



## **z/OS Management Facility V2.4 Resource Requirements**

**Tian Na  
([tiantn@cn.ibm.com](mailto:tiantn@cn.ibm.com))  
2020.01.08**

## Table of Contents

<b>z/OSMF Introduction .....</b>	<b>3</b>
<b>z/OSMF Hardware Requirements .....</b>	<b>4</b>
<b>z/OSMF Software Requirements .....</b>	<b>5</b>
<b>z/OSMF startup tuning .....</b>	<b>6</b>
<b>New functions of z/OSMF V2R4.....</b>	<b>9</b>
<b>Setup an appropriate performance goal on WLM.....</b>	<b>10</b>
<b>z/OSMF regression tests and considerations .....</b>	<b>12</b>
<b>Summary .....</b>	<b>15</b>
<b>Document Change History .....</b>	<b>16</b>

### **z/OSMF Introduction**

IBM z/OS Management Facility (z/OSMF) provides system management functions in a task-oriented, web browser-based user interface with integrated user assistance, so that you can more easily manage the day-to-day operations and administration of your mainframe z/OS systems. By streamlining some traditional tasks and automating others, z/OSMF can help to simplify some areas of z/OS system management.

z/OSMF provides a framework for managing various aspects of a z/OS system through a web browser interface.

z/OSMF provides you with a single point of control for:

- Viewing, defining, and updating policies that affect system behavior
- Monitoring the performance of the systems in your enterprise
- Managing software that runs on z/OS
- Performing problem data management tasks
- Consolidating your z/OS management tools.

z/OSMF allows you to communicate with the z/OS system through a web browser, so you can access and manage your z/OS system from anywhere. Multiple users can log into z/OSMF using different computers, different browsers, or multiple instances of the same browser.

In this document, we will introduce the resources requirement for z/OSMF and give you some guidance on tuning z/OSMF startup. Also there are some new functions and regression tests in this document. The results of the regression test are equivalent with the previous version (z/OSMF V2.3).

## z/OSMF Hardware Requirements

z/OSMF requires a certain amount of CPU resource and memory for satisfactory operation. Some z/OSMF scenarios consume more CPU time than others. The target system for z/OSMF should have:

- **At least 50 PCI(Processor Capacity Index)** for completion the z/OSMF startup.  
Due to some scenarios need more CPU while the CPU resource is not enough, so there maybe have some TCCL(thread context classloader) timeout. Don't worry about that, you can Increase the timeout threshold by specifying a larger value in an override file. You can find details about changing this time out value in z/OSMF startup tuning part in this document.

And configured a zIIP will helpful to the z/OSMF performance.

- **Four gigabyte (4G)** of central storage.  
For z/OSMF, the minimum memory requirement is 4G. But be aware that z/OS V2R4 IPL on IBM z14 (z14) server, the minimum hardware memory requirement is 8 gigabyte (8G) of central storage.

If you configured z/OSMF with auto-start it will start automatically during system IPL. In that time, the resource and dependency the z/OSMF needed may not ready completely. So the auto-start time might be longer than manual start.

**Note:** PCI is generally equivalent to millions of instructions per second (MIPS).

## **z/OSMF Software Requirements**

Java must be installed and operational on your z/OS system, at the required minimum level:

### **IBM® 64-bit SDK for z/OS®, Java™ Technology Edition, V8 SR4 FP10 (5655-DGH)**

By default, the SDK resides in the directory /usr/lpp/java/J8.0\_64 on your system. If you installed it in another location, be sure to include the JAVA\_HOME statement in your IZUPRMxx parmlib member.

z/OSMF uses the Liberty Profile that is supplied with z/OS. The WebSphere Liberty profile must be mounted on your z/OS system. The default mount point is: /usr/lpp/liberty\_zos. To determine whether WebSphere® Liberty profile is mounted, check for the existence of the mount point directory on your z/OS system.

If WebSphere® Liberty profile is mounted at a non-default location, you must specify the location in the IZUSVR1 started procedure on the keyword WLPDIR=.

Almost all function provided by z/OSMF using web browser. Now the following web browsers are supported by z/OSMF, and are recommended for best results:

**Microsoft Internet Explorer Version 11**

**Microsoft Edge (Windows 10)**

**Mozilla Firefox ESR Version 60 or later**

## z/OSMF startup tuning

z/OSMF startup time varies considerably depending upon many factors. These include CPU resource, z/OSMF Server configuration, number of starting applications, type of applications, etc.

There are some tuning tips to improve the z/OSMF startup performance:

- **Configure share class cache and AOT cache size**
- **Configure a zIIP**
- **Enable plugins that you needed instead of enable all plugins**
- **Increase the timeout value in the resource constrained environment**
- **Disable KC index build**

The first four are introduced in z/OSMF v2r3 resource requirement. So there is only a brief introduction. For details please find in <https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD106415>

The last two tips are new. They will be introduced in detail in this paper.

- **Configure share class cache and AOT cache size**

Shared classes use optimizations to maintain performance under most circumstances. The JVM might automatically store a small amount of Ahead Of Time (AOT) compiled native code in the cache when it is populated with classes. The AOT code enables any subsequent JVMs attaching to the cache to start faster. AOT data is generated for methods that are likely to be most effective.

So increasing the shared class cache and AOT cache properly can improve the startup performance.

You can use `-Xscmxsize` to specified the size of shared class cache and use `-Xscmaxaotssize` to specified the size of AOT cache. There are also some useful commands for managing the cache:

- To check the information of a cache:  
`java -Xshareclasses:cacheDir=/global/zosmf/data/logs/.classCache,name=cache_name,printStats`
- To destroy all caches under the cache directory:  
`java -Xshareclasses:cacheDir=/global/zosmf/data/logs/.classCache,destroyall`
- To list all caches under the cache directory:  
`java -Xshareclasses:cacheDir=/global/zosmf/data/logs/.classCache,listAllCaches`

The z/OSMF used default cache directory is `/global/zosmf/data/logs/.classCache`. If you used the customized directory change this to yours.

- **Configure a zIIP**

IBM® z Integrated Information Processor (zIIP) is a purpose-built processor designed to operate asynchronously with the general processors in the mainframe to help improve utilization of computing capacity and control costs. It is designed for select data and transaction processing workloads and for select network encryption workloads. zIIPs allow customers to purchase additional processing power without affecting the total million service units (MSU) rating or machine model designation. IBM does not impose IBM software charges on zIIP capacity, but charges apply when additional general purpose CP capacity is used.

## z/OS Management Facility V2.4 Resource Requirements

IBM zIIPs can take over some specific workloads, such as Java and XML, DB2 for z/OS, OMEGAMON, RMF and so on. During z/OSMF startup, there are a lot of Java programs to be process. So if you already configured zIIP in your machine, don't forget to use it. It will give more help on z/OSMF startup. In our lab environment, we configured with 3% of one GCP (about 50 PCI) and an additional dedicated zIIP. We saw it will save 80% ~ 90% time on z/OSMF startup.

To learn more about IBM zIIP, please find following link:

<https://www.ibm.com/us-en/marketplace/z-integrated-information-processor-ziip>

- **Enable plugins that you needed instead of enable all plugins**

As we all known, z/OSMF provide rich functionality by adding plugins. While the core functions on z/OSMF is also very strong and powerful. It covered almost all commonly used functions. So if your environment has limited resource you can consider enable plugins that you needed instead of enable all plugins.

For starting z/OSMF core, you can comment all the plugins in IZUPRMxx. And to enable the plugins you wanted just uncomment them in the IZUPRMxx.

- **Increase the timeout value in the resource constrained environment**

On start-up, the z/OSMF server (IZUSVR1) does not fully initialize in specific limited resources. The following message or a similar message is written to the server job log:

```
ERROR CWWKZ0002E: An exception occurred while starting the application
IzuManagementFacilitySoftwareDeployment. The exception message was:
com.ibm.ws.container.service.state.StateChangeException: java.lang.IllegalStateException:
Unable to acquire TCCL store lock
```

The cause is the thread context classloader (TCCL) store lock timeout threshold was exceeded. By default, this time limit is 5 seconds. In a resource constrained environment, this time limit can be exceeded.

Increase the timeout threshold by specifying a larger value in an override file. For example, adding `JVM_OPTIONS=-Dcom.ibm.ws.classloading.tcclLockWaitTimeMillis=300000` to `/global/zosmf/configuration/local_override.cfg` and restart z/OSMF server.

You can experiment with this value by reducing it to find the minimum possible timeout for your environment. The z/OSMF server is not negatively affected if the value is set higher than necessary.

- **Disable KC index build**

When z/OSMF starts, the embedded knowledge center within z/OSMF will build its index which consumes a lot of CPU time. On small system which could have less MIPS or is simulation system such as zPDT, the startup time of z/OSMF will take more time and CPU which customers are not able to tolerate. After investigating, one of facts stalling the startup of z/OSMF is help index rebuilding.

With APAR PH06678, a new option "KCINDEX" is provided in IZUSVR which provide a ability for customer to control if the index rebuilding will happen. If "KCINDEX=N" is provided when starting IZUSVR, the embedded knowledge center will not build its index. As index is not rebuilt, startup time has 12% improvement. CPU usage has 24% improvement. I/O has 16% improvement. The result is based-on our performance test environment. The performance result may vary slightly according to different environment. Given the benchmark test result, it's recommended to restart z/OSMF server with index

## z/OS Management Facility V2.4 Resource Requirements

rebuilding disabled building after the very first time start of z/OSMF. The default KCINDEX is Y means the embedded knowledge center will build its index during z/OSMF startup.



## New functions of z/OSMF V2R4

There are three main new functions on z/OSMF V2R4: z/OSMF Lite configuration, new Security Configuration Assistant task and new z/OSMF Diagnostic Assistant task.

- **z/OSMF Lite configuration**

z/OSMF Lite intends to provide most flexibility to customers from configuration perspective. It allows user to start with a minimum z/OSMF configuration and only do necessary additional setup based on the services they need.

It takes approximately 2 - 3 hours to set up a z/OSMF Lite configuration. But due to the internal design, the startup time on z/OSMF Lite are equal to the startup time on z/OSMF core.

- **New Security Configuration Assistant task**

You can use the Security Configuration Assistant task to verify that security is configured properly for the current z/OSMF host system and its users. It has two key features:

- Displays the security setup details for z/OSMF in a graphical user interface (GUI).
- Performs security checks on your setup and provides the results in an easy-to-read table format.

Like other tasks in desktop, the first validate the services will spend more time than the second. In the lab environment, we try to validate the cloud service, the first time is about 3-4s and second time will decrease to less 1s. For validate all services, the first time is about 17s, and second time is about 13s. Due to the environment variations, the results may have volatility.

- **New z/OSMF Diagnostic Assistant task**

You can use the z/OSMF Diagnostic Assistant task to collect diagnostic data about z/OSMF and download it as a compressed file package(.zip).

The download speed is various depend on the file size. We tested in lab environment, it takes about more than 2 minutes to download the file about 320M. So if you want to reduce the download time you can limit data collection by selecting **runtime and server side logs** option. This will only collect the data from <USER\_DIR>/data/logs.

## Setup an appropriate performance goal on WLM

In z/OS system, we usually using Workload Management (WLM) define performance goals for workloads in business terms. WLM will monitor the system and adapt how much resource, such as CPU or storage, should be given to meet those goals.

z/OSMF workload can also benefit from WLM by setting up an appropriate performance goal. The following will introduce the general method to setup z/OSMF performance goal.

**First, confirm the WLM feature is enabled.** Usually z/OSMF already defined this feature in the server.xml file which specified zosWlm-1.0.

**Second, create the classification rules for z/OSMF.** The CB classification rule is defined for WAS related transaction. z/OSMF also use this classification rule. The CN (Collection Name) and TC (Transaction Classes) should be created. The "Qualifier Name" of "Transaction Class" should be align with what is defined in z/OSMF parmlib option "WLM\_CLASSES". By default, they are "IZUGHTTP" for z/OSMF transactions and "IZUGWORK" for long running task of Software Management plugin. It is necessary to specify the server directory name in qualifier name. The server directory name is zosmfServer which we abbreviated as zosmf\*. Note that it must match exactly so mixed case is required. Here is an example:

\* Subsystem Type CB - Rules for z/OSMF (Transactions)

```
Classification:
  Default service class is IZUGWORK
  There is no default report class.
  Qualifier Qualifier Service Report
  #   type      name      Class   Class
  - - - - -
  1 CN          zosmf*    IZUGWORK
  2 . TC        . IZUGHTTP IZUGHTTP RSMPESTRN
  2 . TC        . IZUGWORK IZUGWORK
```

**Third, create the required service classes.** We create two service classes for z/OSMF, IZUGHTTP and IZUGWOR. The IZUGHTTP service class is for foreground z/OSMF transactions (except asynchronous beans). The IZUGWORK service class is for long running asynchronous beans and for unexpected z/OSMF work, not otherwise classified. The following is an example:

\* Service Class **IZUGHTTP** - WebSphere transactions

Base goal:

CPU Critical = YES I/O Priority Group = NORMAL

```
# Duration Imp Goal description
- - - - -
1          2    80% complete within 00:00:02.000
```

\* Service Class **IZUGWORK** - Low Priority

Base goal:

CPU Critical = NO I/O Priority Group = NORMAL

```
# Duration Imp Goal description
- - - - -
1          Discretionary
```

If the workload performance can't satisfy your expectation you can tuning the performance goal by changing the importance and the goal.

**The last, install and active the WLM policy.** You can use both WLM panel and z/OSMF web UI to finish this step.

Be aware, if you tuning WLM goal, assure that you have a sufficient test.

## z/OSMF regression tests and considerations

Performance tests data in the IBM laboratory was collected for a single user or a small number of users by repeating a given scenario multiple times within a fixed test interval. Many scenarios can be repeated easily by clicking a refresh button, although others require multiple mouse clicks to re-drive the desired action.

The tests were executed on a 3906-701 processor (z14) LPAR with a single dedicated GCP and 8 GB central storage. The following is the software versions for our performance test:

- z/OS V2.4 and z/OSMF V2.4
- IBM 64-bit SDK for z/OS Java V8 SR5

The regression workload was composed of a broad mixture of transactions using the following z/OSMF applications:

- z/OSMF File API
- z/OSMF Job API
- z/OSMF Console API
- z/OSMF Incident Log
- z/OSMF MSS SSO enable

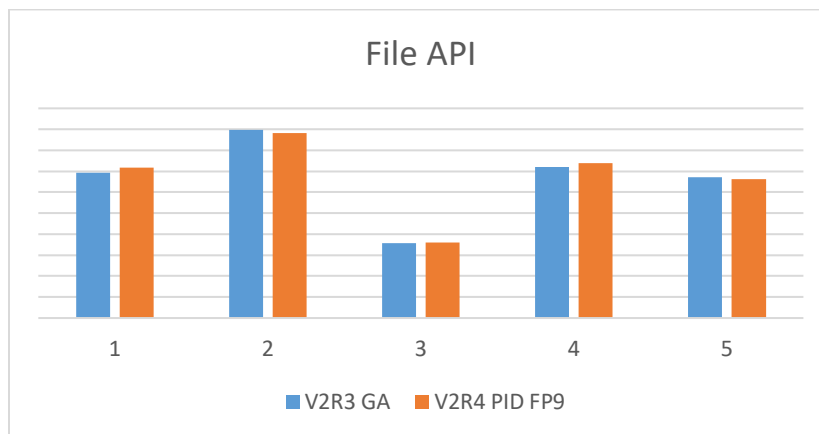
The test method we used is: First send requests incessantly within 20 minutes as warm up. And then collect performance data in 5 minutes. The comparisons between z/OSMF V2R4 and z/OSMF V2R3 are based on the ITR results.

The X axis is stand for each scenario and the Y axis is stand for the ITR. ITR is Internal Throughput Rate which is an indicator for performance in IBM.

### z/OSMF File API

For the z/OSMF File API test, there are five scenarios:

- List dataset without volume or X-IBM-Attribute, List 10 datasets names
- Browse a dataset of 80 bytes data
- List dataset with volume and X-IBM-Attribute:base request header, List 1 dataset name
- List dataset without volume or X-IBM-Attribute, List 1 datasets name
- read zFS file and then write file content back



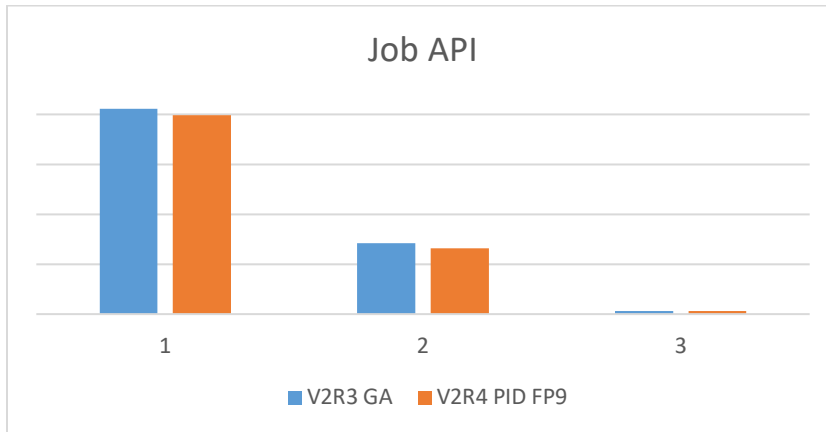
## z/OS Management Facility V2.4 Resource Requirements

From the chart, we can see the V2R4 results are almost equal to V2R3.

### z/OSMF Job API

For the z/OSMF Job API, there are three scenarios:

- Get Job status without step completion code
- Get Job status with step completion code
- Submit job

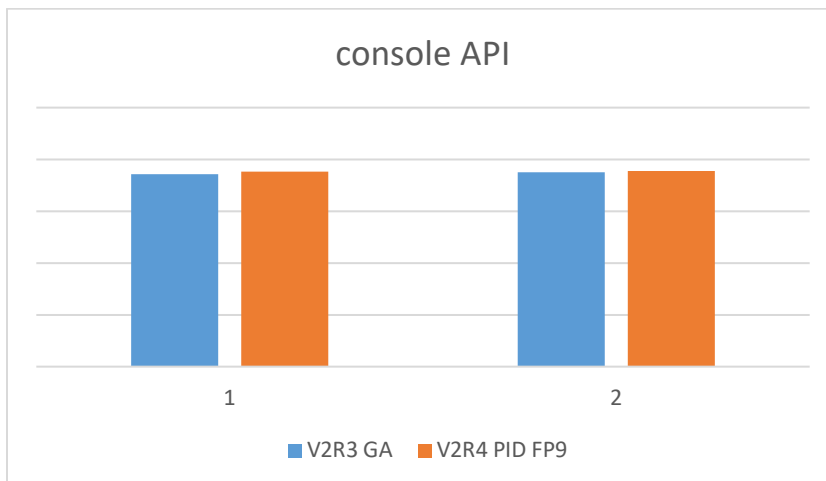


From the above chart we can see the V2R4 job API performance are almost equivalent to the V2R3.

### z/OSMF Console API

For the z/OSMF Console API, there are two scenarios we tested:

- Single command "D T"
- Single command "D T" and unsol-key detection

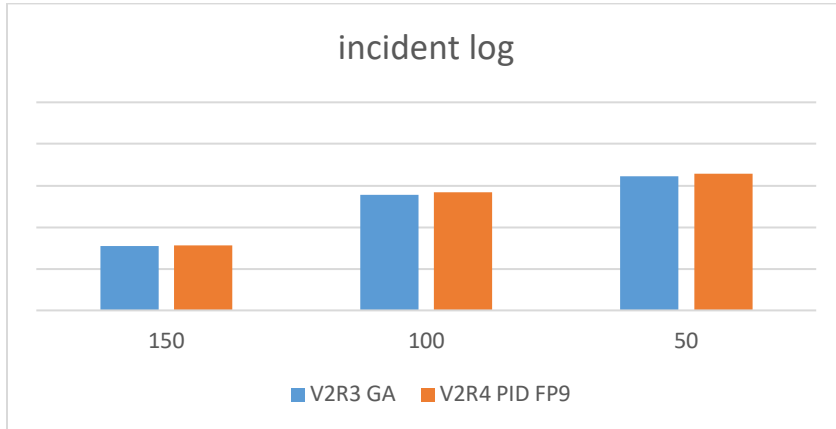


From the above chart we can see the equivalent performance between z/OSMF V2R3 and V2R4.

### z/OSMF Incident Log

For the z/OSMF Incident Log, there are three scenarios:

- 50 incident logs
- 100 incident logs
- 150 incident logs

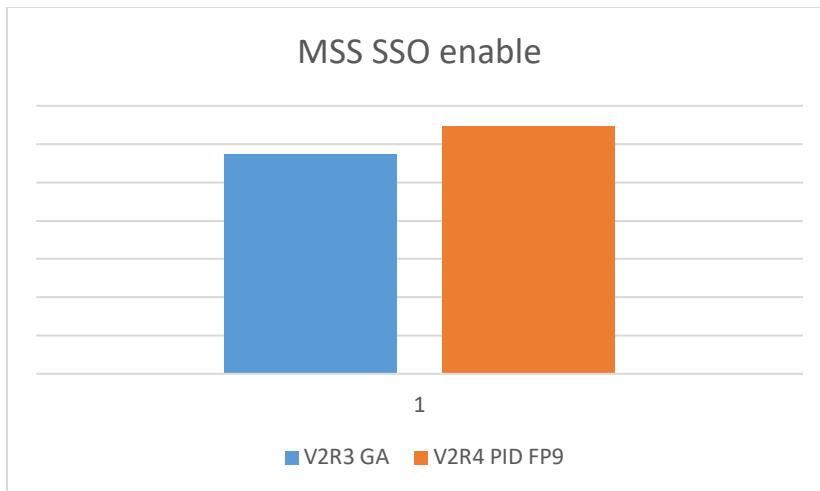


From the above chart we can see the z/OSMF V2R4 performance is almost equivalent to the z/OSMF V2R3.

There is a good practices that don't filter a very large range history if not necessary. If the history range is too large, will need longer time to show the all incidents.

### z/OSMF MSS SSO enable

For z/OSMF MSS SSO enable, the scenario is getting resource from remote system.



From the above chart we can see the z/OSMF V2R4 performance is better than z/OSMF V2R3.

### Summary

z/OSMF provides system management functions in a task-oriented, web browser-based user interface with integrated user assistance, so that you can more easily manage the day-to-day operations and administration of your mainframe z/OS systems.

For certain applications on z/OSMF will cost more CPU resources. So there are some hardware requirements:

- **At least 50 PCI** with longer time out value.
- **4GB** central storage

z/OSMF require the following Java level:

**IBM® 64-bit SDK for z/OS®, Java™ Technology Edition, V8 SR4 FP10 (5655-DGH)**

z/OSMF uses the Liberty Profile that is supplied with z/OS. The WebSphere Liberty profile must be mounted on your z/OS system.

Now the z/OSMF supported the following web browsers:

- **Microsoft Internet Explorer Version 11**
- **Microsoft Edge (Windows 10)**
- **Mozilla Firefox ESR Version 60 or later**

There are some methods to ensure the z/OSMF start successfully and speed up the startup process:

- **Configure share class cache and AOT cache size**
- **Configure a zIIP**
- **Enable plugins that you needed instead of enable all plugins**
- **Increase the timeout value in the resource constrained environment**
- **Disable KC index build**

There are three main new functions on z/OSMF V2R4: z/OSMF Lite configuration, new Security Configuration Assistant task and new z/OSMF Diagnostic Assistant task. z/OSMF Lite configuration make setting up z/OSMF simply; Security Configuration Assistant task help you validate security easily; Security Configuration Assistant task help you collect diagnostic data by just click the button.

There is an introduction to setup an appropriate performance goal on WLM. You can tuning your z/OSMF performance goal by reference this part.

There are also some regression tests did in lab. The results of the regression test are almost equivalent with the previous version (z/OSMF V2.3).

## Document Change History

December 4, 2019	Initial version
December 9, 2019	Update z/OSMF startup tuning part
December 26, 2019	Add Setup an appropriate performance goal on WLM part and other updates
December 30, 2019	Some updates on words
January 8, 2020	Finished