EPV for zLINUX
Preparing Input for a Demo

IT Cost
Under Control

Supporting
EPV for zLINUX version 15

November 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
WEB: http://www.epvtech.com

Contents

About this manual

This manual is intended to help anyone who wants to provide the data needed to prepare an EPV for zLINUX 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/LINUX 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/LINUX input data, a simple four step process to do that is presented.

2 Required z/VM data

The following CP monitor record types are mandatory in order to run EPV for z/LINUX. If you don't provide some of them EPV will produce a partial output or, in some cases, no output at all.

They are:

DOMAIN	RECORD	DESCRIPTION
0, SYSTEM DOMAIN	1	System Data (per processor)
0, SYSTEM DOMAIN	2	Processor Data (per processor)
0, SYSTEM DOMAIN	3	Real Storage Data (Global)
0, SYSTEM DOMAIN	5	Expanded Storage Data (per processor)
0, SYSTEM DOMAIN	16	CPU Utilization Data in a Logical Partition
0, SYSTEM DOMAIN	17	Physical CPU Utilization for Logical Partition Management
0, SYSTEM DOMAIN	20	Extended Channel Measurement Data (Per Channel)
1, MONITOR DOMAIN	4	System Configuration Data
1, MONITOR DOMAIN	5	Processor Configuration Data
1, MONITOR DOMAIN	6	Device Configuration Data
1, MONITOR DOMAIN	7	Memory Configuration Data
1, MONITOR DOMAIN	8	Paging Configuration Data
1, MONITOR DOMAIN	9	Sample Profile
1, MONITOR DOMAIN	17	Expanded Storage Data
3, STORAGE DOMAIN	4	Auxiliary Storage Management
4, USER DOMAIN	3	User Activity Data
4, USER DOMAIN	4	User Interaction Data
5, PROCESSOR DOMAIN	13	CPU-Measurement Facility
6, I/O DOMAIN	3	Device Activity
6, I/O DOMAIN	4	Cache Activity

Figure 1

3 Preparing z/VM input data

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.

3.1 Create a monitor saved segment

CP monitor records have to be written in what is called a "DCSS monitor saved segment".
If you already created a monitor saved segment go to Chapter 3.2.
If you are not sure if a monitor saved segment already exists you can check it by issuing the following privileged command: Q NSS NAME MONDCSS.

Before defining the segment you must verify that enough contiguous free space is available on the spool by issuing the following command:

Q NSS MAP ALL

```
q nss map all
FILE FILENAME FILETYPE MINSIZE   BEGPAG ENDPAG TYPE CL £USERS PARMREGS VMGROUP
0016 SCEEX    DCSS       N/A     02100  028FF  SR   A  00000  N/A      N/A
0028 CMS      NSS     0000256K   00000  0000D  EW   A  00007  00-15    NO
                                 00020  00023  EW
                                 00F00  013FF  SR
0023 ZCMS     NSS     0000256K   00000  0000D  EW   A  00000  00-15    NO
                                 00020  00023  EW
                                 00F00  013FF  SR
0027 NLSUCENG DCSS       N/A     02000  020FF  SR   A  00000  N/A      N/A
0026 NLSAMENG DCSS       N/A     02000  020FF  SR   A  00004  N/A      N/A
0025 HELPSEG  DCSS       N/A     00C00  00CFF  SR   A  00000  N/A      N/A
0021 NLSKANJI DCSS       N/A     02000  020FF  SR   A  00000  N/A      N/A
0002 GCS      NSS     0000256K   00000  0000C  EW   R  00000  OMITTED  YES
                                 00400  0044E  SR
                                 0044F  0044F  SW
                                 00450  005FF  SN
                                 01000  0101A  SR
                                 0101B  011FF  SN
0018 PERFOUT  DCSS       N/A     08A00  08FFF  SN   A  00000  N/A      N/A
0017 SCEE     DCSS       N/A     00900  009FF  SR   A  00000  N/A      N/A
0014 CMSDOS   DCSS-M     N/A     00B00  00B0C  SR   A  00000  N/A      N/A

                                                      MORE...  HERCZVM1
```

Figure 4

You MUST choose a free memory range looking at the BEGPAG and ENDPAG columns

To define a Monitor Saved Segment you have to send the following privileged commands[1]:

- `CP DEFSEG MONDCSS 9000-CFFF SC RSTD`
- `CP SAVESEG MONDCSS`

[1] For more information please visit: http://www.vm.ibm.com/perf/tips/mondcss.html

Please note we have chosen the 9000-CFFF page interval; you can freely choose a free interval space. The command below will show you the entire memory allocation map thus helping you to choose the position where to allocate the MONDCSS saved segment:

VMFSGMAP SEGBLD ESASEGS SEGBLIST

After you have defined the MONDCSS saved segment, with the above parameters, you would have a layout like this[2]:

```
208 M+----------------------+
|                          |  S
|                          |  A
|     Data = rest          |  M
|     4096 pages           |  P   8192 pages
|.....................     |  L
| Config = 4096 pages      |  E
176 M+----------------------+ ---
|   (8124 pages)           |  E
|   Data = rest            |  V
|                          |  E  8192 pages
|.....................     |  N
| Config = 68 pages        |  T
144 M+----------------------+
```

Figure 2

As you can see the segment is split into 4 areas: sample data, sample config, event data and event config. Area size is assigned by using defaults.

However you can set any area size depending on your needs[3].

Example:

Assume the values you need are:

- Sample Data 1886 pages
- Sample Config 350 pages
- Event Data 256 pages
- Event Config 68 pages

First you need to allocate a 10 MB segment to get memory for the total 2560 pages (rounded to the next MB):

```
CP DEFSEG MONDCSS 2200-2ABC SC RSTD
CP SAVESEG MONDCSS
```

Then you have to set partitions size[4]:

[2] This is the layout of the default MONDCSS segment shipped with z/VM starting with z/VM 6.2.0.
[3] See "Calculating the Space Needed for the Saved Segment" (Chapter 9) in z/VM Performance Manual (SC24-6208-04).

CP MONITOR START PARTITION 324

Finally you have to set the config areas size:

CP MONITOR SAMPLE CONFIG SIZE 350
CP MONITOR EVENT CONFIG SIZE 68

The above configuration commands will produce the following memory map:

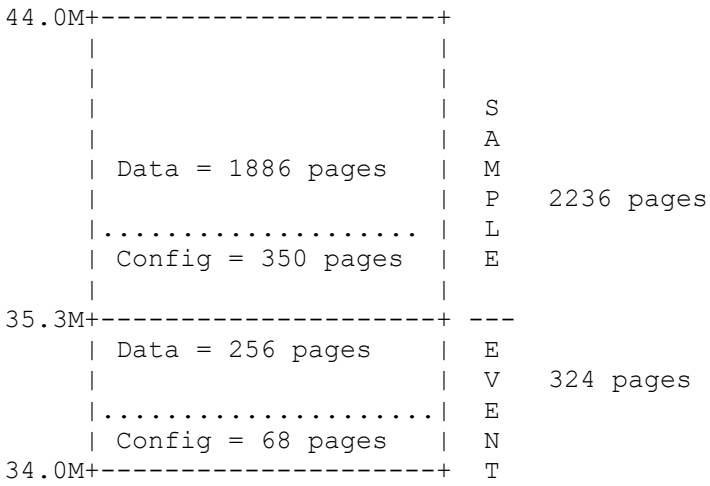

Figure 3

[4] The MONITOR START PARTITION command sets size of EVENT partition with the left over being the Sample size.

3.2 Define the MONWRITE virtual machine

The MONWRITE virtual machine is used to run the MONWRITE IBM-supplied program that can retrieve CP monitor data from the saved segment.

MONWRITE performs the following functions:

- Load the monitor saved segment into its virtual storage;
- Connect to the *MONITOR system service;
- Accesses the saved segment and retrieves its data;
- Writes the monitor data on a tape or in a CMS file.

If you already created the MONWRITE virtual machine go to Chapter 3.3.

In the USER DIRECT file enter the following definitions; they will be used to create the MONWRITE virtual machine:

```
USER MONWRITE MONWRITE 4M 8M BCDEFG
MACHINE ESA
ACCOUNT XXXXX
IPL CMS
OPTION QUICKDSP
SHARE ABSOLUTE 3%
IUCV *MONITOR MSGLIMIT 255
NAMESAVE MONDCSS
CONSOLE 009 3270 T
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK MAINT 0190 0190 RR * CMS
LINK MAINT 019D 019D RR *
LINK MAINT 019E 019E RR *
MDISK 191 3390 3105 090 610RES  MR READ     WRITE    MULTIPLE
```

Save the USER DIRECT then enter the following command to reload configuration and activate the MONWRITE virtual machine:

```
DIRECTXA USER DIRECT
```

3.3 Create and run the MONWRITE PROFILE EXEC

If you already created a MONWRITE PROFILE EXEC, including all the records needed by EPV for zLINUX, go to Chapter 3.4.

Below is an example of the MONWRITE PROFILE EXEC customized to start only the SAMPLE monitoring records needed by EPV for zLINUX.

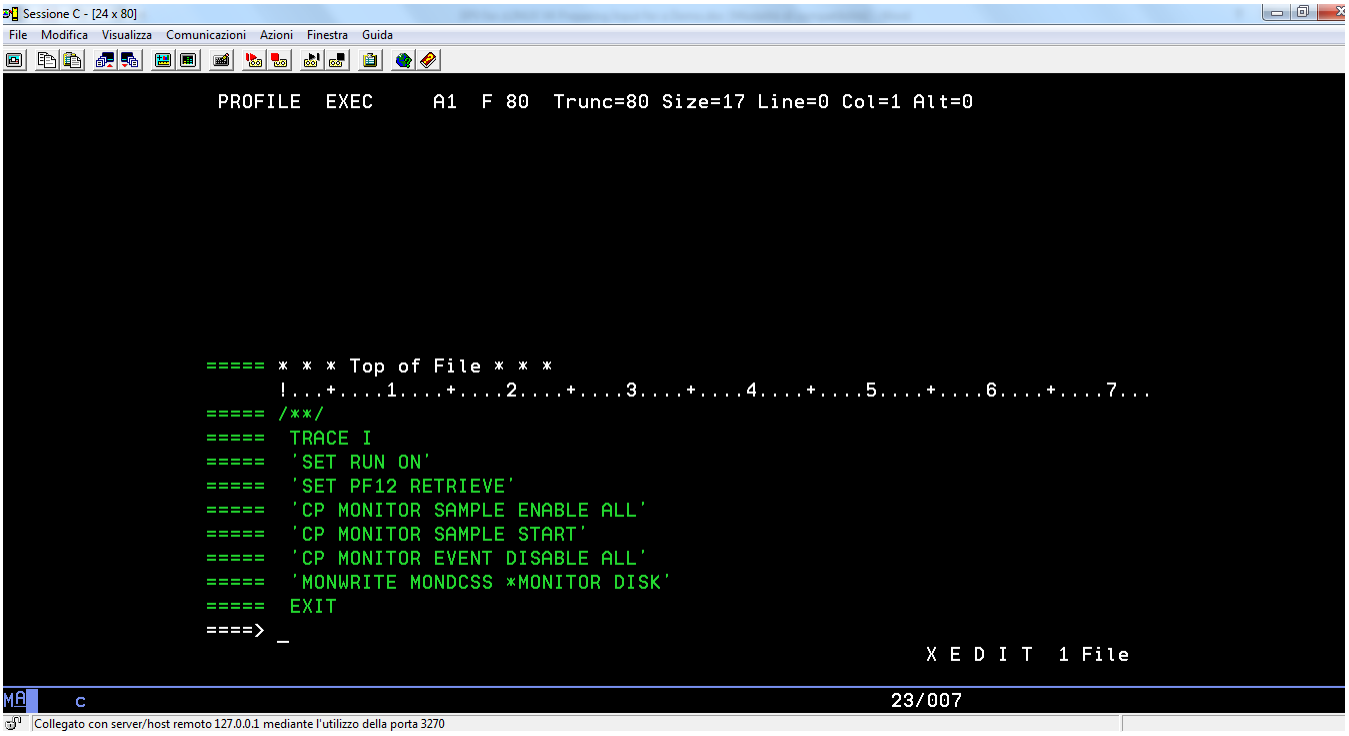

Figure 5

The "MONWRITE MONDCSS *MONITOR DISK" command tells the MONWRITE program to save the segment contents in the MONDCSS *MONITOR DISK; by default disk A is used.

If you want to write the records on another disk, specifying it with the FMode parameter, you should issue the "MONWRITE MONDCSS *MONITOR DISK FName FType FMode" command.

To edit the PROFILE EXEC logon to the MONWRITE virtual machine and issue the following command: FLIST PROFILE * *

After the PROFILE was saved with the FF command, exit the FLIST editor and then run the PROFILE EXEC by issuing the PROFILE command.
You should get messages telling you if CP monitor collection has been activated and if MONWRITE has successfully connected to *MONITOR (see an example in Figure 6).

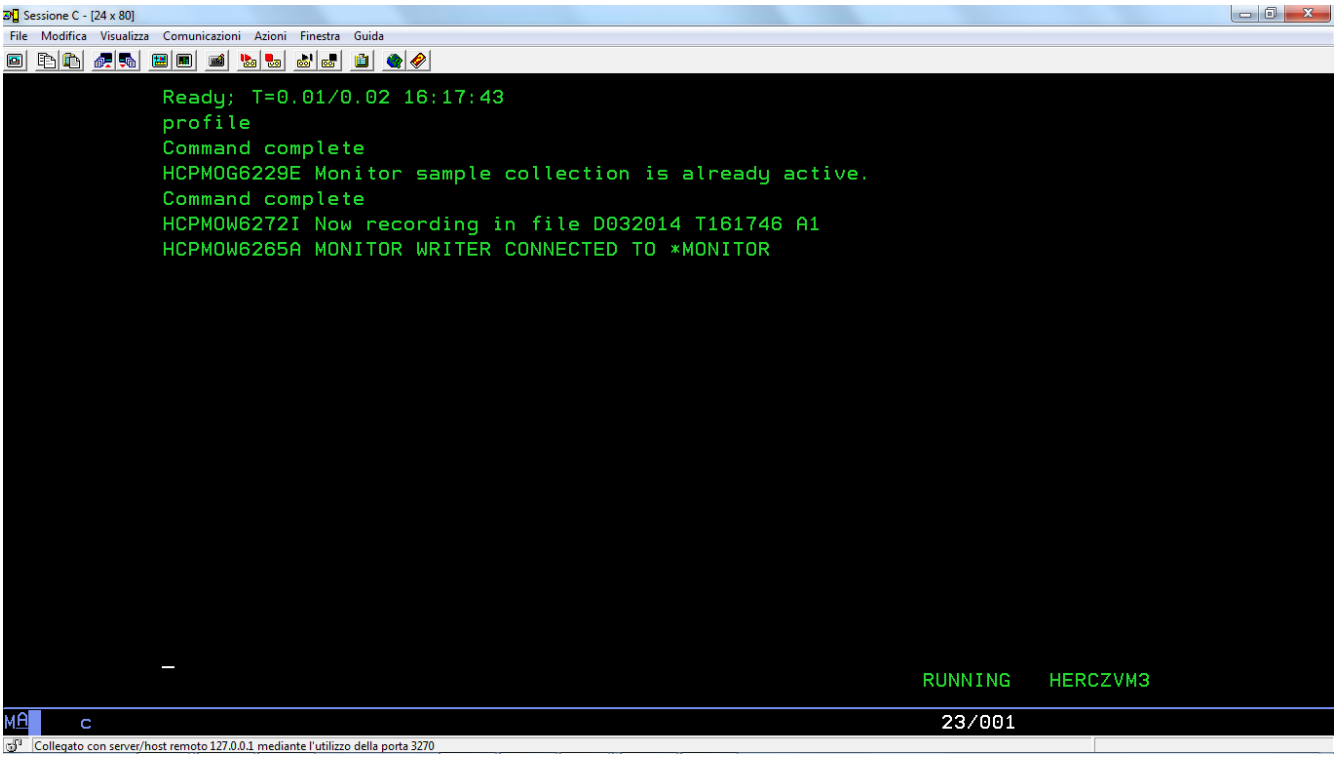
Sessione C - [24 x 80]
File Modifica Visualizza Comunicazioni Azioni Finestra Guida
Ready; T=0.01/0.02 16:17:43
profile
Command complete
HCPMOG6229E Monitor sample collection is already active.
Command complete
HCPMOW6272I Now recording in file D032014 T161746 A1
HCPMOW6265A MONITOR WRITER CONNECTED TO *MONITOR
RUNNING HERCZVM3
MA c 23/001
Collegato con server/host remoto 127.0.0.1 mediante l'utilizzo della porta 3270

Figure 6

3.4 Stop data collection

To stop collecting data you have to issue the following command from the z/VM console:

CP MONITOR STOP

To stop also MONWRITE you must issue the **MONWSTOP** command from the MONWRITE virtual machine.

3.5 Transfer the collected data to PC

Logon to the MONWRITE virtual machine and issue the “FLIST * * a” command. You will get a list of files.

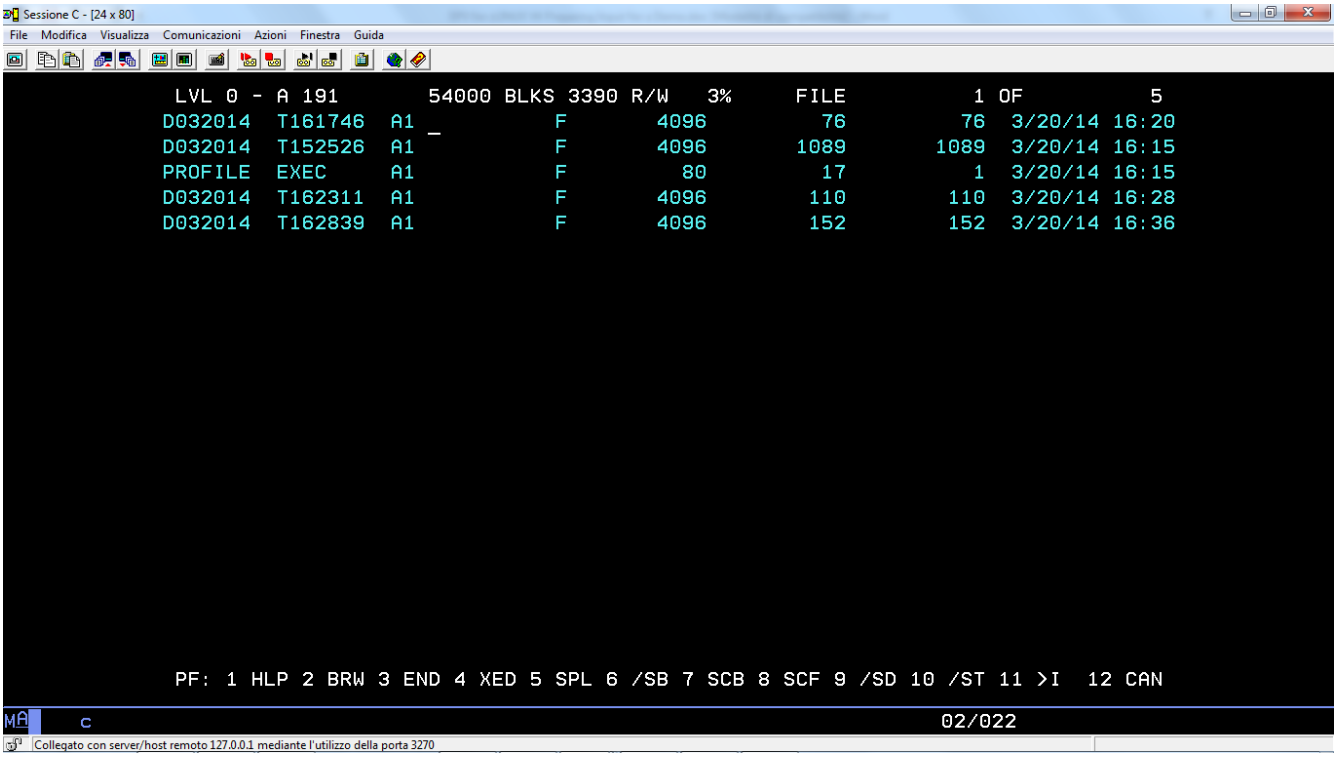

Figure 7

The files containing CP monitor records are listed as Dmmddyy Thhmmss (date and time when monitoring was started).

You have to transfer these files to PC in binary mode.

You can use the 3270 emulator, FTP or any other method.
The FTP server is part of TCP/IP stack thus it must be configured and started on the z/VM system.

3.6 Compress 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.

3.7 Send 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 to send a small file by FTP or e-Mail, so we can confirm everything is correct before spending more time sending large amounts of data.

4 Additional input data

To collect Linux process activity data, you have to schedule a shell script running the ps command (see an example in Attachment A) on some zLinux guest.

Process activity data has to be transferred to PC in ASCII mode.

5 Customer support

For any technical problems or questions about EPV for zLINUX please email:

epv.support@epvtech.com

For any other issue about EPV for zLINUX please email:

epv.info@epvtech.com

Attachment A – Collecting process activity

Two scripts must be scheduled in the zLinux crontab of each system you want to collect process activity data and executed regularly:

- epvzlinux_collector.sh, to collect process activity measurements; normally scheduled at 5/10 minutes interval;
- epvzlinux_archive.sh, to send the collected measurements to the server where EPV for z/LINUX runs; normally scheduled once a day just after midnight.

An example of both scripts is provided here. It is also included in the TOOLS folder of the installation CD.

1) epvzlinux_collector.sh script

```
#!/bin/bash
#
#########################################################
# NOTE: please set this parameters
#########################################################

# full path of data directory to store process activity measurements
CURRENT_DIR=/home/epvuser/PRC/CURRENT

# interval value - it must corresponds to what you set in system crontab
INTRVL=300

#########################################################
# WARNING: Do not modify below this line
#########################################################

HOST=`hostname`
DATE=`date +%y%h%d`
DATETIME=`date +%d/%m/%y:%H:%M:%S`
FILENAME=$CURRENT_DIR/$HOST-$DATE".zlnxprc"

CPU=`cat /proc/cpuinfo | grep 'processor\>' | wc -l`

echo "***ps*** $HOST $DATETIME INT=$INTRVL CPU=$CPU" >> $FILENAME
echo "***header*** ruser comm ppid pid time vsz" >> $FILENAME

ps -Ao "ruser=fullusername" -o "comm ppid pid time vsz" | awk ' $5 != "00:00:00" && $1 !=
"fullusername" {printf ("%s;%s;%s;%s;%s;%s\n",$1,$2,$3,$4,$5,$6) }' >> $FILENAME
```

The script scheduling has to be defined in crontab as follows:

*/5 * * * * /home/epvuser/epvzlinux_collector.sh > /home/epvuser/PRC/CURRENT/zlinux_prc.log 2>&1

This entry will run the collector script each day every 5 minutes.

2) epvzlinux_archive.sh script

```
#!/bin/bash
#
#########################################################
# NOTE: please set this parameters
#########################################################

# full path of data directory where process activity measurements are collected
CURRENT_DIR=/home/epvuser/PRC/CURRENT

# full path of data directory where process activity measurements are archived
ARCHIVE_DIR=/home/epvuser/PRC/ARCHIVE

# days to keep files in archive directory
DAYS_TO_MAINTAIN=7

# remote host address
REMOTE_HOST=192.168.1.1

# remote username
REMOTE_USR=yourusr

# remote password
REMOTE_PWD=yourpwd

# remote directory
REMOTE_DATA=remote_archive

#########################################################
# WARNING: Do not modify below this line
#########################################################

function proc_ftp {
file_toftp=$1
file_local=$2
ftp -n $REMOTE_HOST <<ENDFTP 2> $CURRENT_DIR/ftplog.log
user $REMOTE_USR $REMOTE_PWD
prompt off
hash on
```

```
cd $REMOTE_DATA
lcd $CURRENT_DIR
put $file_local $file_toftp
quit
ENDFTP
}
cd $CURRENT_DIR

HOST=`hostname`
filegroup=(`ls | grep "\.zlnxprc"`)

for file in ${filegroup[@]}; do

zlnxfile=$CURRENT_DIR/$file
zlnxfile_arch=$ARCHIVE_DIR/$file

filedate_raw=`expr match "$file" '.*\([\-][0-9][0-9][A-Z][a-z][a-z][0-9][0-9][\.]*\)'`
filedate=`expr match "$filedate_raw" '.*\([0-9][0-9][A-Z][a-z][a-z][0-9][0-9]*\)'`

fyearet=${filedate:0:2}
fmonlet=${filedate:2:3}
fday=${filedate:5:2}

case "$fmonlet" in
Jan ) fmon=01;;
Feb ) fmon=02;;
Mar ) fmon=03;;
Apr ) fmon=04;;
May ) fmon=05;;
Jun ) fmon=06;;
Jul ) fmon=07;;
Aug ) fmon=08;;
Sep ) fmon=09;;
Oct ) fmon=10;;
Nov ) fmon=11;;
Dec ) fmon=12;;
esac

let"fyear=($fyearet+2000)"
mday=`date +%d`
mon=`date +%m`
year=`date +%Y`

currdate_fmt=$year$mon$mday
filedate_fmt=$fyear$fmon$fday

if [[ $filedate_fmt < $currdate_fmt ]]; then
```

zlnxfile_toftp="$HOST.zlnxprc"
proc_ftp $zlnxfile_toftp $file
mv $zlnxfile $zlnxfile_arch

fi

done

find $ARCHIVE_DIR \(-name '.zlnxprc' \) -mtime $DAYS_TO_MAINTAIN -exec rm {} \;*

exit

The script scheduling has to be defined in crontab as follows:
0 0 * * * /home/epvuser/zlinux_prc_archive.sh > /home/epvuser/PRC/CURRENT/zlinux_prc_tran.log 2>&1

This entry will run the archive script every day at midnight.

Related documentation

The following manuals complement the information provided in this manual:

- *EPV for zLINUX V15 Installation and Customization Guide*
- *EPV for zLINUX V15 Database Layout*
- *EPV for zLINUX V15 Release Notes*
- *EPV for zLINUX V15 List of Views*
- *EPV V15 User Interface*
- *EPV V15 Operations Guide*