zWorkload Reporter

COMPRESS TOOL

Installation Guide

Author: zProducts
Company: SVA System Vertrieb Alexander GmbH
Borsigstraße 26
65205 Wiesbaden

Date: 06.09.2022
Version: 2.2

Table of content

1 Pre notes

This chapter describes important information about SMF data.

1.1 Required SMF Data

To use all reports of the zWorkload Reporter, the SMF record types 30, 70-74, 78 & 113 with a set of subtypes are required.

Record Type 30 (1E) — Common address space work

// Subtype 2 - Activity since previous interval ended

// Subtype 3 - Activity for the last interval before step termination

Record Type 70 (46) — RMF Processor Activity

// Subtype 1 - Contains measurement data for general and special purpose processors, logical partitions, and internal coupling facilities

Record Type 71 (47) — RMF Paging Activity

Record Type 72 (48) — Workload Activity, Storage Data, and Serialization Delay

// Subtype 3 - Workload Activity - is written for each service class and active report class in the active service policy. A report class becomes active as soon as work has been assigned to that report class.

Record Type 73 (49) — RMF Channel Path Activity

Record Type 74 (4A) — RMF Activity of Several Resources

// Subtype 2 - XCF Activity

// Subtype 4 - Coupling Facility Activity

Record Type 78 (4E) — RMF Activity of Several Resources

// Subtype 2 - Reports virtual storage activity. It contains a Common Storage data section and may contain one or more Private Area data sections.

Record type 113 (71) — Hardware capacity, reporting, and statistics

// Subtype 2 - Contains hardware data event counters for IBM® System z10 or later CPCs.

1.2 SMF 30 & 70 synchronization

It is essential for some zWR reports that SMF 30 and 70 record types are synced to the same interval. Otherwise, some reports can show confusing graphs. Therefore, please check the z/OS parmlib member → SMFPRMxx.

```
…
INTVAL(15)                      /* SMF INTERVAL 15                 */
LISTDSN                         /* LIST DATA SET STATUS AT IPL     */
SYS(TYPE(30,70:79,89,99(14),113),
    EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT,
          IEFUSI,IEFUJI,IEFU29),INTERVAL(SMF,SYNC),NODETAIL)
```

Further information about SMF synchronization can be found at the official IBM documentation.

Link: https://www.ibm.com/docs/en/zos/2.4.0?topic=intervals-defining-smf-synchronization

1.3 SMF 113 (HIS) synchronization

We recommend to synchronize the SMF 113 recording interval to the global SMF recording interval. Otherwise some confusing or incomplete SMF 113 zWR graphs might be displayed. Since SMF 113 records are written by the HIS address space, the write out differs by default from the global SMF recording interval. To synchronize SMF 113 recording with global SMF recording interval it is necessary to start the HIS address space and execute following modify command with SI=SYNC option:

```
/F HIS,...,SI=SYNC"
```

Further information about SMF 113 synchronization can be found at the official IBM documentation.

Link: https://www.ibm.com/docs/en/zos/2.4.0?topic=command-starting-configuring-stopping-his-data-collection

2 Installation

The Compress Tool is responsible for compressing the SMF records before sending them to the zWorkload Reporter (zWR) server.

The installation of the Compress Tool is very straightforward and can be done in a few steps. The software is distributed in xmit format compressed into a zip file.

For the Compress Tool you need Java installed with a minimum version 7 to run the Compress Tool utility.

2.1 Upload xmit file

Please unpack the file 'zCompressV2R2.zip' on your workstation. The uncompressed folder contains a xmit file 'Compress.V2R2.xmit' that you must upload to your z/OS system where you can access all required SMF records.

1. Allocate a sequential dataset with the following attributes for the xmit file:

```
6 Cyls
RECFM=FB
LRECL=80
BLKSIZE=3120
```

2. Upload Compress.V2R2.xmit into the new dataset in binary mode:

```
ftp <your_host>
bin
cd <directory_of_Compress.V2R2.xmit>
put Compress.V2R2.xmit 'YOUR.DATASET.XMIT'
```

3. Receive the Dataset using the following TSO command:

```
RECEIVE INDA('YOUR.DATASET.XMIT')
```

Respond to the Message INMR906A "Enter restore parameters or 'DELETE' or 'END' " with the dataset name you wish to deploy the Compress Tool installation library to.

```
DA('YOUR.DATASET.COMPRESS.V2R2.INSTALL')
```

2.2 Installation Jobs

Please browse into YOUR.DATASET.COMPRESS.V2R2.INSTALL and you can see 5 member.

#README	can be ignored because it includes information included in this document
A0#UNPKG	JCL to receive XMIT#JAR and XMIT#JCL
COMP#FTP	JCL to select, compress and send SMF records to the zWR server
EXP#SFTP	JCL example for SFTP transfer
XMIT#JAR	XMIT that contains the binary Java code library of the Compress Tool
XMIT#JCL	XMIT with further installation JCL

2.2.1 A0#UNPKG

Please edit the JCL, as described in the header of the member, and submit the job to receive XMIT#JAR and XMIT#JCL.

The JCL contains two placeholders that must be replaced by the correct values.

$INPUT	This is the DSN of the current dataset
$HLQ	This is the installation HLQ for the Compress Tool of your choosing

After successfully receiving both member please browse in the new JCL library $HLQ.ZBSS.COMPRESS.V2R2.JCL

2.2.2 C1#UPDAT (only when updating V1 to V2R2)

When you already have the Compress Tool V1 installed, you only need to replace the old JAR with the new V2R2 JAR dataset using this JCL. If this is the first installation, please skip this step and continue with the next one.

Edit the member C1#UPDAT and submit the job. Please replace the following placeholder with the correct values of your installation

$ZFSDIR	This is the USS (OMVS) path to your installed V1 “Compress.jar”
$HLQ	This is the installation HLQ for the Compress Tool of your choosing

2.2.3 A1#ZFS

Please review, edit this JCL, and submit the job to create a new ZFS container, mount the ZFS in Unix System Services (OMVS) and copy the JAR dataset into the ZFS. Java cannot access JAR libraries located in a z/OS dataset and therefore must be installed into a ZFS.

Please replace the following placeholders.

$HLQ	This is the installation HLQ for the Compress Tool of your choosing
$ZFSDIR	This is the USS (OMVS) path that the new ZFS should be mounted at
$ZFSNAME	The DSN for the new ZFS
$ZFSVOL	The volser for the ZFS allocation. If SMS managed, you can remove this parameter from the JCL

If the job ended with return code 0 the Compress Tool was installed successfully. Please review and add the new ZFS mount to your BPXPRMxx member. If the Compress Tool ZFS mount point is located on the version root file system, it might be necessary to add the creation of the directory to your z/OS post installation routines or your user mods.

2.2.4 B1#CPJXX (optional)

If the Java Batch Launcher (JVMLDMxx) is not installed into SYS1.SIEALNKE or into another Linklist Library, then you can use the JCL to install the module JVMLDMxx .

Please edit the job and update the paths and the Java version according to your installation. It is not important, if your installation uses a 31-bit or 64-bit Version. We recommend using the 64-bit Version for larger Datasets. The provided JCL uses Java 8.0 and the standard USS paths.

```
//COPY      EXEC PGM=BPXBATCH
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//STDIN     DD DUMMY
//STDPARM   DD *
SH  cp -X /usr/lpp/java/J8.0/mvstools/JVMLDM80
    "//'SYS1.SIEALNKE(JVMLDM80)'"
```

After the job runs successfully, please update the LLA via the following command

```
F LLA,REFRESH
```

2.2.5 A2#COMPR (optional)

This is a IVP job to verify the Compress Tool is installed correctly. You can skip this step and instead use the JCL COMP#FTP to compress and ftp one day of smf data to verify the installation.

Please edit the member and replace the values for ZBSS_HOME and JAVA_HOME according to your installation.

```
//ABNLIGNR DD DUMMY
//STDENV   DD *
export ZBSS_HOME=/usr/lpp/zbsscompress
export JAVA_HOME=/usr/lpp/java/J8.0
```

3 Compress and FTP

After the installation of the Compress Tool is finished, please return to the Installation library YOUR.DATASET.COMPRESS.V2R2.INSTALL and review the member COMP#FTP.

This is the sample JCL that you can use for the periodical SMF upload to the zWR server. It selects the required SMF record types and subtype from your SMF dump datasets, compresses them and finally sending them to our server. If you are using another transportation protocol like sftp or ftps or if you are using an inhouse ftp relay for all outgoing file transfers, you must adapt the last step accordingly.

Please review and edit the JCL COMP#FTP. The following placeholders must be replaced.

<CUST>	Customer name
<DATETIME>	Timestamp. Needs to be changed every day dynamically. For example, by the Scheduler software
<SMFIN>	Input SMF dump datasets containing the records of the last day
<SMFOUT.GZIP>	Output GZIP dataset that was created by the Compress Tool
<IPADDRESS>	The IP address of the zWR server
<PORT>	The FTP port number that was provided by us
<JOBCARD>	Please add a valid job card
<HOME>	Your USS Home path

3.1 GZIP encryption

If it is necessary to encrypt outgoing data, it is possible to encrypt the GZIP on your side with GnuPG mechanism. Therefore, you need a system which can encrypt with GnuPG. Please contact the SVA support team for a corresponding key. This key is essential for encrypting the GZIP.

3.2 Sort step

The SORT step selects the correct SMF records.

```
//*  ============================================================
//*  SMF RECORDS EXTRACT
//*  ============================================================
//SORT     EXEC PGM=SORT
 //SYSOUT   DD SYSOUT=*
 //SORTWK01 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(250,250))
 //SORTWK02 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(250,250))
 //SORTWK03 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(250,250))
 //SORTIN   DD DISP=SHR,DSN=<SMFIN>
 //SORTOUT  DD DISP=(,KEEP),SPACE=(CYL,(999,999),RLSE)
 //SYSIN    DD    *
   INCLUDE COND=((6,1,BI,EQ,30,AND,23,2,BI,EQ,2),OR,
                 (6,1,BI,EQ,30,AND,23,2,BI,EQ,3),OR,
                 (6,1,BI,EQ,70,AND,23,2,BI,EQ,1),OR,
                 (6,1,BI,EQ,71),OR,
                 (6,1,BI,EQ,72,AND,23,2,BI,EQ,3),OR,
                 (6,1,BI,EQ,73),OR,
                 (6,1,BI,EQ,74,AND,23,2,BI,EQ,2),OR,
                 (6,1,BI,EQ,74,AND,23,2,BI,EQ,4),OR,
                 (6,1,BI,EQ,78,AND,23,2,BI,EQ,2),OR,
                 (6,1,BI,EQ,113,AND,23,2,BI,EQ,2))
   OPTION COPY,VLSCMP,NULLOUT=RC16
/*
```

3.3 Compress step

Please update the PGM to your Java Batch Launcher version, the OUTPUT DSN and the values for ZBSS_HOME and JAVA_HOME:

```
// IF (SORT.RC = 0) THEN
//*  ============================================================
//*  SMF DATASET COMPRESS TO GZIP
//*  ============================================================
//COMPRESS EXEC PGM=JVMLDM80,REGION=0M,
//   PARM='/+D de.sva.zbss.zoscompress.CompressDataset'
//*
//INPUT    DD DISP=SHR,DSN=*.REBLOCK.SYSUT2
//OUTPUT   DD DISP=(,CATLG),SPACE=(CYL,(999,100),RLSE),UNIT=3390,
//            DSN=<SMFOUT.GZIP>
//*
//STEPLIB  DD DSN=SYS1.SIEALNKE,DISP=SHR    <needs to be checked>
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
//ABNLIGNR DD DUMMY
//STDENV   DD *            <needs to be checked> (paths)
export ZBSS_HOME=/<HOME>/compress
export JAVA_HOME=/usr/lpp/java/current_64
/*
...
// ENDIF              IF (SORT.RC = 0)
```

3.4 Send step

We provide three supported transfer protocols to transfer the GZIP files to our servers. We recommend using SFTP but the other protocols are also possible. The following overview shows the advantagaes and disadvantages for every transfer protocol:

Features	FTP	FTPS ★	SFTP ★ (recommended)
Note	File Transfer Protocol	Also known as: FTP over SSL	Also known as: FTP over SSH (Secure Shell)
Implement strong encryption algorithms	✗ No data encryption	✓ FTPS implements strong algorithms like AES and Triple DES to encrypt file transfers	✓ SFTP implements strong algorithms like AES and Triple DES to encrypt file transfers
Enrcypts usernames and passwords	✗ No username and password encryption	✓ Implicit and explicit authentication mode	✓ Username and passwords are always enrcypted
Supports key-based authentication	✗ No key-based authentication	✗ Key-based authentication not supported	✓ SSH keys can be used to authenticate SFTP connections
Supports certificates	✗ No certificates are supported	✓ Connections are authenticated using certificates	✗ Certificates are not supported
Firewall-friendly	✓ Uses multiple ports for command and data channel in passiv mode	✓ Uses multiple ports for command and data channel in passiv mode	✓ Only one port is used (connection via SSH)

3.4.1 Naming convention recommendations

SVA provides some sample Jobs for transferring the GZIP files. Within the examples there is always the keyword <SMFOUT.GZIP>. This keyword must be changes to the name of the GZIP file that you upload to our servers. There are no specific rules but there are some recommendations:

The file should contain your customer abbreviation and LPAR or CEC name followed by a timestamp. This is not a must but makes it easier for tracking and processing. The suffix of the filename should also be ".gzip". When you encrypt your GZIP file with GnuPG the file suffix should be ".gzip.gpg".

Example: Illustration of the format

```
<CUSTOMER>.<LPAR>.<TIMESTAMP>.gzip.gpg
```

Example: Possible file name

```
SVA.IPO1.D201231.gzip.gpg
```

If you would like to send several files instead of one it is important that individual files have a unique name. Apart from the suffix, you can name the files and send the data in as many different files as you like.

Example: Several files by LPAR

```
SVA.IPO1.D201231.gzip.gpg
SVA.IPO2.D201231.gzip.gpg
SVA.IPO3.D201231.gzip.gpg
```

Example: Several files by CEC

```
SVA.CEC1.D201231.gzip.gpg
SVA.CEC1.D201231.gzip.gpg
SVA.CEC1.D201231.gzip.gpg
```

Example: One file per customer environment

```
SVA.D200615.GZIP
```

3.4.2 Via FTP

In the following example please change the keywords <IPADDRESS>, <PORT>, <FTP-USER> and <FTP-PASSWORD> with the provided information from SVA to transfer the files via FTP. The recommendations regarding the keyword <SMFOUT.GZIP> are described in the chapter Naming convention recommendations.

```
//*  ============================================================
//*  SEND SMF GZIP TO ZWR (FTP EXAMPLE)
//*  ============================================================.
// IF (COMPRESS.RC = 0) THEN
//ZWRFTP  EXEC PGM=FTP,PARM='<IPADDRESS> <PORT>    (EXIT'
//SYSPRINT DD SYSOUT=*
//INPUT    DD *
<FTP-USER> <FTP-PASSWORD>
bin
LOCSITE FWF
put <SMFOUT.GZIP>
quit
// ENDIF              from ZWRFTP
//
```

3.4.3 Via FTPS

FTPS is a extention for FTP to provide cryptography with certificates to secure communications. There are the following prerequisite:

- Keyring for TSO User
- Upload and connect Public Certificate (DTRUST.SSL.CERT) which is included in the archive

To create a Keyring for a TSO User and connect a trusted certificate to this Keyring, following example JCL can be used. Replace ID (SVAZBSS), Ringname (UKEYRING), Label (“Public Certificate”) with your values.

1. Create Keyring

```
//*-------------------------------------------------------
//* ADD NEW KEYRING
//*-------------------------------------------------------
//ADDRING EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 RACDCERT ADDRING(UKEYRING) ID(SVAZBSS)
//
```

2. Allocate Dataset with following attributes

```
5 Tracks
RECFM=VB
LRECL=84
BLKSIZE=27998
```

3. Upload Public Certificate

```
ftp <your_host>
put CERTIFICATE 'YOUR.RACF.CERT'
```

4. Add SITE Certificate with label

```
//*-------------------------------------------------------------------
//* ADD CERT PUBLIC KEY TO KEYRING
//*-------------------------------------------------------------------
//ADD     EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
RACDCERT ADD('YOUR.RACF.CERT.')        +
   SITE                                     +
   TRUST WITHLABEL(CERT_SSL)
```

5. Connect SITE Certificate to Keyring

```
//*-------------------------------------------------------------------
//* CONNECT SWISSSIGN CERTIFICATE TO KEYRING
//*-------------------------------------------------------------------
//CONNECT EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
RACDCERT ID(SVAZBSS) CONNECT(SITE LABEL('CERT_SSL')   +
RING(UKEYRING) USAGE(CERTAUTH) )
```

In the following example please change the keywords <IPADDRESS>, <PORT>, <FTP-USER> and <FTP-PASSWORD> with the provided information from SVA for transferring the files via FTPS. The recommendations regarding the keyword <SMFOUT.GZIP> are described in the chapter Naming convention recommendations.

```
//*  ============================================================
//*  SEND SMF GZIP TO ZWR (FTPS EXAMPLE)
//*  ============================================================
//ZWRFTP  EXEC PGM=FTP,PARM='-r TLS <IPADDRESS> <PORT>   (EXIT'
//SYSFTPD DD *
 TLSMECHANISM FTP
 SECURE_MECHANISM TLS
 KEYRING UKEYRING
 CLIENTERRCODES EXTENDED
 LOGCLIENTERR TRUE
 SECURE_FTP REQUIRED
 DEBUG SEC
//SYSPRINT DD SYSOUT=*
//INPUT    DD *
<FTP-USER> <FTP-PASSWORD>
bin
LOCSITE FWF
put <SMFOUT.GZIP>
quit
```

3.4.4 Via SFTP

If you decide to transfer the files via SFTP, please consider the following prerequisites:

1. Create or use existing (technical) z/OS user with OMVS segment to create an SSH-key-pair and for submitting the job.
   Note: If you use an existing user beware of a possible existing ssh-key-pair. Otherwise, it will be overwritten.
2. SFTP authentication is passwordless, therefore an SSH-Key-Pair is required. To create an SSH-key-pair for this user go to USS, switch to the previously created z/OS user, and generate a SSH-key-pair using the following commands via “TSO OMVS":

```
su - -s <technical_user_id>
/bin/ssh-keygen -t rsa -f ${HOME}/.ssh/id_rsa -N ""
chmod 700 .ssh
chmod 600 .ssh/id_rsa
chmod 644 .ssh/id_rsa.pub
```

3. The SSH-key-pair was generated under $HOME/.ssh/ directory. Switch the directory to $HOME/.ssh and list directory to verify.

```
cd $HOME/.ssh
id_rsa
id_rsa.pub
```

4. Please provide the content of the public ssh key file 'id_rsa.pub' to SVA. You can list the content with the following command:

```
cat $HOME/.ssh/id_rsa.pub
```

5. After SVA stores your public key in the zWR destination server, you need to accept the ssh fingerprint. To accept the fingerprint, go to USS with the ID that created the SSH-key-pair, use the following command and confirm the prompt with ‘yes’. SVA will provide the keywords <PORT>, <FTP-USER> and <IP-ADDRESS>.

```
sftp -P <PORT> <FTP-USER>@<IP-ADDRESS>
```

The transfer of the GZIP via sftp requires two steps. The first step copies the GZIP from MVS to USS (PATH needs to be checked).

```
//*  ============================================================
//*  SEND SMF GZIP TO ZWR (SFTP EXAMPLE)
//*  ============================================================
//*  ============================================================
//*  Step: Copy compressed Dataset to USS
//*  ============================================================
//CPTOUSS  EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//INPUT    DD DISP=SHR,DSN=*.COMPRESS.OUTPUT
//SYSTSIN  DD *,SYMBOLS=EXECSYS
  ALLOC F(OUTPUT) +
  PATH('/&SYSNAME./tmp/SMF.DUMP.GZIP') + <needs to be checked>
  PATHMODE (SIRWXU) PATHOPTS (OCREAT,ORDWR)
  OCOPY INDD(INPUT) OUTDD(OUTPUT) BINARY
/*
```

The second step creates dynamically a batchfile in the USS for the sftp client and then transfer the GZIP to zWR servers. Please change the keywords <USER>, <IPADDRESS>, <PORT>, <GZIP-NAME> and <FTP-PASSWORD> with the provided information from SVA for transferring the files via SFTP.

```
//* ============================================================.
//* SYMBOLS EXPORT
//* ============================================================
//E1 EXPORT SYMLIST=(USER)
//E2 EXPORT SYMLIST=(IPADDR)
//E3 EXPORT SYMLIST=(PORT)
//E4 EXPORT SYMLIST=(GZIP)
//* ============================================================
//* SYMBOLS
//* ============================================================
//SET1 SET USER=<USER>
//SET2 SET IPADDR=<IPADDRESS>
//SET3 SET PORT=<PORT>
//SET4 SET GZIP=<GZIP-NAME>
//*  ============================================================
//*  Step: Create USS sftp batch cmd file
//*  ============================================================
//USSFILE  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT2   DD PATH='/tmp/sftp_batch_file',
// PATHMODE=SIRWXU,
// PATHOPTS=(ORDWR,OCREAT,OTRUNC),
// PATHDISP=(KEEP,DELETE),
// FILEDATA=TEXT
//SYSUT1   DD *,SYMBOLS=EXECSYS
put &GZIP
bye
//*  ============================================================
//*  Step: send compressed dataset to SVA zWR server
//*  ============================================================
//SFTP     EXEC PGM=BPXBATCH
//STDERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//BPXPRINT DD SYSOUT=*
//STDPARM  DD *,SYMBOLS=JCLONLY
SH
/bin/sftp -P &PORT -b /tmp/sftp_batch_file
&USER.@&IPADDR
```

3.5 Inhouse transfer relay

If your internal security regulations or other concerns prohibit directly sending data from the mainframe to the internet, then a transfer server that is located (mostly in a DMZ) in your datacenter is a suitable solution. Please ask your internal security or network department if there is already a company wide solution.

To accomplish the set up with a transfer server in between of your mainframe and our zWR server, you must change the IP-address in the jobs to your transfer server. You will now send the GZIP files from your mainframe to your transfer server. Please verify that your internal firewall is not blocking this connection. If your security or network department have security specifications about the transfer protocol, please proceed according to these specifications otherwise you can use ftp, ftps or sftp. The last step is to set up the transfer server to send the GZIP files to our zWR server. In this step you need to use the provided IP-address and port from us.

3.6 Upload frequency

You can upload automate once a day or schedule the upload more frequently with the minimum being no more than hourly. Going under one hour upload frequency is technically possible but not allowed. If you have this requirement, then we recommend our product zGuard. With zGuard you can get real time monitoring and reporting and many other features. Contact the SVA support for more information.

3.7 Sending duplicate smf data

In general overlapping data is not a problem but it creates more overhead during processing, so the import can take significantly longer.