**Abstract:**

This tool collects more granular CPU figures from RMF III than the Post Processer from RMF I – RMF III provides CPU utilization figures for much shorter period of time than RMF I -- in a manner of automatically accumulating these more granular CPU utilization figures to a sequential file.

**Author:**

Andrew Jan

**Requirements:**

This tool is to extract values from the RMFIII’s VSAM files. So the RMFIII should be properly configured and started to write data to the pre-allocated VSAM files.

Please read the “**Setting up the Monitor III gather session RMFGAT**” section in the chapter: “Setting up RMF” from the manual: “**z/OS Resource Measurement Facility User’s Guide**” about how to define VSAM files for RMFIII gathering.

Please also read the “**Starting a Monitor III gatherer session**”, “**Modifying RMF section options**” and “**Stopping RMF session**” sections in the chapter: “**Starting and controlling Monitor session**” from the same book: “**z/OS Resource Measurement Facility User’s Guide**” about how to start, stop and modify RMFIII data collections.

Please have a look at the attached file: “***Sample\_RMFIII\_Parameters\_ERBRMF04.txt***” for how to specify the RMFIII’s parameters. In the meantime, refer to “**Description of Monitor III data gatherer options**” section in chapter “**Short-term data gathering with Monitor III**” from the same book: “**z/OS Resource Measurement Facility User’s Guide**” for the meanings of these parameters.

The book: “**z/OS Resource Measurement Facility User’s Guide**” can be found from the **z/OS Internet Library**:

<https://www-304.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary?OpenDocument> “

**Explanation of the Purpose:**

The Short-term data gatherer: RMF III offers more granular CPU utilization figure than RMF I.

We may configure RMF III to collect CPU and other resources’ utilization figures for as short as every few seconds (typically 30 seconds), whereas RMF I normally keeps the utilization data for 30 minutes.

For some z/OS systems in real world, the fluctuating business demands may cause the utilization of the CPU and other resources to keep changing significantly that we have to keep track of the resource utilizations for much smaller periods than what the Post Processer can produce for us from the RMF I records.

While writing system performance data to memory and then VSAM files, RMF III provides ISPF panels to let us watch current z/OS system performance data or older data kept on these VSAM files. Unfortunately, we may only access as far as one or two days ago for these system performance related figures because more current or newer data keeps coming to have to overwrite those old figures in these VSAM data sets.

To keep the utilization figures – among all resources, the CPU utilization figures are the most important -- for good so we may analyze them later without having to manually invoke the RMF III’s ISPF panels all the time, I therefore developed this tool to extract CPU utilization figures from the RMF III’s VSAM data sets.

**How to install this tool:**

1. For running RMF3CPC, create an output sequential data set with a record length long enough to contain all your LPAR’s figures. Our output records are fixed as 600 bytes in length to contain 10 LPARs’ information (see the attached file: “***Sample\_Result\_of\_Running\_RMF3CPC.zip***”).   
     
   Adjust the record length of your output sequential data set for collecting more LPARs or optionally update the logic of the source code: “***RMF3CPC.asm***” around the label: “**LOOP\_LPAR EQU \***” in “***RMF3CPC.asm***” to omit certain LPARs.
2. For running RMF3CPU, create an output sequential data set with a record length as 50. The sample result is as the attached file: “***Sampe\_Result\_of\_Running\_RMF3CPU.zip***”.
3. Compile the programs: “**RMF3CPC**” or “**RMF3CPU**” and “**XIEC070I**” from the attached source code files: *“****RMF3CPC.asm*** *”, “****RMF3CPU.asm****” and “****XIEC070I.asm****”* to create LMDs (Loaded Modules) into one of your LMD PDS (Partitioned Data Sets) that is on the Link List concatenation.  
     
   Issue ‘**F LLA,REFRESH**’ to refresh the PDS directories in LLA, once these LMD are successfully created.
4. Issue first ‘**D MPF**’ to find out the current MPF list version.  
     
   Update the current MPFLST that is effective. For example, if it is ‘**00**’ from the ‘**D MPF**’ command, then update **MPFLST00** to add the following statement:  
   “IEC070I,AUTO(YES),SUP(NO),USEREXIT(XIEC070I) /\*RMF3 VSAM \*/”   
     
   Please refer to the attached file: “***Sample\_MPFLST00.txt***” for the MPFLST statement.   
     
   Issue ‘**T MPF=00**’ to let the change take effect. (**00** or the **other value** that you have found out)
5. Create a new Started Procedure, named ‘**RMF3CPC**’ or another name of your choice in your PROC library like ‘**SYS1.PROCLIB**’ or **your own PROCLIB**.   
     
   Please refer to the attached file: “***Sample\_RMF3CPC\_Started\_PROC.jcl***” for a sample of the Started Procedure.  
     
   Sample RACF settings are like the following:  
     
   RDEF STARTED **RMF3CPC.\*** STDATA(USER(IBMSTCER) GROUP(SYS1)   
   TRUSTED(NO))  
     
   RALT STARTED **RMF3CPC.\*** OWNER(OSROWNER)  
     
   RDEF OPERCMDS MVS.START.STC.**RMF3CPC.\*** UACC(N) OWNER(IBMUSER)  
     
   PE MVS.START.STC.**RMF3CPC.\*** CLA(OPERCMDS) ID(**RMFGAT**) AC(UPDATE)   
   (***RMFGAT*** *is the PROC’s or address space’s or daemon’s name for RMF III)*  
     
   PE SLR.\*\* ID(IBMSTCER) AC(U) (*this is because our output file is* ***SLR****…. something*)  
     
   SETR REFRESH RACLIST(STARTED)  
     
   SETR REFRESH RACLIST(OPERCMDS)  
     
   Update those ID or keywords to reflect your system’s own definitions and naming conventions.
6. The result is as the attached file “***Sample\_Result\_of\_Running\_RMF3CPC.zip***” for “**RMF3CPC**” or   
   “***Sampe\_Result\_of\_Running\_RMF3CPU.zip***” for “**RMF3CPU**”.

**Maintenance:**

When an RMF III’s VSAM data set is full and RMF III is switching to the next VSAM for newer data writing, an MVS console message: “**IEC070I**” will pop up showing which VSAM data set is full. Here we take advantage of the z/OS system’s MPFLST facility -- see attached file: “***Sample\_MPFLST00.txt***” -- to intercept “**IEC070I**” message and trigger the attached program: “***XIEC070I.asm***” where a Started Procedure named “**RMF3CPC**” -- see attached “***Sample\_RMF3CPC\_Started\_PROC.jcl***” -- is invoked by issuing “**S RMF3CPC**” inside the program.

The started procedure: “**RMF3CPC**” is, in turn, invoking the attached program: “***RMF3CPC.asm***”

Please read the chapters named: “**Using Monitor III VSAM data set support**” and “**Monitor III data reporter tables**” in the manual: “**z/OS Resource Measurement Facility Programmer’s Guide**” for more information.

The book: “**z/OS Resource Measurement Facility Programmer’s Guide**” can be found from the **z/OS Internet Library**:

<https://www-304.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary?OpenDocument> “

To compile “***XIEC070I.asm***”, you should include the attached macro library: “***#IF\_#ELSE\_MACROs.zip***”. Create a 80-column PDS (Partitioned Data Set) of your own and upload these members to that PDS as macros for “***XIEC070I.asm***” to compile against.

The attached program: “***RMF3CPC.asm***” is for the collection of **all LPAR**’s CPU information of **the entire CPU** while the other attached program: “***RMF3CPU.asm***” is collecting only the CPU figures for **this LPAR** where the program runs on.

Choose either “***RMF3CPC.asm***” or “***RMF3CPU.asm***” that is fit for purpose.

However, if “***RMF3CPC.asm***” is used to collect RMF III data all LPARs, please make sure that both “**Performance data control**” and “**Cross partition authority**” features have been checked or ticked for this LPAR that the RMF III is intended to run on.

The two features can be checked on the **CPU’s Support Element** from “**CPC Operational Customization**” 🡪 “**Change LPAR Security**”.