that distributes it in multiple folders;
- the EPV Real Time Agent manages all the data reading and storing activities.

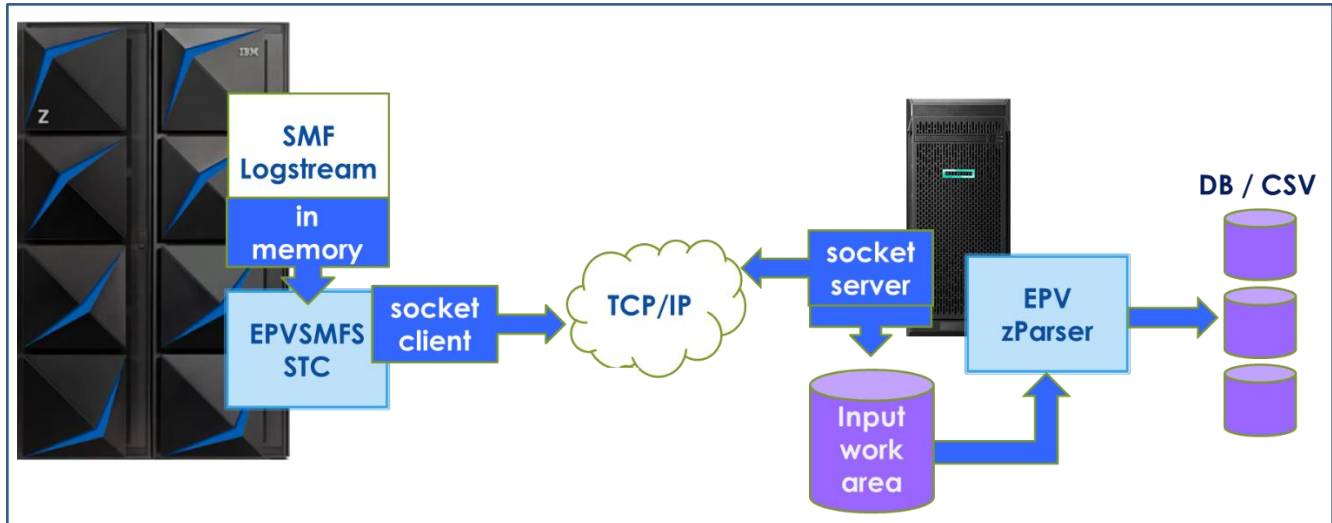


Figure 1

On z/OS, SMF data is produced and immediately available in memory. Our EPVSMFS started task takes data and sends it via TCP/IP connection to the distributed environment where the other EPV Real Time components run..

Data received by the EPV Real Time Socket Server is read and stored into an input work area which is read by an EPV Real Time Agent thread. Data is then inserted in one of the EPV Real Time databases.

On the first day you start using EPV Real Time, data is inserted in the first database. During the night, an automatic process makes a switch from the first database to the second one: from that moment on, data is inserted in the second database. This behaviour goes on and on every day, in a round-robin fashion.



2 Pre-requisites

The following pre-requisites must be satisfied:

- z/OS 2.2 or above,
- SMF recording to logstreams.



3 Installation

To install EPV Real Time you need to perform the simple steps described in the following chapters.

3.1 z/OS component

EPV Technical Support will provide the EPVSMFH and EPVSMFP programs in TSO TRANSMIT format.

You need to:

1. transfer the EPVSMFH and EPVSMFP files to z/OS via FTP using the 'bin' option;¹
2. unload the files to a PDS using the TSO RECEIVE command.

Once each file is transferred, to perform step 2 you need to issue the following TSO command:

```
RECEIVE INDSN('dsname')
```

where dsname is the data set name used for the file transfer.

When prompted for additional parameters, enter the following:

```
DSN('outdsname') [VOL(volser)]
```

where outdsname (default is userHLQ.ASMLIB) is the name of the partition data set which has to contain the files being received, and volser is the volume serial number of the device on which this data set will reside.²

Then, the programs should be copied to a LOAD library with the following characteristics:

- Record format: U
- Record length: 0
- Block size: 32760
- Data set name type : LIBRARY

Finally, prepare a member in your PROCLIB where you'll put the JCL which is explained in Chapter 5.

¹ You can also use "Send File to Host" on your 3270 Emulator to transfer the file to the destination system.

² The VOL operand is optional and can be omitted if the PDS exists.



3.2 Distributed component

Starting from EPV Setup v15.20 and EPV zParser v15.59, we introduced the support to EPV Real Time.

To use EPV Real Time you need to update the EPV zParser and EPV Setup at least to the versions mentioned before. To install the EPV zParser and EPV Setup please follow the related user manuals.³

If you opt for a Linux/Unix server, you need to check that the Perl interpreter is installed.

To complete the Socket Server installation, the following additional Perl Modules should be installed:

- Socket;
- Time::HiRes;
- IO::Handle.

³ EPV zParser V15 Installation and Customization.



4 Customization

To customize EPV Real Time you need to perform the simple steps described in the following chapters.

4.1 Customization on z/OS

In order to activate the in-memory, you should set the appropriate LSNAMES and INMEM parameters in SMFPRMxx.

In the following example we defined:

- a IFASMF.SYS1.ALLSMF logstream to collect all the SMF records from TYPE 0 to type 255);
- an in-memory resource to record SMF 110 records in memory for real-time processing.

```
LSNAME(IFASMF.SYS1.ALLSMF,TYPE(0:255))  
INMEM(IFASMF.EPV,RESSIZMAX(128M),TYPE(110))
```

In the INMEM definition:

- IFASMF.EPV is the name of the in-memory resource (the IFASMF suffix is mandatory);
- RESSIZMAX is the amount of memory assigned to the in-memory resource;
- TYPE is the list of SMF record types to send to the in-memory resource.⁴

You can use the DISPLAY SMF,M command to display the in-memory resources and the connections to them.

```
d smf,m  
IFA714I hh.mm.ss SMF STATUS  
IN MEMORY CONNECTIONS  
Resource: IFASMF.INMEM30  
No Connections  
Resource: IFASMF.INMEM  
Con#: 0001 Connect Time: yyyy.ddd hh:mm:ss  
ASID: 0029  
Resource: IFASMF.INMEM127  
No Connections  
Resource: IFASMF.SMFDB2.SYSTM1.INMEM  
No Connections
```

Figure 2⁵

In the above example four in-memory resources have been defined but only one of them has an active connection.

⁴ Do not use the DEFAULTLSNAME parameter because, in this case, the records that are written in-memory are not written also to the logstream. With the LSNAMES parameter, the records are written both to the logstream and in-memory.

⁵ From “z/OS: z/OS MVS System Management Facilities (SMF)”, SA38-0667-40



One last thing is necessary: to implement a RACF class-profile named as the in-memory resource (IFASMF.EPV in the above example) inside the class FACILITY and allow a READ access to the EPV STC's user described in Chapter 7.

4.2 Customization on distributed platforms

First of all, you need to install the EPVROOT environment. Into the main folder you'll find the SERVER folder that will permit you to access the EPV web applications, including the EPV Control Center that is the platform that allows you to customize your EPV products.

The procedure to reach the EPV Real Time customization is very easy:

- Start the EPV application server in the SERVER folder (Figure 3).

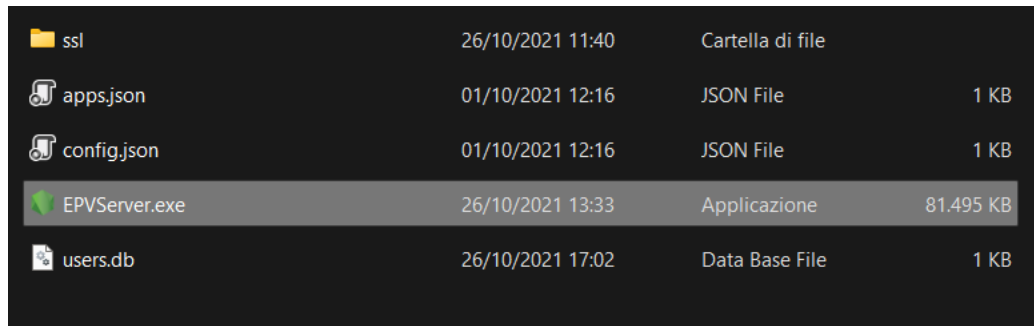


Figure 3

- Enter the web interface with your browser connecting to <http://yourIP:port> or localhost.
- Login through the EPV Hub authentication platform or signup (Figure 4).

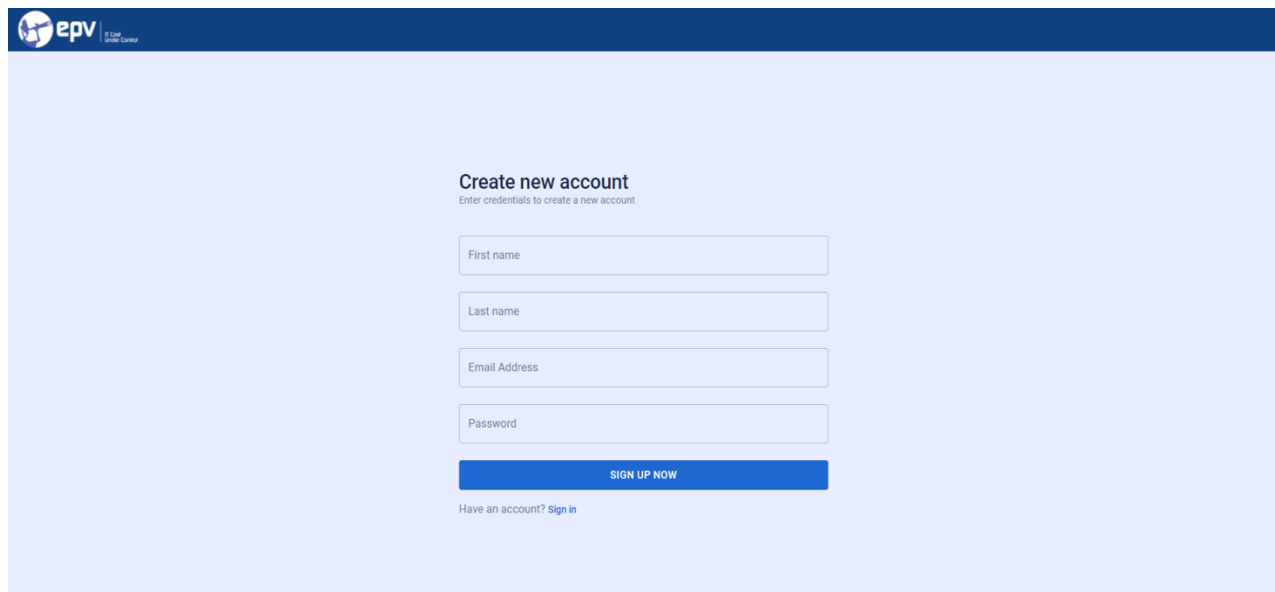


Figure 4



- Access the EPV Control Center web application (Figure 5-6).

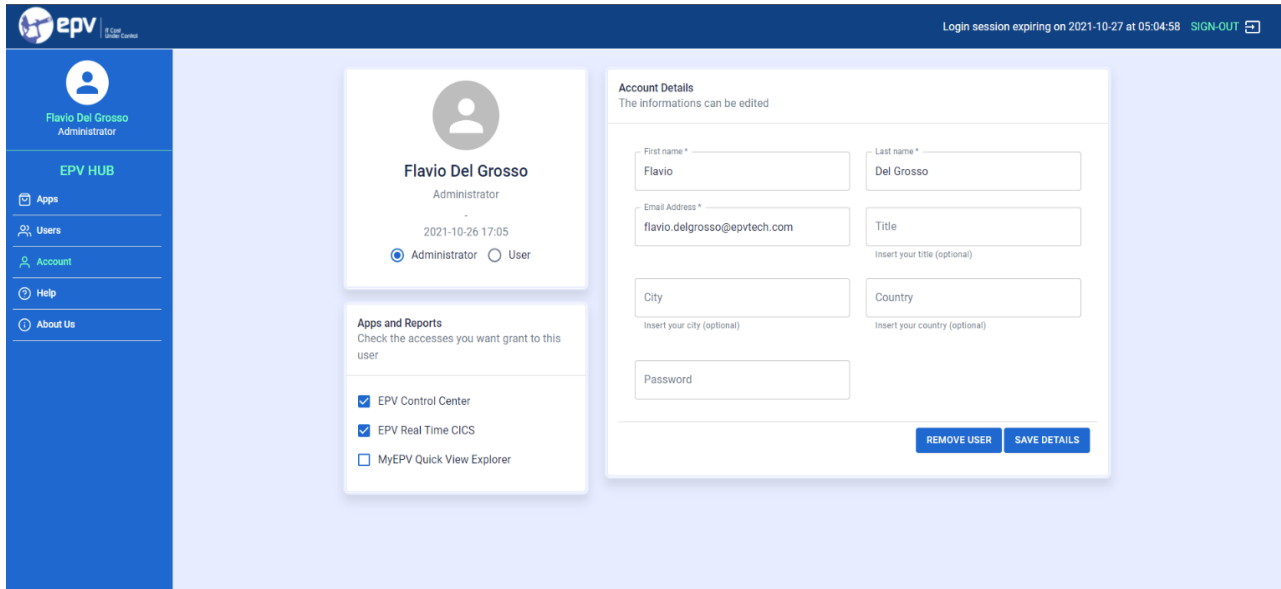


Figure 5

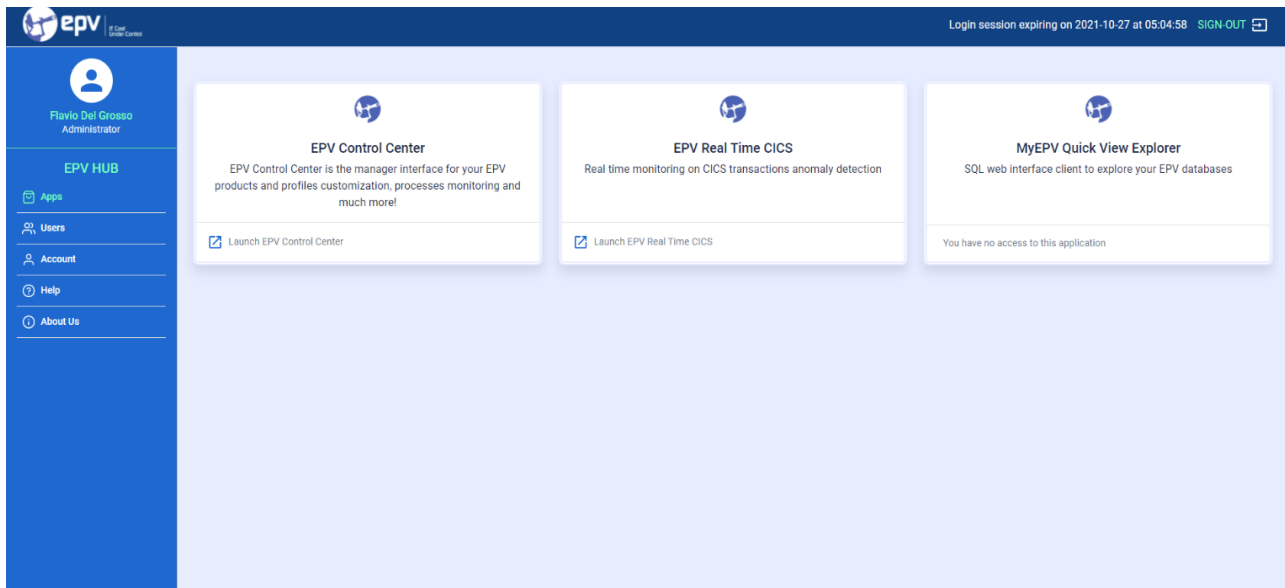


Figure 6

- Choose your profile if not set yet.
- Enter the Settings menu and define your database connection parameters (Figure 7).

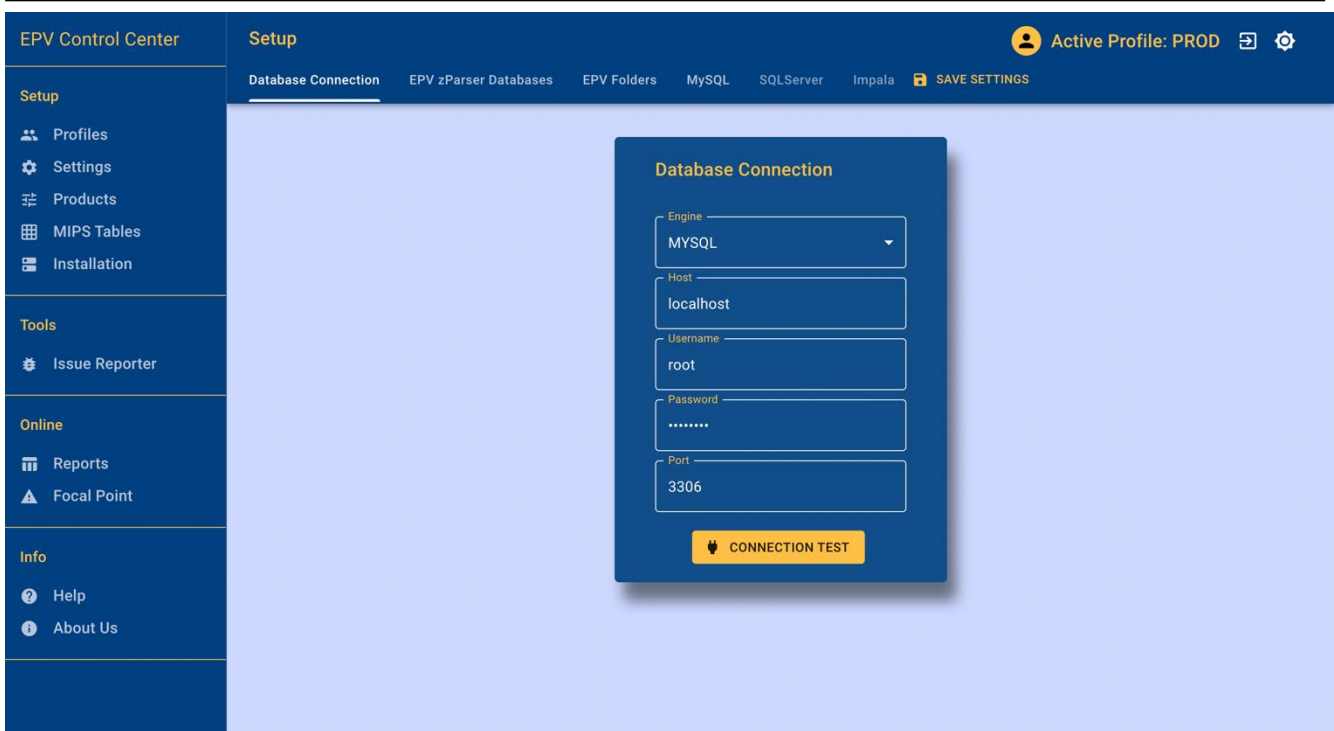


Figure 7

- Enter into the EPV Real Time card from the home page.
- Go to Real Time Config (Figure 8).

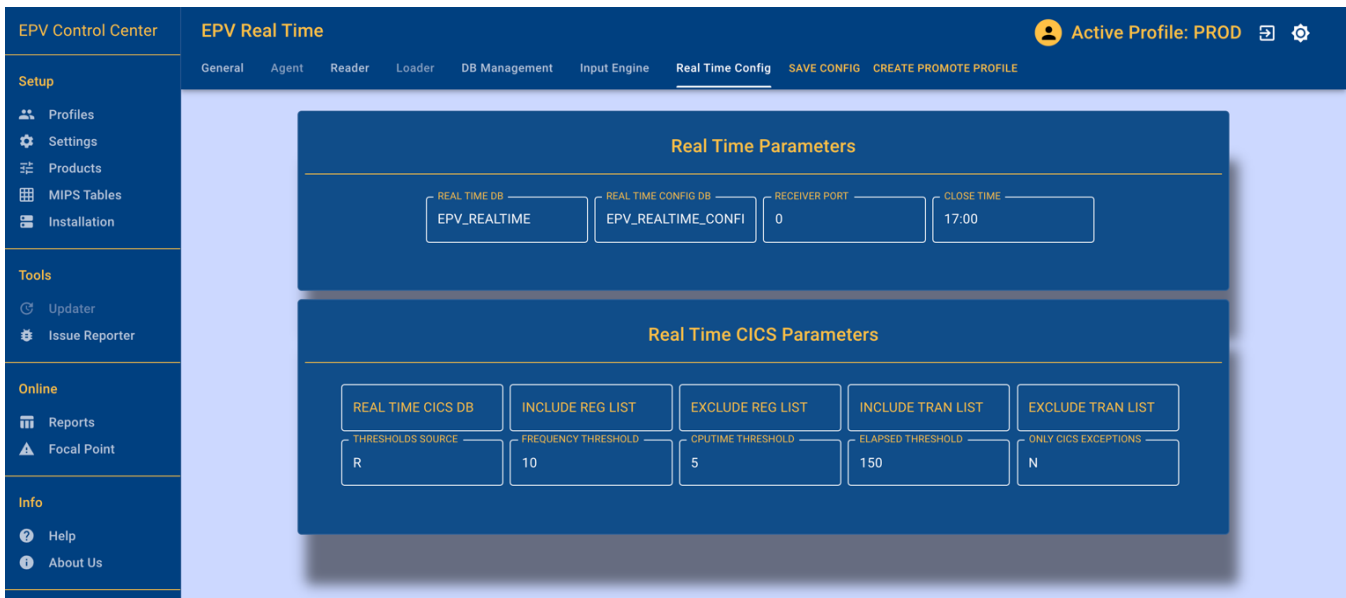


Figure 8



- Set the RECEIVER PORT used by the Socket Server different from 0.
- You can customize also the input folder where you'll write the STARTBTC to start the EOD (end of day). Navigate through the EPV Real Time Input Engine menu and access the General Settings (Figure 9).

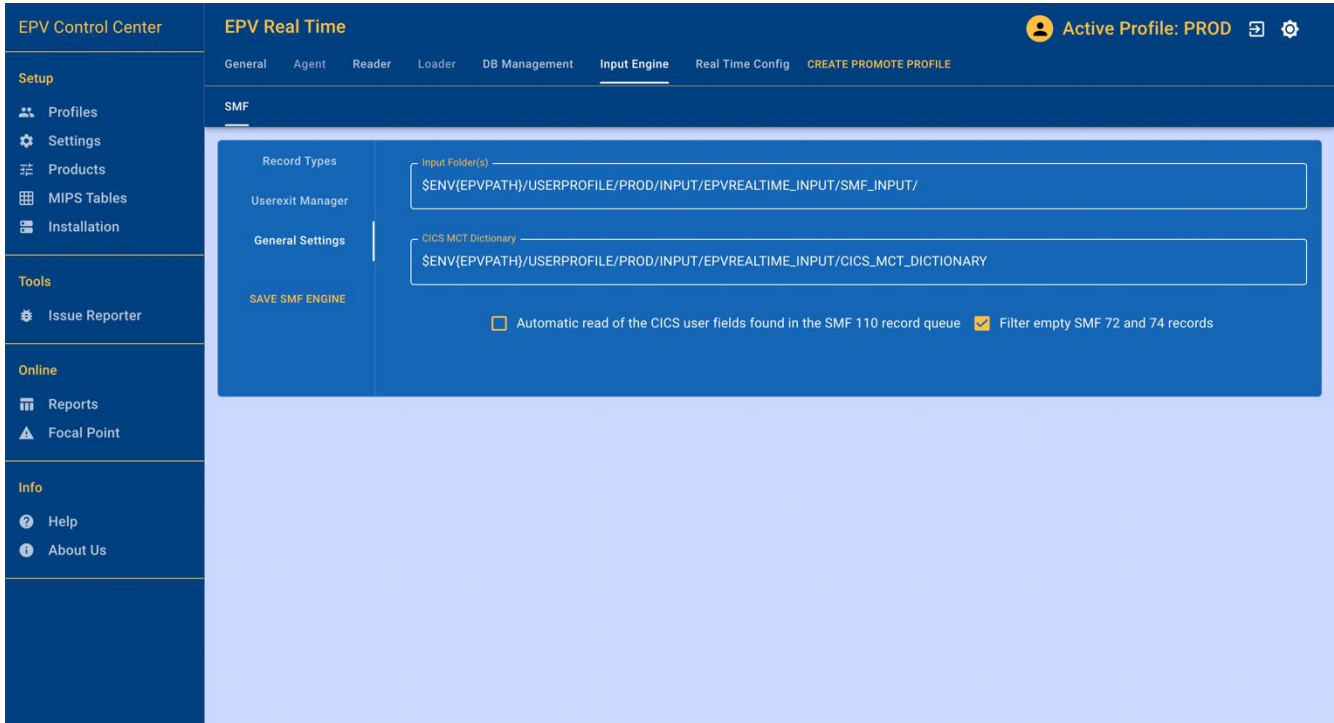


Figure 9

- Eventually, if you need to change some parameter, then after modifying them click on SAVE CONFIG.
- Run the EPVRealTimeConfigDBUpdater_RestoreDefault procedure located in your \$EPVROOT/USERPROFILE/\$PROFILE/EPVREALTIME/PROCS. This will allow to create zParser like databases and manage the switch between each databases at the end of day.

You can find the EPV Real Time parameters in \$EPVROOT/USERPROFILE/\$ProfileName/EPVREALTIME/CONFIG.PL:

- \$EPVDBS->{RT}; Real Time Database.
- \$EPVDBS->{RT_CONFIG}; Real Time Config Database.
- \$RT_SocketReceiver_PORT; Local server receiver port.
- \$RT_CloseTime; Real Time Close Time (00:00 to 23:59).
- \$DBMAXEOD; Number of Staging DBs.
- \$Max_threads; Maximum number of parallel SMF reading threads; when set to 0, the default value used is 20.

Another parameter used by EPV Real Time is @SMFLOGS that you can find in \$EPVROOT/USERPROFILE/\$ProfileName/EPVREALTIME/SMF/SpecificForSMF.PL where you



can specify the folder used by EPV Real Time Socket Server to store SMF data coming from the mainframe.

The default Close Date Time is set to 17:00. This time is when the Socket Server will stop, but you can customize it. Make sure to schedule the starting of the process on a daily basis on a time you prefer.

Remember to schedule the creation of an empty flag file named STARTBTC (no extension required) after midnight. The file should be created in the folder defined in @SMFLOGS. When this file is received by EPVRealTimeAgent, EPVRealTimeEndOfDay is started and the following activities are performed:

- Staging database switch, for example from EPV_REALTIME_1 to EPV_REALTIME_2.
- New stage database clear: following the previous example, the EPV_REALTIME_2 database is cleared by EPVRealTimeDbClear, which means that all the tables inside it are dropped and recreated from scratch.
- Delta calculation for some metrics that needs to be normalized, for example SMF 30 subtype 6; the program name that handles this is EPVRealTimeDbDeaccum and, if you switch from EPV_REALTIME_1 to EPV_REALTIME_2, data is normalised in EPV_REALTIME_1.



6 Running

EPV Real Time Agent

To start it, you need to run EPVRealTimeAgent.sh/BAT placed in \$EPVROOT/USERPROFILE/\$ProfileName/EPVREALTIME/PROCS/AGENT_PROCS

EPV Real Time Socket Server

To start it, you EPVRealTimeSocketServer.sh/BAT placed in \$EPVROOT/USERPROFILE/\$ProfileName/EPVREALTIME/PROCS/SOCKET_PROCS

Realtime component on z/OS

The following JCL needs to be executed to run the EPV Real Time component on z/OS.

```

//EPVSMFH  PROC
//*****
//*
//* PROPRIETARY STATEMENT:
//*   LICENSED MATERIALS - PROPERTY OF EPV
//*   "RESTRICTED MATERIALS OF EPV"
//*   COPYRIGHT EPV TECHNOLOGIES 2020 2025
//*
//* DESCRIPTIVE_NAME:
//*   READ INMEMORY SMF RECORDS AND SEND THEM TO SOCKET SERVER
//*
//*****
*
//IEFPROC  EXEC PGM=EPVSMFH,REGION=0M,TIME=1440
//*   SYSIN='xxx.xxx.xxx.xxx,yyyy,EPV,25'
//*   xxx.xxx.xxx.xxx IP ADDRESS
//*   yyyy           PORT NUMBER
//*   EPV           2.ND LEVEL QUALIFIER OF IN-MEMORY RESOURCE
//*   25           NUMBER OF RECORD ACCUMULATED ON BUFFER
//STEPLIB DD DSN=userHLQ.ASMLIB,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSIN DD *
127.98.97.120,8989,EPV,25
/*

```

Figure 9



The parameters you need to customize in the SYSIN DD are:

- the Socket Server IP address;
- the Socket Server IP port;
- the in-memory resource's second qualifier (EPV in the above example);
- number of records to accumulate before executing the send command; this number has to be set depending on customer workload, configuration and acceptable delay in information availability; the maximum number of records to accumulate is 50.

To start the Address Space, you need to execute the following command:

- S EPVSMFH (or the address space name you choose).

In the next figure you can see an example of the information provided in the started task's SYSOUT:

```
-----EPVSMFS  start-----
      with this parameter
IP.....:127.98.97.120
PORT.....:8989
IN-MEMORY:IFASMF.EPV
N.RECORDS:25
KB allocated for
buffer:1632
-----
Initialize RC=0
CONNECT attempt to IFASMF.EPV begins now...
CONNECT attempt made.
Connect RC=0000 RSN=0000
Validating connect token, RC=0
Test connection with Socket Server=127.98.97.120: 8989
attempt n.:1
Test connection with 127.98.97.120:8989 successfully
2020:10:27 11:00:00.226 ->start write - 2020:10:27 11:00:00.639 ->end
write, send 289806
2020:10:27 12:00:00.254 ->start write - 2020:10:27 12:00:00.464 ->end
write, send 289838
```

Figure 10

To stop the Address Space, you need to execute the following command:

- P EPVSMFH (or the address space name you choose).



7 Customer support

For any technical problem with or question about EPV Products, please write an e-mail to:

epv.support@epvtech.com

For any other issue about EPV products, please write an e-mail to:

epv.info@epvtech.com



Related documentation

The following manuals complement the information provided in this manual:

- *EPV zParser V15 Installation and Customization*