October 2017

IBM® z/VSE
Best Practices

Using SCRT (Java Version) with z/VSE
Version 1.0

Heiko Schnell (hschnell@de.ibm.com)
Jens Remus (jremus@de.ibm.com)
z/VSE Team (zvse@de.ibm.com)
IBM Germany Research & Development

Feedback Please!

Feel free to send your feedback on this document to the author of this document or ideas for other Best Practices documents to the z/VSE Team.

Target Audience

This document provides the Best Practices for using SCRT (Java Version) with z/VSE.

This document is targeted for the following:

- z/VSE Support Specialists
- z/VSE System Administrators
- z/VSE Customers

Table of Contents

Introduction

The purpose of this document is to describe best practices to use the Java Version of IBM Sub-Capacity Reporting Tool (SCRT) with accounting records from z/VSE.

Background and History

With z/VSE V4 IBM introduced sub-capacity pricing based on the four-hour rolling average utilization. To support this pricing model z/VSE can produce SCRT89 accounting records using the Capacity Measurement Tool (CMT). The CMT is shipped as part of z/VSE V4 and later. Until now z/VSE customers could produce the sub-capacity report by running the Classic version of SCRT on z/VSE.

On 11 October 2016 IBM announced that the Classic version of SCRT which runs on a z/VSE system will no longer be supported after October 2017. Customers previously running the Classic version of SCRT on their z/VSE must now use the new Java version of SCRT.

Possibilities to use CMT on z/VSE with the Java Version of SCRT

There are multiple possibilities to run the IJBCMTRP utility on z/VSE, transfer the SCRT89 records from z/VSE to a Java platform, and to use the Java version of SCRT to produce the sub-capacity report.
The Java version of SCRT expects the SCRT89 records as binary FTP block mode[1] data blocks.

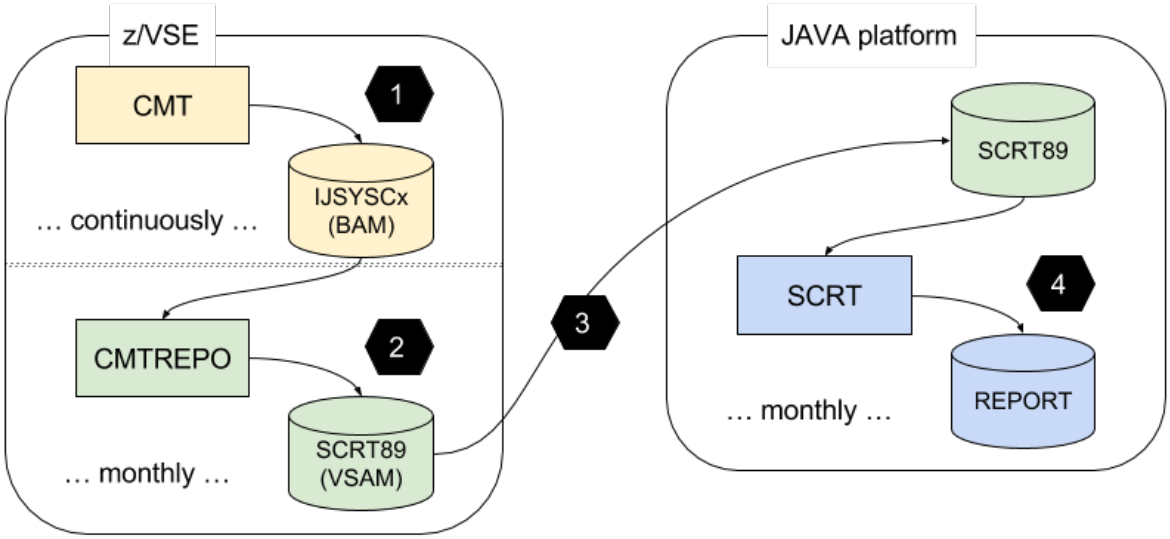

Figure 1: Overview of processing SCRT89 data from z/VSE on a Java platform

With all these possibilities, the basic steps shown in Figure 1 are used:

1. CMT continuously writes SCRT89 records into the data set IJSYSCC, IJSYSC1, and IJSYSC2.
   Then, once each month, the following steps must be done:
2. Run IJBCMTRP utility on z/VSE (see sample job CMTREPO in ICCF library 59) to extract the SCRT89 data for the reporting period from the data sets into a VSAM ESDS file.
3. Transfer the VSAM ESDS file, containing the SCRT89 records, from the z/VSE system to a Java platform where SCRT will run.
4. Run SCRT on the Java platform to process all the SCRT89 records and write the sub-capacity report file. Then this report file can be transferred, verified and submitted to IBM.

This document focuses on the following:

- Using FTP with BINARY and MODE B on the host side to transfer the SCRT89 records.
- Using REXX to pre-format the SCRT89 records as FTP block mode data blocks,
  and host file-transfer programs (referred to as IND$FILE) to transfer them.
- Using the VSE Script Connector to run IJBCMTRP utility on z/VSE
  triggered by a script running on the Java platform.
- Using the new SCRT Command Line Interface.
- Automation of monthly tasks (basic steps 2 to 4) using a batch file or a shell script.

[1] Each binary record must be prefixed with a FTP block header as described by RFC 959 File Transfer Protocol (FTP) – Section 3.4.2 – Block Mode on pages 21 and 22 → https://tools.ietf.org/html/rfc959#page-21.

Preparing z/VSE to Record and Extract the SCRT89 Records on z/VSE

When you prepare your z/VSE system to perform the basic steps 1 and 2 you can use the skeletons provides in ICCF library 59. These are also mentioned in the z/VSE Planning Manual as follows:

Member Name	Library	Use
SKCMT	59	Catalogs the procedure CMTLAB.PROC which contains DLBLs for the CMT data files, and catalogs the procedure CMTSTART.PROC to start the CMT measurement.
SKCMTINI	59	Initializes the CMT data files. This should ONLY be done before the first start of the CMT measurement!
SKCMTREP	59	Extracts the SCRT89 records from the CMT data files written by the CMT, and writes them into a VSAM ESDS dataset for further processing using the SCRT.

For details please refer to chapter “Sub-capacity reporting for z/VSE systems” in the SCRT Users Guide “Using the Sub-Capacity Reporting Tool”. There you can find the following information:

- Introduction to sub-capacity pricing for z/VSE systems
- Requirements for sub-capacity pricing for z/VSE systems
- Identities reported for z/VSE systems
- Sub-capacity reporting for multiple z/VSE versions
- Overview of the CMT
- Configuring the CMT files
- Steps for preparing z/VSE to use CMT

Preparing the Java Platform to Run the SCRT

To prepare the Java platform for basic step 4 (run SCRT) you need to download and install the latest version of the SCRT from:
http://www.ibm.com/systems/z/swprice/subcap/scrt/

IBM provides the latest Java version of SCRT for z/OS Java, Linux and Microsoft Windows.
Even if it is not explicitly mentioned and supported, the version provided for Linux may also run on other UNIX like systems, like OS X or macOS.

After downloading the Linux version, you need to install SCRT. To start the GUI version of the installer you can use the command `"./installSCRT.bin"` from the download directory in a terminal window. If you are installing SCRT to a system that does not have a graphical environment, you must install in console mode by using the command `"./installSCRT.bin -i console"` from the download directory. The installer will guide you through the rest of the installation.

To start the GUI version of the SRCT program in a terminal window you can use the command `"java -classpath "*" com.ibm.scrt.visual.Main_GUI"` from the installation directory. This will avoid error `"java.lang.NoClassDefFoundError: com/ibm/json/java/JSONObject"`, which you might see if `jsj.jar` is missing in the class path.

Using FTP to Transfer SCRT89 Records from z/VSE to a Java Platform

The most obvious method for basic step 3 is using FTP to transfer the SCRT89 records from z/VSE to a Java platform. To use FTP, you need a TCP/IP product like "TCP/IP for z/VSE" or "IPv6/VSE" on your z/VSE system.

In principle, there are 2 different ways to use FTP:

1. Using a FTP server on z/VSE and a FTP client on the Java platform.
2. Using a FTP client on z/VSE and a FTP server on the Java platform.

In both cases, you must transfer the SCRT89 records as binary FTP block mode data blocks. Therefore, you need to specify **`BINARY`** and on the z/VSE side also **`MODE B`** (block mode) to ensure the data is transferred as binary FTP block mode data blocks, that are stored in a binary stream on the Java platform.

For additional information, you may also refer to chapter "Using SCRT on Windows and Linux" in the SCRT Users Guide "Using the Sub-Capacity Reporting Tool".

In the following samples, **`<VSAM-ESDS-file>`** is used as a place holder for the DLBL name of the VSAM ESDS file, containing the SCRT89 records, on the z/VSE system. You must also define this file depending on the batch job and product used to run the FTP server or client on z/VSE.
As a destination file **`SCRT89.bin`** is used on the Java platform. If you wish, you can add path information to the destination file or rename it. But the extension of the file name on the Java platform should be either ".bin", ".scrt89" or ".vse", to allow SCRT to process this correct.

When you are using "TCP/IP for z/VSE", depending on your configuration, you may need to disable UNIX and EXTTYPES to allow a binary transfer. A description of the FTP client and server commands can be found in the chapter "FTP" of the "TCP/IP for VSE User Guide" under headings "VSE FTP Client Command Descriptions" and "VSE FTP Server Commands". For details about the SITE command refer to section "SITE Commands". The "TCP/IP for VSE User Guide" is available on the following sites:
http://publibz.boulder.ibm.com/cgi-bin/bookmgr/XKS/iest2010
http://www.csi-international.com/products/zVSE/TCP-IP/TCP-IP-doc.htm

Sample commands that can be used with FTP client on the Java platform and the FTP daemon from "TCP/IP for z/VSE" on the z/VSE side:

```
QUOTE SITE UNIX OFF
QUOTE SITE EXTTYPES OFF
BINARY
QUOTE MODE B
QUOTE SITE RECFM V
QUOTE SITE LRECL 32756
GET <VSAM-ESDS-file> SCRT89.bin
```

Sample commands that can be used with FTPBATCH client from "TCP/IP for z/VSE" on the z/VSE side and the FTP server on the Java platform:

```
LQUOTE SITE UNIX OFF
LQUOTE SITE EXTTYPES OFF
BINARY
LQUOTE MODE B
LQUOTE SITE RECFM V
LQUOTE SITE LRECL 32756
PUT <VSAM-ESDS-file> SCRT89.bin
```

When you are using "IPv6/VSE", you can find the description of the FTP client and server commands in the "Chapter 1 – Commands" and the "Chapter 2 – FTP COMMANDS" of the "IPv6/VSE Users Guide". For details about the SITE command refer to the "Chapter 5 – Using the FTP Server" under the heading "FTP Server SITE Commands". The "IPv6/VSE Users Guide" is available on the following site: http://publibz.boulder.ibm.com/cgi-bin/bookmgr/XKS/iesipv10

Sample commands that can be used with FTP client on the Java platform and on the z/VSE side the BSTTFTPC (FTP Server) from "IPv6/VSE":

```
BINARY
QUOTE MODE B
QUOTE SITE INPUT VSAM <VSAM-ESDS-file>
GET SCRT89.bin
```

Sample commands that can be used with BSTTFTPC batch FTP client from "IPv6/VSE" on the z/VSE side and the FTP server on the Java platform:

```
TYPE I
LMODE B
INPUT VSAM <VSAM-ESDS-file>
STOR SCRT89.bin
```

Note: The **`LMODE B`** command will put the BSTTFTPC FTP client into block mode without telling the remote host FTP server.

Using REXX and Host File-Transfer Program (Referred to as IND$FILE)

If you don't have a TCP/IP product installed or there is no FTP client or server on your Java platform, your basic step 3 will have the two parts, formatting and transfer the data. First you need to format the SCRT89 records as FTP block mode data blocks (as defined in RCF 959), so that it can be sent to the Java platform using another method. For example, you can use the host file transfer (also referred as IND$FILE) from the terminal emulator running on your Java platform to transfer the formatted data.

To format the SCRT89 records as FTP block mode data blocks, you can use the REXX sample: ftp://public.dhe.ibm.com/eserver/zseries/zos/vse/download/xmps/scrt89_to_htf_rexx.jcl

This example demonstrates how to utilize REXX/VSE to copy records from one VSAM ESDS dataset to another while prepending FTP block mode headers.

The provided job will first catalog the REXX procedure FTPBLOCK.PROC into the sub library PRD1.BASE. Within the header of the procedure you can find a description how this can be used:

```
/**
 * (c) Copyright IBM Corporation 2017
 *
 * This file is used to demonstrate how to utilize IBM Corporation's
 * REXX/VSE to copy records from one VSAM ESDS dataset to another
 * while prepending FTP block mode headers.
 *
 * You have a royalty-free right to use, modify, reproduce and
 * distribute this demonstration file (including any modified
 * version), provided that you agree that IBM Corporation has
 * no warranty, implied or otherwise, or liability for this
 * demonstration file or any modified version.
 *
 * The primary use case is to transfer SCRT89 records via the
 * Host Transfer File (HTF) from the host to a workstation for
 * further processing using the SCRT.
 *
 * Usage:
 *   // EXEC REXX=FTPBLOCK,PARM='<input-label> <output-label>'
 *   // EXEC REXX=FTPBLOCK,PARM='<input-label> <output-label> ( <options>'
 *
 * Parameters:
 *   <input-label> specifies the name of the file label of the input
 *     VSAM ESDS dataset.
 *   <output-label> specifies the name of the file label of the output
 *     VSAM ESDS dataset.
 *
 * Options:
 *   HTF: indicates that records should not exceed the maximum Host
 *     Transfer File (HTF) record length. The REXX procedure prints a
 *     warning message for every overlong record and returns with
 *     return code 4 if the maximum HTF record length was exceeded.
 *   SPLIT: indicates that records exceeding the maximum record length
 *     of the output dataset, or the maximum Host Transfer File record
 *     length when used in conjunction with option HTF, are split into
 *     multiple records.
```

```
 *     This option is useful when the output datatset is processed as a
 *     stream of bytes (instead of a sequence of records), as the split
 *     records are then joined again, for example when transferring the
 *     dataset off the host (e.g. to a workstation file).
 *
 * Return Codes:
 *   0 successful completion.
 *   4 option HTF was given and at least one record exceeded the maximum
 *     HTF record length.
 *   8 an error occurred. Refer to the error message for details.
 *
 * Example:
 *   // DLBL SCRT89,'SCRT89.RECORDS',0,VSAM,CAT=VSESPUC
 *   // DLBL SCRT89T,'SCRT89.RECORDS.FTPBLOCK.TEMP',0,VSAM,CAT=VSESPUC
 *   // EXEC REXX=FTPBLOCK,PARM='SCRT89 SCRT89T ( HTF SPLIT'
 *
 * References:
 *   - RFC 959, section 3.4.2 Block Mode,
 *     https://www.ietf.org/rfc/rfc959.txt
 */
```

After the REXX procedure has been stored, the provided job will allocate a temporary SCRT89 dataset:

```
* ALLOCATE TEMPORARY SCRT89 DATASET ...
// EXEC IDCAMS,SIZE=AUTO
   DELETE (SCRT89.RECORDS.FTPBLOCK.TEMP) CLUSTER PURGE -
      CATALOG(VSESP.USER.CATALOG)
   IF  LASTCC = 8 THEN -
   DO
      SET LASTCC = 0
      SET  MAXCC = 0
   END
   DEFINE CLUSTER ( -
      NAME (SCRT89.RECORDS.FTPBLOCK.TEMP) -
      CYLINDERS (2 2) -
      SHAREOPTIONS (1) -
      RECORDSIZE (80 32756) -
      VOLUMES (SYSWK1) -
      REUSE -
      NONINDEXED -
      FREESPACE (15 7) -
      NOCOMPRESSED -
      TO (99366) -
   ) -
   DATA ( -
      NAME (SCRT89.RECORDS.FTPBLOCK.TEMP.@D@) -
      CONTROLINTERVALSIZE (32768) -
   ) -
   CATALOG (VSESP.USER.CATALOG)
   IF LASTCC NE 0 THEN CANCEL JOB
/*
```

Then it will call the just cataloged REXX procedure FTPBLOCK.PROC:

```
* COPY AND FTP-BLOCK SCRT89 RECORDS TO TEMPORARY DATASET ...
// DLBL SCRT89,'SCRT89.RECORDS',0,VSAM,CAT=VSESPUC
// DLBL SCRT89T,'SCRT89.RECORDS.FTPBLOCK.TEMP',0,VSAM,CAT=VSESPUC
// LIBDEF PROC,SEARCH=(PRD1.BASE)
// EXEC REXX=FTPBLOCK,PARM='SCRT89 SCRT89T ( HTF SPLIT'
/*
```

After that the job will copy the FTP data block formatted SCRT89 records to the host transfer file (HTF):

```
* COPY FTP-BLOCKED SCRT89 RECORDS TO HOST TRANSFER FILE (HTF) ...
* See: IBM z/VSE System Utilities, chapter Batch Access to the
*      z/VSE Host Transfer File
* ************************************************************ *
* CLOSE THE HOST TRANSFER FILE IN CICS:                        *
* CEMT SET FILE(INWFILE) CLOSE                                 *
* ************************************************************ *
// PAUSE - CLOSE THE HOST TRANSFER FILE IN CICS (SEE ABOVE)
// DLBL SCRT89,'SCRT89.RECORDS.FTPBLOCK.TEMP',0,VSAM,CAT=VSESPUC
// EXEC INWMUTIL,SIZE=AUTO
LOAD,FILENAME=SCRT89,FILETYPE=BIN,USERID=SYSA
/*
* ************************************************************ *
* OPEN THE HOST TRANSFER FILE IN CICS:                         *
* CEMT SET FILE(INWFILE) OPEN                                  *
* ************************************************************ *
// PAUSE - OPEN THE HOST TRANSFER FILE IN CICS (SEE ABOVE)
```

Last, but not least, it will delete the temporary SCRT89 data set and show on the console how you can transfer the formatted SCRT89 records stored in the host transfer file (HTF) to the Java platform:

```
* DELETE TEMPORARY SCRT89 DATASET ...
// EXEC IDCAMS,SIZE=AUTO
   DELETE (SCRT89.RECORDS.FTPBLOCK.TEMP) CLUSTER PURGE -
      CATALOG(VSESP.USER.CATALOG)
   IF LASTCC NE 0 THEN CANCEL JOB
/*
* ************************************************************ *
* TRANSFER THE SCRT89 RECORDS STORED IN THE HOST TRANSFER FILE AS    *
* 'SCRT89 BIN' TO THE WORKSTATION FILE 'SCRT89.BIN' VIA THE TERMINAL *
* SESSION 'A' USING IBM PERSONAL COMMUNICATIONS AS FOLLOWS:          *
*                                                                    *
* 1. OPEN THE HOST TRANSFER FILE IN CICS IF NOT ALREADY DONE:        *
*    CEMT SET FILE(INWFILE) OPEN                                     *
* 2. IN A IUI TERMINAL SESSION EITHER HIT PF6/PF9 OR NAVIGATE TO     *
*    PC FILE TRANSFER (FAST-PATH 386)                                *
* 3. ON A WORKSTATION COMMAND LINE ISSUE:                            *
*    RECEIVE SCRT89.BIN A:SCRT89 BIN (FILE=HTF BINARY NOCRLF KEEP    *
* ************************************************************ *
```

Using the VSE Script Connector to Execute the IJBCMTRP Utility on z/VSE

For automation of basic step 2, the VSE Script Connector can be used to execute the IJBCMTRP utility on z/VSE controlled by a script running on the Java platform.

To allow the VSE Script Connector to connect to your z/VSE System, you must start the VSE Connector Server on your z/VSE System by using the STARTVCS job. For details please refer to chapter "Installing and Operating the Java-Based Connector" in the "IBM z/VSE e-business Connectors, User's Guide" under the heading "Starting the VSE Connector Server".

To prepare the Java platform you must download and install the VSE Connector Client and the VSE Script Server. Both can be found here: http://www.ibm.com/systems/z/os/zvse/downloads/

For details on installation of the VSE Connector Client please refer to chapter "Installing and Operating the Java-Based Connector" in the "IBM z/VSE e-business Connectors, User's Guide".

For details on installation of the VSE Script Server, please refer to chapter "Installing the VSE Script Connector" in the "IBM z/VSE e-business Connectors, User's Guide".

You may also refer to the technical article "How to setup SSL with the VSE Script Connector" that can be found on http://www-03.ibm.com/systems/z/os/zvse/documentation/security.html#howto

For additional information, how it can be used you can refer to the chapter "Using the VSE Script Connector for Non-Java Access" in the "IBM z/VSE e-business Connectors, User's Guide".
This chapter contains these main topics:

- How the VSE Script Connector Can Be Used
- Overview of the Protocol Used Between Client and Server
- Writing VSE Scripts Using the VSE Script Language
- Sample Files You Can Use for Writing VSE Script Clients
- Example of Writing a VSE Script Client (and its VSE Script)
- Obtaining Data from the Middle-Tier Using VSE Script Clients

After the installation, you can also find the HTML-based documentation of the VSE Script Server in the /doc subdirectory where you have installed the VSE Script Server as following files:

- index.html VSEScript Language and Functions Reference
  - Language reference
  - Functions reference
  - Interfaces to invoke a script
- server.html VSE Script Server
  - Overview
  - Script Repository
  - Connection Pool
  - Starting the VSE Script Server
  - Administrating the VSE Script Server
  - Client Programs
  - The VSEScriptServer.properties file
  - The Connection.properties file
  - Examples

Before you can use the VSE Script Server you must define the connection to the z/VSE host in the properties file **`Connections.properties`**.

The file **`Connections.properties`** is a text file that you can edit using any text editor. Comment lines begin with a # character in the first column. You might specify something like:

```
#Connection.properties
connection.1.name=VSE01
connection.1.ip=9.111.22.3
connection.1.port=2893
connection.1.userid=SYSA
connection.1.password=SYSA
connection.1.ldapsignon=false
connection.timeout=100
```

Each connection to a VSE backend is defined by a set of 5 to 10 properties, each starting with **`connection.n.`** where n stands for a number starting from 1. Each connection has properties for its logical name, the IP address of the VSE system, the port number used to connect to VSE, the userid and password. Passwords can be specified using the property **`connection.n.password.`**
During startup of the VSE Script Server, the password is encrypted and stored using the property **`connection.n.encpassword.`** The optional property **`connection.n.ldapsignon`** controls if LDAP sign-on is to be used or not. In addition, you can define the global setting **`connection.timeout = seconds`**. The time in seconds before an unused connection is closed.

After starting the VSE Script Server, the file **`Connections.properties`** might look like:

```
#Encrypted passwords
#Wed Sep 20 15:09:33 CEST 2017
connection.1.encpassword=15sdFBYB6e9U
connection.1.ip=9.111.22.3
connection.1.ldapsignon=false
connection.1.name=VSE01
connection.1.port=2893
connection.1.userid=SYSA
connection.timeout=100
```

For additional information, you may also refer to:

- Chapter “Installing the VSE Script Connector”
  in the “IBM z/VSE e-business Connectors, User’s Guide”
- VSE Script Server in the HTML-based documentation under /doc/server.html.

To start the VSE Script Server, you may call the **`runserver.bat`**, **`runserver.cmd`**, or **`runserver.sh`** from the from the installation directory. On some systems like macOS you might need to specify the classpath in the execution command. Then use following command from the installation directory:

```
java -classpath "*:/.../IBM/VSE Connector Client/*" com.ibm.vse.script.VSEScriptServer
```

When the VSE Script Server is running, you can use a VSE Script Client to execute a VSE Script.

Even if there are a lot of possibilities this document will focus only on the easy way to run the script locally by using the **com.ibm.vse.script.RunScript** Java class, or the **runscript.bat, runscript.cmd,** or **runscript.sh.**

The script that you want to execute should be stored in the **/scripts** subdirectory.

When you have a script **executeCMTREPO.src** in the **/scripts** subdirectory, you can use **runscript.bat** or **./runscript.sh** with the parameter **executeCMTREPO.src** to start the script, that uses the running VSE Script Server to execute the JOB on a z/VSE System specified in the **Connections.properties** file.

To do this you need a script based on the sample **scripts/samples/executePowerJob.src** to execute the job CMTREPO using the IJBCMTRP utility on z/VSE.

For this, you can save the following sample as **scripts/executeCMTREPO.src:**

```
// ***************************************************************
//  Sample Script to execute POWER JOB CMTREPO.job
// ***************************************************************

string jin, jout;
int rc;
// read the local file into variable jin
readFile("./scripts/CMTREPO.job", &jin, &rc);
// execute the job in jin and store the joboutput in jout
executePowerJob("VSE01",&jin,&jout,&rc);

if(rc!=0) do;
 println("Error during function executePowerJob");
 exit(2);
endif;

int y,z;
arraysize(&jout, &y);
z=0;
// print the job output line by line
while(z<y) do;
 println(jout[z]);
 z=z+1;
endwhile;
```

You also need to store your customized **CMTREPO** job in the **scripts** directory.

Therefore, you may adjust and save the following job based on the skeleton SKCMTREP from ICCF library 59 as **scripts/CMTREPO.job**:

```
* $$ JOB JNM=CMTREPO,CLASS=0,DISP=D,NTFY=YES
* $$ LST CLASS=A,DISP=D
// JOB CMTREPO   EXTRACT RECORDS FROM CMT DATASETS
* ********************************************************************
* *     SAMPLE JOB TO EXTRACT SCRT89 RECORDS FROM THE CMT DATASETS *
* ********************************************************************
* *  THE FOLLOWING VARIABLES ARE USED AND HAVE TO BE CHANGED       *
* *                                                                *
* *    --V001-- DISK TYPE OF THE DISK WHERE THE CMT FILES RESIDE   *
* *             - ECKD OR FBA                                      *
* *                                                                *
* *    --V002-- PHYSICAL UNIT ADDRESS (CUU) OF THE DISK            *
* *             WHERE CMT FILES RESIDE                             *
* *                                                                *
* *             START LOCATION OF CMT FILE (BLOCKS OR TRACKS)      *
* *                                                                *
* *    --V302-- LOGICAL UNIT NAME (SYSNNN) FOR CMT DATA FILE1      *
* *    --V303-- LOGICAL UNIT NAME (SYSNNN) FOR CMT DATA FILE2      *
* *                                                                *
* *    --V105-- VSAM CATALOG NAME CONTAINING THE CMT REPORT FILE   *
* *                                                                *
// SETPARM DTYPE=--V001--
/. STEP31
// IF DTYPE = ECKD THEN
// GOTO LCKD
/. LFBA
// DLBL IJCMTC1,'CMT.SCRT89.DATA.FILE1',99/366,SD
// EXTENT --V302--
// ASSGN --V302--,--V002--
// DLBL IJCMTC2,'CMT.SCRT89.DATA.FILE2',99/366,SD
// EXTENT --V303--
// ASSGN --V303--,--V002--
// GOTO STEP32
/. LCKD
* FOR ECKD
// DLBL IJCMTC1,'CMT.SCRT89.DATA.FILE1',99/366,SD
// EXTENT --V302--
// ASSGN --V302--,--V002--
// DLBL IJCMTC2,'CMT.SCRT89.DATA.FILE2',99/366,SD
// EXTENT --V303--
// ASSGN --V303--,--V002--
/. STEP32
// DLBL CMTREPO,'CMT.SCRT89.DATA.REPORT',,VSAM,CAT=--V105--
// EXEC IJBCMTRP
MONTH    = LAST
OUTPUTDS = DD:CMTREPO
/&
* $$ EOJ
```

Using the SCRT Command Line Interface

Since version 24.12 SCRT supports a command line interface (CLI) on Windows and Linux platforms. It is intended as an alternative to the graphical user interface (GUI) on those platforms to support an automated workflow.

It is possible to call the SCRT CLI from a script for automation of basic step 4.

IBM provides the latest Java version of SCRT for z/OS Java, Linux and Microsoft Windows.
Even if it is not explicitly mentioned and supported, the version provided for Linux may also run on other UNIX like systems, like OS X or macOS.

If you are installing SCRT to a Linux system that does not have a graphical environment, you must install in console mode by invoking the installation program with the **`"-i console"`** option.
To install in console mode you can use the following command:

```
./installSCRT.bin -i console
```

The console mode installer will guide you through the rest of the installation.

For details on the SCRT CLI please refer to section “Using the SCRT command line interface” in chapter “Using SCRT on Windows and Linux systems” in the SCRT Users Guide “Using the Sub-Capacity Reporting Tool”.

SCRT provides a sample **`scrt-cli.bat`** batch script (for Windows systems) and a **`scrt-cli.sh`** shell script (for Linux systems) that you can customize and use to invoke the CLI.
You will find a detailed description, how the samples **`scrt-cli.bat`** or **`scrt-cli.sh`** can be used, under the heading “Using the SCRT CLI sample scripts”.

Alternatively you can start the SCRT command line interface by calling the scrtc.exe or scrtc.command direct with all the necessary options and the path to the input SCRT89 dataset.

Under the heading “Specifying command line options” you can find description of the options and how it can be specified.

One or more SMF or SCRT89 input files must be specified at the end of the SCRT command line.

- Names of SMF input data files (from z/OS) must have a **`.smf`** extension.
- Names of SCRT89 input data files must have a **`.bin`**, **`.scrt89`**, or **`.vse`** extension.
- You can specify relative or fully-qualified path names.
- If a path contains blank characters, enclose it with double quotation marks.
- If the last option you specify accepts multiple arguments, like **`--exclude`** or **`--lpar-comment`**, you must precede your SMF or SCRT89 paths by a double hyphen surrounded by white space **`--`** so that those paths are not interpreted as additional arguments for the preceding option.

The invocation, on a Linux or a UNIX like system, might then look like following sample. The command should be entered on a single line. The line breaks in the example are for formatting purposes only.

```
./scrtc.command --customer-name "John Customer" --customer-number "8970000001"
  --contact-name "John Smith" --contact-email "smithj@example.com"
  --contact-phone "800-555-1234" --no89 "/path/to/no89data.txt"
  --lpar-comment "CPC=2817-00000,LPAR=SYS1,COMMENT=\"This is my comment\""
  --output report.csv "/path/to/SCRT89.bin"
```

Automation of Monthly Tasks with a Batch File or Shell Script

For an automation of the monthly tasks (basic steps 2 to 4) you can use a batch file or shell script running on the Java platform, that does the following:

1. Ensure that the prerequisites for the next steps are resolved.
2. Run a VSE Script to execute the CMTREPO job using the IJBCMTRP utility on z/VSE.
3. Call FTP client to transfer the SCRT89 records from z/VSE to the Java platform.
4. Call SCRT by using the command line interface to generate the SCRT report.

Step 1 – Ensure that the Prerequisites for the Next Steps are Resolved

As a prerequisite for step 2 the VSE Connector Server must run on the z/VSE System and on the Java platform the VSE Script Server must be started. This cannot be automated easily, but the Servers can also run permanently and be used for automation of other things.

Step 2 – Run a VSE Script to Execute the CMTREPO Job

To execute the VSE Script, you first need to change the actual directory to the installation directory of the VSE Script Connector. Normally this is the directory **`"IBM/VSE Script Server"`** under your home directory. Please refer to section “Using the VSE Script Connector to Execute the IJBCMTRP Utility on z/VSE” on page 10 for details on running the VSE Script.

Step 3 – Call FTP Client to Transfer the SCRT89 Records from z/VSE to the Java Platform

To transfer the SCRT89 records from z/VSE to the Java platform, you need a running FTP Server on z/VSE and a FTP client on the Java platform, that can be used from the script or batch file.

When you have a command line FTP client you can save the FTP commands, that you normally enter in the command window, in a file (e.g. the file **`ftpCommands.txt`** in the **`scripts`** sub-directory). Then you can redirect the standard input for the execution of the FTP client from the file with the FTP commands.

You may also read the password into a variable and pass it to the FTP client, instead of specifying it in **`ftpCommands.txt.`** But then you need an FTP client where you can pass the password as a parameter. You may use the command **`read -sp "Please enter the VSE password" PASSWORD`** and pass **`$PASSWORD`** as parameter to your FTP client on a Linux or UNIX like system.

See also section “Using FTP to Transfer SCRT89 Records from z/VSE to a Java Platform” on page 5 for detailed information which FTP commands you need dependent on the used FTP server on z/VSE.

Step 4 – Call SCRT by Using the Command Line Interface to Generate the SCRT Report

To call the SCRT CLI you can either use a single command or an individual customized script or batch file based on the samples provided with SCRT.
For details refer to section “Using the SCRT command line interface” in chapter “Using SCRT on Windows and Linux systems” in the SCRT Users Guide “Using the Sub-Capacity Reporting Tool” and the section “Using the SCRT Command Line Interface” on page 14 in this document.

When using Linux or a UNIX like system, the script could look like the following example:

```
#!/bin/sh
# ********************************************************************
# Sample script for automation of CMT / SCRT monthly tasks
# ********************************************************************
# Prerequisites:
# - The VSE Connector Server must run on the z/VSE System (STARTVCS)
# - The VSE Script Server must be started (runserver)
# ********************************************************************
echo "**************************"
echo "* Run SCRT monthly tasks *"
echo "**************************"
# Step 1 - Solve Prerequisites
echo "Please ensure that the VSE Connector Server is active (STARTVCS)."
echo "Please ensure that VSE Script Server is running (runserver)."
read -p "Press any key to continue" dummy
# Step 2 - Execute CMTREPO using IJBCMTRP Utility on z/VSE
cd
cd "IBM/VSE Script Server"
./runscript.sh executeCMTREPO.src
# Step 3 - Transfer the SCRT89 Records from z/VSE
cd scripts
ftp 9.111.22.3 < ftpCommands.txt
# Step 4 - Call SCRT CLI to generate the SCRT Report
cd "/Applications/IBM Sub-Capacity Reporting Tool"
./scrtc.command --customer-name "John Customer" --customer-number "8970000001" \
  --contact-name "John Smith" --contact-email "smithj@example.com" \
  --contact-phone "800-555-1234" --no89 "/path/to/no89data.txt" \
  --lpar-comment "CPC=2817-00000,LPAR=SYS1,COMMENT=\"This is my comment\"" \
  --output report.csv "/path/to/SCRT89.bin"
```

References

References about IBM z Systems Software Pricing and the Sub-Capacity Reporting Tool (SCRT)

Sub-Capacity for z/VSE
http://www.ibm.com/systems/z/swprice/subcap/zvse.html

Sub-Capacity Reporting Tool (SCRT) Overview and Download
http://www.ibm.com/systems/z/swprice/subcap/scrt/
http://www.ibm.com/systems/z/swprice/subcap/scrt/download.html

Sub-Capacity Reporting Tool (SCRT) Instructions – SCRT Guidance
http://www.ibm.com/systems/z/swprice/subcap/scrt/instruct.html
Users Guide "Using the Sub-Capacity Reporting Tool"
http://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=ZSL03022USEN
https://public.dhe.ibm.com/common/ssi/ecm/zs/en/zsl03022usen/zsl03022.pdf
Chapters: "Sub-capacity reporting for z/VSE systems", "Using SCRT on Windows and Linux systems"

References about Transferring SCRT89 Records as FTP Block Mode Data Blocks

RFC 959 File Transfer Protocol (FTP)
https://tools.ietf.org/html/rfc959
Section 3.4.2 – Block Mode on pages 21 and 22
https://tools.ietf.org/html/rfc959#page-21

When no TCP/IP or FTP product is available on z/VSE
REXX sample to format the SCRT89 records as FTP block mode data blocks
http://www.ibm.com/systems/z/os/zvse/downloads/samples.html#rexx
ftp://public.dhe.ibm.com/eserver/zseries/zos/vse/download/xmps/scrt89_to_htf_rexx.jcl

When you are using "TCP/IP for z/VSE"
TCP/IP for VSE User Guide is available on the following sites:
http://publibz.boulder.ibm.com/cgi-bin/bookmgr/XKS/iest2010
http://www.csi-international.com/products/zVSE/TCP-IP/TCP-IP-doc.htm
Sections: "VSE FTP Client Command Descriptions", "VSE FTP Server Commands", "SITE Commands"

When you are using "IPv6/VSE"
IPv6/VSE Users Guide is available on the following site:
http://publibz.boulder.ibm.com/cgi-bin/bookmgr/XKS/iesipv10
Chapters: "Commands", "FTP Commands", "Using the FTP Server" Section "FTP Server SITE Commands"

References about VSE Connector and VSE Script Connector

Download of "VSE Connector Client" and "VSE Script Server"
http://www.ibm.com/systems/z/os/zvse/downloads/

IBM z/VSE e-business Connectors, User's Guide
http://publibz.boulder.ibm.com/cgi-bin/bookmgr/download/IESCUE72.pdf
Chapters: "Installing the VSE Script Connector", "Using the VSE Script Connector for Non-Java Access"
"Installing and Operating the Java-Based Connector", "Installing and Operating the Java-Based Connector"

How to setup SSL with the VSE Script Connector
http://www-03.ibm.com/systems/z/os/zvse/documentation/security.html#howto

HTML-based documentation of the VSE Script Server in the /doc subdirectory:

index.html	VSEScript Language and Functions Reference
server.html	VSE Script Server

Disclaimer

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This information could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) at any time without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (*e.g.*, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein. The customer is responsible for the implementation of these techniques in its environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. Unless otherwise noted, IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights.

Trademarks

IBM, the IBM logo, ibm.com, z/OS and z/VSE are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

IPv6/VSE is a registered trademark of Barnard Software, Inc.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
OS X and macOS and are trademarks of Apple Inc., registered in the U.S. and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.