



WebSphere Liberty Profile on z/OS

Configuration/Management Best Practices

With a focus on Liberty servers for CICS, MQ, z/OSMF and z/OS Connect

Mitch Johnson

mitchj@us.ibm.com

Washington Systems Center

mitchj@us.ibm.com

<https://ibm.biz/BdahJr>



Notes and Disclaimers

- Additional information included in this presentation was distilled from experience implementing security using RACF with z/OS products like CICS, IMS, Db2, MQ, etc. as well as Java runtimes environments like WebSphere Application Server and WebSphere Application Server Liberty (commonly called Liberty).
- There will be additional information on slides that will be designated as Tech/Tips. These contain information that at perhaps at least interesting and hopefully, useful to the reader.
- A product icon will appear on a slide where the information is specific to a particular product. The icon will be  for z/OS, or  for Java, or  for Liberty, or  for CICS, or  for MQ, or  for IMS, or  for Db2 or  for z/OS Connect. Don't hesitate to ask questions as to why an icon does or does not appear on certain slides.
- The examples, tips, etc. present in this material are based on firsthand experiences.

Liberty Basics



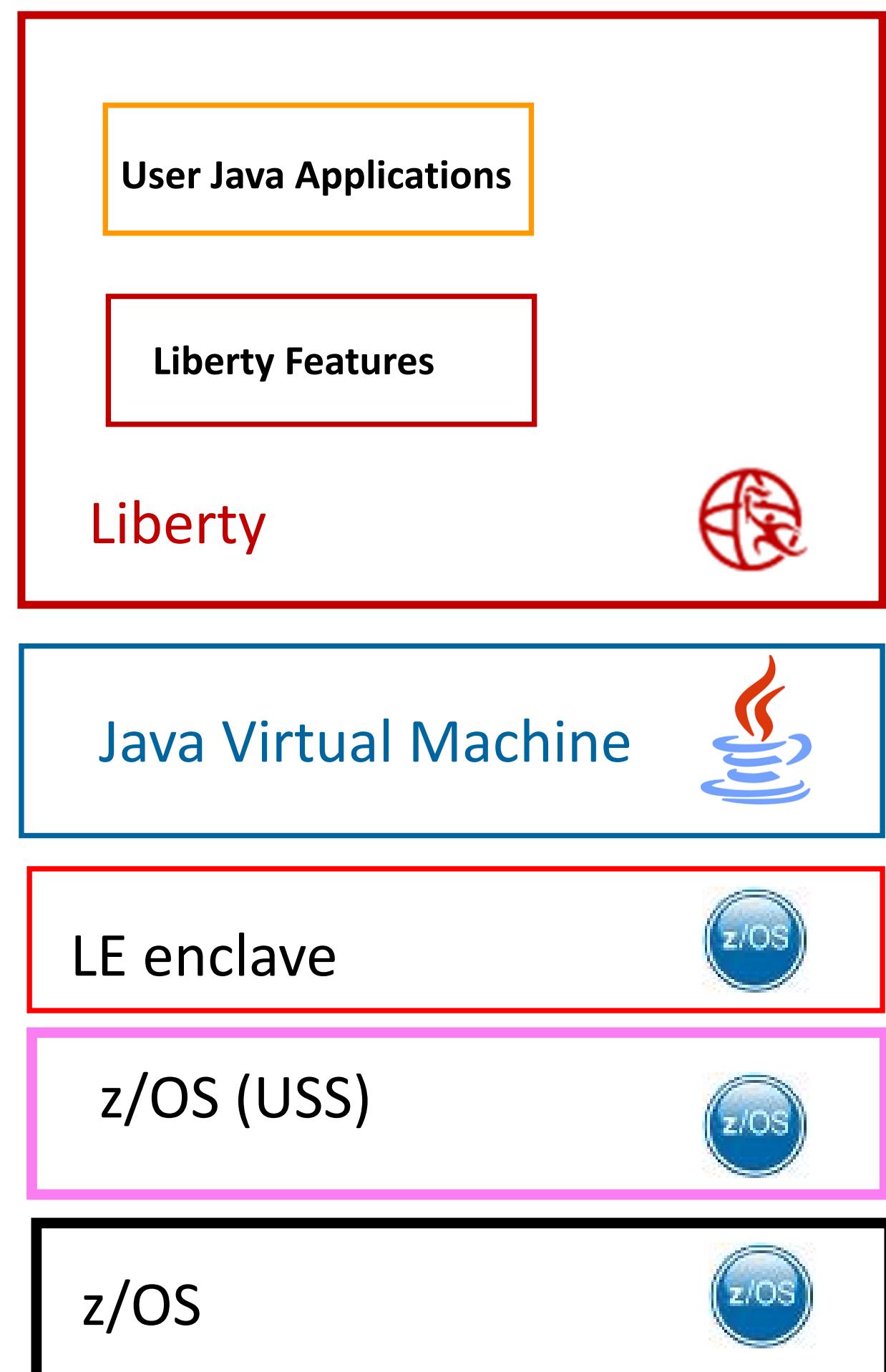
Liberty basics: It helps to think of a Liberty server on z/OS as a stack or layers of software

- At the foundation, Liberty servers run on z/OS to exploit z/OS quality of services
- Liberty servers are OMVS processes that are either directly started invoking a script or binary executable(e.g. An OMVS command or JCL) or spawn by parent process (e.g., CICS or z/OS Connect) .
- A Liberty server process runs in a Language Environment (LE) enclave configured for a Unix System Services environment tailored for running Java applications.
- The Liberty server provides a Java runtime environment enabled for providing access to z/OS services like SAF, WLM, RRS, SMF, etc. to user Java applications.
- A Liberty server also provides an execution environment for multiple concurrent Java application threads and manages connections and security for accessing z/OS resources.

Knowing and understanding the different layers and their relationships is important regarding:

- Understanding which layer a configuration options, e.g., environment variables, Java directives, etc.
- Monitoring and understanding the health of the server
- Performing problem determination and performance tuning

z/OS Connect and zOSMF should be considered a Java applications





These layers of software are shown in this example of a Liberty JCL procedure

```
//ZCEESRVR PROC PARMS='serverName'  
/*  
// SET INSTDIR='/usr/lpp/liberty_zos/21.0.0.9'  
/*  
//ZCON      EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=8G,  
// PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv &PARMS'  
//STDOUT    DD   SYSOUT=*  
//STDERR    DD   SYSOUT=*  
//STDIN     DD   DUMMY  
//MSGLOG    DD   SYSOUT=*  
//STDENV    DD   *  
_BPX_SHAREAS=YES  
_CEE_RUNOPTS=HEAPPOOLS (ON) ,HEAPPOOLS64 (ON)  
JAVA_HOME=/usr/lpp/java/J17.0_64  
WLP_USER_DIR=/var/wlp  
IBM_JAVA_OPTIONS=-Dcom.ibm.ws.zos.core.angelName=zCEEAngel -Xmx512m  
OPENJ9_JAVA_OPTIONS=-Xoptionsfile=/var/zcee/properties/myServer.property
```

OMVS
LE
JAVA
LIBERTY

- Environment variables provide the initial configuration information used for starting the server.
 - For a Liberty server started with JCL, environment variables are provided by STDENV DD input (as in this example).
 - For a Liberty server in CICS, environment variables are provided in JVM profile files located in the CICS region's JVMPROFILEDIR system initialization table (SIT) parameter.
 - For a z/OSMF Liberty server, environment variables are generated from a PARMLIB member and stored in a configuration file.

Tech-Tip: STDENV input can reside in an OMVS file or a MVS data set

Use the STDENV DD statement to scale servers and share configuration properties horizontally

```
//ZCON      EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=8G,
//  PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv &PARMS'
//STDERR    DD SYSOUT=*,FREE=CLOSE,SPIN=(UNALLOC,1M)
//STDOUT    DD SYSOUT=*
//STDIN     DD DUMMY
//STDENV    DD PATH='/var/liberty/properties/&PARMS..property',
//                      PATHOPTS=ORDONLY
//
// or
//STDENV    DD DISP=SHR,DSN=JOHNSON.LIBERTY.STDENV(COMMON)
//          DD DISP=SHR,DSN=JOHNSON.LIBERTY.STDENV(LIBSRVR)
```

Either one OMVS property file or multiple PDS members.

The last occurrence environment variable encountered determines the value of the environment variable.

Member COMMON

```
_BPX_SHAREAS=YES
_CEE_RUNOPTS=HEAPPOOLS(ON),HEAPPOOLS64(ON)
JAVA_HOME=/usr/lpp/java/J8.0_64           (environment variable is overridden by member LIBSRVR)
ZCON_ENV_DEBUG=TRUE
WLP_USER_DIR=/var/wlp                      (environment variable is overridden by member LIBSRVR)
```

Which value used for a Java option or property depends on which environment variable is used to specify the option or property.

Member LIBSRVR

```
OPENJ9_JAVA_OPTIONS=-Dcom.ibm.ws.zos.core.angelName=ZCEEANGL
IBM_JAVA_OPTIONS=-Xoptionsfile=/var/zcee/properties/javaHCD.property -Xmx512m -verbose:sizes
JAVA_HOME=/u/johnson/java/J17.0_64
WLP_USER_DIR=/var/wlp/ats
```

Using //STDENV DD * is discouraged because of the 80-character record limit.



Liberty basic : The environment variable **WLP_USER_DIR** is key

- The **WLP_USER_DIR** environment variable provides the starting directory location where:
 - A server's configuration file can be located
 - A default directory location for writing logs and other files
 - A default directory where a server can locate application artifacts
 - A directory where state information is saved[#]
- The same value for **WLP_USER_DIR** must be used for starting a server as was used when the server was created.
- There can be multiple “**WLP_USER_DIR**” directories on an LPAR
- Each server (**serverName**) will have a unique subdirectory in the directory location specified by **WLP_USER_DIR**.

- **WLP_USER_DIR** can be shared by multiple servers
 - \${WLP_USER_DIR}
 - /servers
 - /**serverName1**
 - /**serverName2**
 - /**serverName3**
 - |-shared/
 - |--apps/
 - |--config/
 - |--resources/

- The location of the **serverName** directory is based on the concatenation of the value of the **WLP_USER_DIR** environment variable with the constant *servers* in non-CICS Liberty servers. For CICS Liberty servers, the value of **WLP_USER_DIR** is derived from the **WORK_DIR** and **WLP_OUTPUT_DIR** environment variables provided in the **JVMServer** JVM profile file
- The **serverName** directory structure and its initial contents are created when the server is created. **serverName** can be a mount point with a dedicated file system mounted at this mount point. This can be used to isolate servers to dedicated file systems.
- The number, size and output location of messages.log and trace files in the *logs* directory can be controlled with the Liberty **<logging>** configuration element or the output location controlled by using the **com.ibm.ws.logging.log.directory** Java directive as a JVM options override, more on this later.
- By default, the ‘owner’ of these directories and files is the identity that creates them.



Liberty basic: The server.xml file

Creating a Liberty server involves creating the server's required OMVS directory structure and the initial server configuration file, e.g., *server.xml*

The server's directory structure will be located at *{\$WLP_USER_DIR}/servers/serverName* and the *server.xml* file is key configuration file for any Liberty server.

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.3</feature>
    </featureManager>

    <!-- To access this server from a remote client, add a host
attribute to the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9080"
                  httpsPort="9443" />

    <!-- Automatically expand WAR files and EAR files -->
    <applicationManager autoExpand="true"/>

</server>
```

An administrator adds or removes features (security, and other required functions) as needed in the *featureManager* configuration elements.

Configures connectivity in *httpEndpoint* configuration elements

Adds other configuration elements as needed.

A sample initial server.xml configuration file

Always use the same value for WLP_USER_DIR when creating a server, starting a server or invoking administrative commands on a server.

Let's do a quick review of Unix permissions bits

Permission is controlled by three 3-bit octal numbers

Owner	Group	Other																																				
<table><thead><tr><th></th><th>Read</th><th>Write</th><th>Execute</th></tr></thead><tbody><tr><th>Bit</th><td>1</td><td>1</td><td>1</td></tr><tr><th>Base-2 Value</th><td>[4]</td><td>[2]</td><td>[1]</td></tr></tbody></table> <p>↓ 4 + 2 + 1 =</p> <p>7 The owner has READ, WRITE and EXECUTE</p>  <p>The owner of the file or directory</p> <p>chmod -R * u+rwx zceesrv1</p>		Read	Write	Execute	Bit	1	1	1	Base-2 Value	[4]	[2]	[1]	<table><thead><tr><th></th><th>Read</th><th>Write</th><th>Execute</th></tr></thead><tbody><tr><th>Bit</th><td>1</td><td>0</td><td>1</td></tr><tr><th>Base-2 Value</th><td>[4]</td><td>[2]</td><td>[1]</td></tr></tbody></table> <p>↓ 4 + 0 + 1 =</p> <p>5 The group has READ and EXECUTE, but not WRITE</p>  <p>IDs that are part of the group for the file or directory</p> <p>chmod g+rwx server.xml</p>		Read	Write	Execute	Bit	1	0	1	Base-2 Value	[4]	[2]	[1]	<table><thead><tr><th></th><th>Read</th><th>Write</th><th>Execute</th></tr></thead><tbody><tr><th>Bit</th><td>0</td><td>0</td><td>0</td></tr><tr><th>Base-2 Value</th><td>[4]</td><td>[2]</td><td>[1]</td></tr></tbody></table> <p>↓ 0 + 0 + 0 =</p> <p>0 Others have no access</p>  <p>IDs that are not the owner and not part of the group; that is, other</p> <p>chmod -R * o+rx resources chmod -R * o-w resources/security</p>		Read	Write	Execute	Bit	0	0	0	Base-2 Value	[4]	[2]	[1]
	Read	Write	Execute																																			
Bit	1	1	1																																			
Base-2 Value	[4]	[2]	[1]																																			
	Read	Write	Execute																																			
Bit	1	0	1																																			
Base-2 Value	[4]	[2]	[1]																																			
	Read	Write	Execute																																			
Bit	0	0	0																																			
Base-2 Value	[4]	[2]	[1]																																			
* indicates recursion																																						

Details of a Liberty servers' configuration directories and files



```
export WLP_USER_DIR=/var/zosconnect
```

	Permission	Bits	Owner	Group
\$WLP_USER_DIR		750	LIBSERV	LIBGRP
--servers/		750	LIBSERV	LIBGRP
-- server1_name1/		750	LIBSERV	LIBGRP
--bootstrap.properties		700	LIBSERV	LIBGRP
--jvm.options		700	LIBSERV	LIBGRP
--server.env		640	LIBSERV	LIBGRP
--server.xml		640	LIBSERV	LIBGRP
--apps/		750	LIBSERV	LIBGRP
--dropins/		750	LIBSERV	LIBGRP
--configDropins/		750	LIBSERV	LIBGRP
--defaults/		750	LIBSERV	LIBGRP
--overrides/		750	LIBSERV	LIBGRP
--logs/		750	LIBSERV	LIBGRP
--messages.log		666	LIBSERV	LIBGRP
--resources/		750	LIBSERV	LIBGRP
--security/		750	LIBSERV	LIBGRP
-- server2_name/		750	LIBSERV	LIBGRP
. . .				
-- server3_name/		750	LIBSERV	LIBGRP

--shared/	750	LIBSERV	LIBGRP
--apps/	750	LIBSERV	LIBGRP
--config/	750	LIBSERV	LIBGRP
--resources/	750	LIBSERV	LIBGRP

There are some questions that need answers with this in a production setting:

- If you have multiple people with a need to change configuration files, do you share the password of LIBSERV (the server's SAF identity)?
No. In fact, LIBSERV should be a PROTECTED identity with no password in the first place. Better to take advantage SAF SURROGAT so permitted users can switch to the owning ID so they can make changes.
- If you have multiple people with a need to read or update configuration files, do you simply connect them to LIBGRP?
No. The owner group may be granted access to other resources (on z/OS SAF profiles notably: SERVER) and you do not want others inheriting that. Better to make the configuration group be something different from the owner group and grant READ/WRITE through that group.
- **The shared directory structure is shared among all servers that have a common value for the WLP_USER_DIR environment variable. Each server can access common server configuration files using the shared.config.dir environment variable or access web applications using the shared.app.dir environment variable.**

Tech-Tip – Invoking OMVS commands using JCL

This JCL confirms that the server's SAF identity can invoke Java

```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=JOHNSON,REGION=0M,
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),USER=LIBSERV
//*****SET SYMBOLS
//*****EXPORT EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
//*****STEP JAVA - INVOKE THE java -version COMMAND
//*****JAVA EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//STDENV DD DUMMY
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export JAVA_HOME=&JAVAHOME; +
$JAVA_HOME/bin/java -version
```

```
java version "1.8.0_301"
Java(TM) SE Runtime Environment (build 8.0.6.35 - pmz6480sr6fp35-20210714_01(SR6 FP35))
IBM J9 VM (build 2.9, JRE 1.8.0 z/OS s390x-64-Bit Compressed References 20210622_7763 (JIT enabled, AOT
enabled)
OpenJ9   - b1f3adb
OMR     - c2f4a18
IBM    - c24a144)
JCL - 20210625_01 based on Oracle jdk8u301-b09
```

Tech-Tip: Using the SAF SURROGAT resource

RACF Surrogate access allows a designated administrative identity the ability to invoke commands and perform functions as if they were running under the identity that will be used for the z/OS Connect server started task. This may be useful because identities associated with started task are normally restricted and cannot be used for accessing TSO or OMVS shells,

Use the following examples as guides and create the surrogate resources and permit access. In these examples, ***LIBSERV*** represents the RACF identity under which the z/OS Connect server will be running and ***adminGrp*** represent the administrative RACF administrative group.

Define a SURROGAT profile for the server's SAF identity

RDEFINE SURROGAT BPX.SRV.*LIBSERV*

Define a SURROGAT submit profile to allow job submission as the server's SAF identity

RDEFINE SURROGAT *LIBSERV*.SUBMIT

Permit a member of the administrative group to act as a surrogate of the Liberty task identity

PERMIT BPX.SRV.*LIBSERV* CLASS(SURROGAT) ID(*adminGrp*) ACC(READ)

PERMIT *LIBSERV*.SUBMIT CLASS(SURROGAT) ID(*adminGrp*) ACC(READ)

Refresh the SURROGAT in storage profiles

SETROPTS RACLIST(SURROGAT) REFRESH

Now any identity in group *adminGrp* can submit JCL with the *USER=LIBSERV* parameter on the job card or use the OMVS switch user command (*su -s LIBSERV*) to execute OMVS scripts or commands as LIBSERV.

Tech/Tip: Using the SAF UNIXPRIV/FACILITY resources

An alternative to using a surrogate access is to permit the identity under which the customization will be done to enhanced Unix privileges. Specifically, permitting the identity to Unix privileges SUPERUSER.FILESYS, SUPERUSER.FILESYS.CHANGEPERMS and SUPERUSER.FILESYS.CHOWN.

- *Permit an administrative identity to write to any local directory or file*
PERMIT SUPERUSER.FILESYS CLASS(UNIXPRIV)
ID(adminUser) ACC(CONTROL)
- *Permit an administrative identity to change permission bit of any local directory or file*
PERMIT SUPERUSER.FILESYS.CHANGEPERMS CLASS(UNIXPRIV)
ID(adminUser) ACC(READ)
- *Permit an administrative identity to change the ownership of any directory or file*
PERMIT SUPERUSER.FILESYS.CHOWN CLASS(UNIXPRIV)
ID(adminUser) ACC(READ)
- *Permit an administrative identity switch to root (su -s root) or the Enable superuser mode(SU) Setup option in ISHELL*
PERMIT BPX.SUPERUSER CLASS(FACILITY) ID(adminUser) ACC(READ)
- *Refresh the UNIXPRIV and/or FACILITY instorage profiles*
SETROPTS RACLIST(UNIXPRIV,FACILITY) REFRESH

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.bpxb200/usspriv.htm

Use the power provided by this access by these commands carefully and only when necessary

Tech-Tip: Use JCL to create the *WLP_USER_DIR* directory structure

Permission bit are set properly by taking advantage of RACF **SURROGAT** or **UNIXPRIV** resources

Example of using **UNIXPRIV** privileges

```
/MYSERVER JOB CLASS=A,REGION=0M,MSGCLASS=H,NOTIFY=&SYSUID
//*****SET SYMBOLS
//*****EXPORT EXPORT SYMLIST=(*)
// SET WLPUSER='/var/wlp'
// SET USER='LIBGRP'
// SET GROUP='LIBSERV'
//*****Step CREATE - Use the mkdir command to create direcotries
//*****CREATE EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export WLP_USER_DIR=&WLPUSER; +
mkdir -p &WLPUSER/shared/apps; +
mkdir -p &WLPUSER/shared/config; +
mkdir -p &WLPUSER/shared/resources; +
chown -R &USER:&GROUP $WLP_USER_DIR/shared
```

Example of using **SURROGAT** privileges

```
/MYSERVER JOB CLASS=A,REGION=0M,MSGCLASS=H,NOTIFY=&SYSUID,USER=LIBSERV
//*****SET SYMBOLS
//*****EXPORT EXPORT SYMLIST=(*)
// SET WLPUSER='/var/wlp'
//*****Step CREATE - Use the mkdir command to create direcotries
//*****CREATE EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export WLP_USER_DIR=&WLPUSER; +
mkdir -p &WLPUSER/shared/apps; +
mkdir -p &WLPUSER/shared/config; +
mkdir -p &WLPUSER/shared/resources
```

N.B. Multiple OMVS commands can be entered as input to *BPXBATCH* utility. Individual OMVS commands are terminated with a semi-colon. The semi-colon should be followed by at least one space. A plus symbol is used to indicate continuation. It is a good practice to not start a command in column 1.

Tech-Tip: Using permission bits to control access



ID=**LIBSERV**
Group=**LIBGRP**

```
export JAVA_HOME=<path_to_64_bit_Java>
export WLP_USER_DIR=/var/zosconnect
./server create zceesrvr
```

/var/zosconnect	751	LIBSERV	LIBGRP
/servers	751	LIBSERV	LIBGRP
/zceesrv1	751	LIBSERV	LIBGRP
/apps	761	LIBSERV	LIBGRP
/configDropins	761	LIBSERV	LIBGRP
/overrides	761	LIBSERV	LIBGRP
/logs	771	LIBSERV	LIBGRP
/messages.log	644	LIBSERV	LIBGRP
/resources	751	LIBSERV	ADMGRP
/security	777	LIBSERV	LIBGRP
/zosconnect	751	LIBSERV	ADMGRP
/apis	761	LIBSERV	ADMGRP
/apiRequesters	761	LIBSERV	ADMGRP
/rules	761	LIBSERV	ADMGRP
/services	761	LIBSERV	ADMGRP
server.xml	460	LIBSERV	ADMGRP
/shared	750	LIBSERV	LIBGRP
/apps	750	LIBSERV	LIBGRP
/config	750	LIBSERV	LIBGRP

~~Often you may be tempted to use command chmod -R 777 *~~

Sample of OMVS commands to manage permission bits

```
export WLP_USER_DIR=/var/zosconnect
cd $WLP_USER_DIR
chmod o+x -R servers
chmod o+x servers/zceesrvr/resources
chmod -R o+x servers/zceesrvr/resources/*
chmod g+r -R servers
chmod g+r servers/zceesrvr/resources
chmod -R g+r servers/zceesrvr/resources/*
chmod g+w server.xml
```

Warning: Access for Owner (u), Group(g), Others(o) depend on user ID (UID) and group ID (GID) as stored with the directory or file, not the actual SAF identity or group.
This has implications when moving entire filesystems from one LPAR to another using utilities like ADRDSSU.

Creating Liberty Servers



Basic Liberty servers are created by invoking Liberty command *server create*

Invoking the command creates the server's required directory structure and populates key configuration files.

To create a basic Liberty server, use the Liberty *server create* command, as in `server create serverName`

- Where `serverName` is any value you wish, such as `wlpopsrv` or `wlpOpenIDAuthServer` and this value will be the name of the server instance. The default value for the serve name is `defaultServer`
- Environment variable `WLP_USER_DIR` must be set to determine the location of the configuration directory and files created by this command. The constant `servers` is appended to the value of this variable, e.g., `{$WLP_USER_DIR}/servers` and the server's name is appended to this root directory and full directory path is the location where the server's configuration files, and default directories are created, e.g., `{$WLP_USER_DIR}/servers/serverName`. The `WLP_USER_NAME` variables is required when starting a server and must be the same value used when the server was created. The default value for a Liberty server is `../wlp/usr`.

N.B. The name of the server does not have to be same as the started task name, as shown in this example (note in this example how the value for `WLP_USER_DIR` is provided by the `PATH` attribute of the `WLPUDIR` DD statement):

```
//WLPOPID PROC PARMS='wlpOpenIDAuthServer'  
///*  
// SET INSTDIR='/usr/lpp/liberty_zos/22.0.0.9'  
// SET USERDIR='/var/wlp'  
///*  
//STEP1 EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,  
// PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv &PARMS'  
//WLPUDIR DD PATH='&USERDIR.'  
//STDOUT DD SYSOUT=*  
//STDERR DD SYSOUT=*  
//MSGLOG DD SYSOUT=*  
//STDENV DD PATH='/etc/system.env', PATHOPTS=(ORDONLY)
```

Tech-Tip: Use JCL to invoke the *server* command using the TSO command processor IKJEFT01

```

//*****SET SYMBOLS
//*****EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
// SET WLPPATH='/usr/lpp/liberty_zos/22.0.0.9'
// SET SERVER='myServer'
// SET WLPUSER='/var/wlp'
//*****Step CREATE - Use the server command to create a server
//*****CREATE EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export JAVA_HOME=&JAVAHOME; +
export WLX_USER_DIR=&WLPUSER; +
&WLPPATH/bin/server create &SERVER +
--template=&TEMPLATE; +

```

Set and export JCL symbols variables

- Export the JCL symbols passed to OMVS environment
- Invoke the *server* command to create the Liberty server



Tech-Tip: Using JCL makes the creation and configuration of servers repeatable and portable

And take advantage of RACF SURROGAT and UNIXPRIV resources

Example of using **SURROGAT** privileges

```
//ZCEESRVR JOB CLASS=A,REGION=0M,NOTIFY=&SYSUID,USER=LIBSERV
//*****
//* SET SYMBOLS
//*****
//EXPORT EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
// SET ZEEPATH='/usr/lpp/IBM/zosconnect/v3r0'
// SET SERVER='zceesrvr'
// SET TEMPLATE='zosconnect:default'
// SET WLPUSER='/var/ats/zosconnect'
// SET USER='ATSSERV'
// SET GROUP='ATSGRP'
//*****
//* Step ZCEESRVR - Use the zosconnect command to create a server
//*****
//ZCEESRVR EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export JAVA_HOME=&JAVAHOME; +
export WLP_USER_DIR=&WLPUSER; +
&ZEEPATH/bin/zosconnect create &SERVER +
--template=&TEMPLATE; +
```

Example of using **UNIXPRIV** privileges

```
//ZCEESRVR JOB CLASS=A,REGION=0M,NOTIFY=&SYSUID
//*****
//* SET SYMBOLS
//*****
//EXPORT EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
// SET ZEEPATH='/usr/lpp/IBM/zosconnect/v3r0'
// SET SERVER='openApi3'
// SET TEMPLATE='zosconnect:openApi3'
// SET WLPUSER='/var/ats/zosconnect'
// SET USER='ATSSERV'
// SET GROUP='ATSGRP'
//*****
//* Step ZCEEAPI3 - Use the zosconnect command to create a server
//*****
//ZCEEAPI3 EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export JAVA_HOME=&JAVAHOME; +
export WLP_USER_DIR=&WLPUSER; +
&ZEEPATH/bin/zosconnect create &SERVER +
--template=&TEMPLATE; +
chown -R &USER:&GROUP $WLP_USER_DIR/servers/&SERVER
```



Liberty environment variables and server XML environment variables

- **WLP_USER_DIR** – This environment variable is used when a server is created to determine where the server's working directories will be created and where the initial *server.xml* file will be created. This variable is also used by the runtime environment to locate the server's existing working directories and the *server.xml* file. Also, the **WLP_USER_DIR** is used to set the shared variables.
- **JAVA_HOME** – The OMVS directory where the Java executables (*/bin* directory) can be located.
- **JVM_OPTIONS** – A z/OS Connect environment variables that provides Java options and/or system properties. The contents of **JVM_OPTIONS** is added to the *java* command line in the *zosconnect* startup script.
- **IBM_JAVA_OPTIONS** – An IBM JAVA environment variable (deprecated and eventually will be replaced by environment variable *OPENJ9_JAVA_OPTIONS*). Environment variable **IBM_JAVA_OPTIONS** variable can be used to provide Java options and/or system properties.
- **OPENJ9_JAVA_OPTIONS** – An OpenJ9 environment variable (eventually will replace the deprecated environment variable *IBM_JAVA_OPTIONS*). Environment variable *OPENJ9_JAVA_OPTIONS* variable can be used to provide Java options and/or system properties.

Note: Any Java option or system property using **JVM_OPTIONS** supersedes the same Java non-standard options or system property when provided by **IBM_JAVA_OPTIONS** or **OPENJ9_JAVA_OPTIONS**

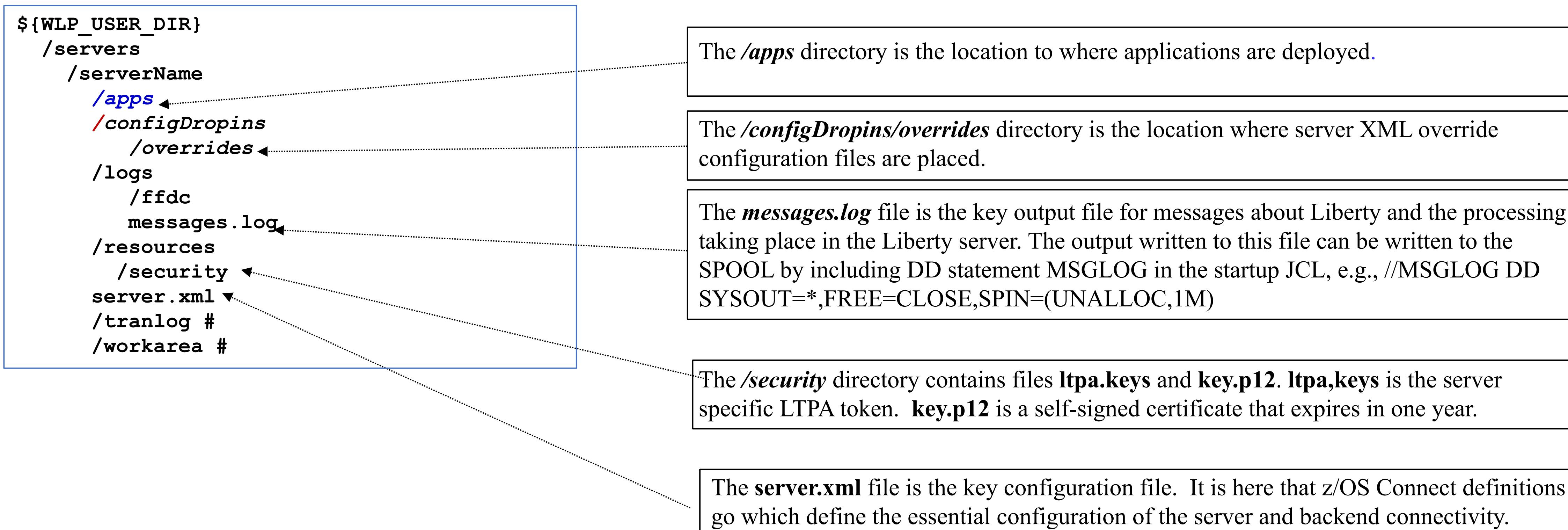
The following environment variables are automatically set by Liberty and can be used as variables in the server XML configuration files.

- **serverName** – whose value will be set to the name of the server and used to set the value of several other environment variables.
- **server.config.dir** – whose value will automatically be set to the value of variable **WLP_USER_DIR** concatenated with the name of the server, e.g. */var/liberty/servers/serverName*
- **shared.config.dir** – whose value will automatically be set to the value of variable **WLP_USER_DIR** concatenated with */shared/config*, e.g. */var/liberty/shared/config*
- **shared.app.dir** – whose value will automatically be set to the value of variable **WLP_USER_DIR** concatenated with */shared/apps*, e.g. */var/liberty/shared/apps*
- **wlp.server.name** - whose value will automatically be set to the value of the server as provided in the PARMS value provided in the JCL procedure.



Details of the contents of a server's directory structure

A server's configuration structure looks like this



- The `WLP_USER_DIR` environment variables sets the value of the root directory of the server's configuration files and directories, e.g., `WLP_USER_DIR=/var/zosconnect`
- #These directories maintain state information and it is a good practice is to add the `--clean` parameter to the server startup JCL, e.g., `PARMS='serverName --clean'`, especially after service is applied.



Tech-Tip: Displaying Liberty messages on the console and/or STDERR spool

server.xml

```
<zosLogging wtoMessage=
  "BAQR0657E,BAQR0658E,BAQR0660E,BAQR0686E,BAQR0687E"
  hardCopyMessage=
  "BAQR0657E,BAQR0658E,BAQR0660E,BAQR0686E,BAQR0687E"/>
```

MVS Console

```
18.12.02 STC00137 +BAQR0686E: Program CSCVINC is not available in the CICS region with
  811           connection ID cscvinc; service cscvincService failed.
18.12.02 STC00137 +BAQR0686E: Program CSCVINC is not available in the CICS region with
  812           connection ID cscvinc; service cscvincService failed.
19.07.12 STC00137 +BAQR0657E: Transaction abend MIJO occurred in CICS while using
  745           connection cscvinc and service cscvincService.
```

STDERR

```
ÝERROR  .. BAQR0686E: Program CSCVINC is not available in the CICS region with connection cscvinc and service cscvincService.
ÝERROR  .. BAQR0686E: Program CSCVINC is not available in the CICS region with connection cscvinc and service cscvincService.
ÝERROR  .. BAQR0657E: Transaction abend MIJO occurred in CICS while using CICS connection cscvinc and service cscvincService.
```



Liberty Java Directives for controlling output

com.ibm.ws.logging.console.format (consoleFormat) - The required format for the console. Valid values are basic or json format.

com.ibm.ws.logging.console.log.level (consoleLogLevel) - This filter controls the granularity of messages that go to the console. The valid values are INFO, AUDIT, WARNING, ERROR, and OFF. By default, the console log level is set to AUDIT.

com.ibm.ws.logging.hideMessage (hideMessage) - Use this attribute to configure the messages that you want to hide from the *console.log* and *message.log* files. If the messages are configured to be hidden, then they are redirected to the *trace.log* file.

com.ibm.ws.logging.log.directory (logDirectory) - Use this attribute to set a directory for all log files, excluding the *console.log* file, but including FFDC. The default log location path is *WLP_OUTPUT_DIR/serverName/logs*

com.ibm.ws.logging.max.file.size (maxFileSize) - The maximum size (in MB) that a log file can reach before it is rolled. The Liberty runtime does only size-based log rolling. To disable this attribute, set the value to 0. The maximum file size is approximate. By default, the value is 20.

com.ibm.ws.logging.max.files (maxFiles) - If a maximum file size exists, this setting is used to determine how many of each of the log files are kept. This setting also applies to the number of exception logs that summarize exceptions that occurred on any day. So, if this number is 10, you might have 10 message logs, 10 trace logs, and 10 exception summaries in the *ffdc* directory. The default value is 2.

com.ibm.ws.logging.message.format (messageFormat) - The required format for the *messages.log* file. Valid values are basic or json format. By default, *messageFormat* is set to the environment variable *WLP_LOGGING_MESSAGE_FORMAT* (if set) or basic.

com.ibm.ws.logging.trace.file.name (traceFileName) - The *trace.log* file is only created if additional or detailed trace is enabled. *stdout* is recognized as a special value; and causes trace to be directed to the original standard out stream.

bootstrap.properties example:

```
com.ibm.ws.logging.message.file.name=basqstrtMessages.log  
com.ibm.ws.logging.log.directory=/u/common/logs
```

N.B. *consoleFormat*, *logDirectory*, etc. can be specified in the *<logging/>* Liberty configuration element. Note the recommendation for the attributes in red is for them to be provided in Java directives.



Use “include” files to manage configuration elements in shared files

- Setup a server.xml using ‘include’ statements and allow other administrator to manage those included files, but not the server.xml itself.
- Control what configuration can be overridden in included files using the ‘onConflict’ option provided with the include element (see Ignore, Replace, Merge).

https://www.ibm.com/support/knowledgecenter/en/SSAW57_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/cwlp_config_include.html

server.xml (owned by ID ADMIN1)

```
<include location="${server.config.dir}/includes/db2.xml onConflict="IGNORE"/>
<include location="${server.config.dir}/includes/cics.xml onConflict="IGNORE"/>
<include location="${server.config.dir}/includes/imsDb.xml onConflict="IGNORE"/>
<featureManager>
  <feature>zosconnect:zosConnect-2.0</feature>
  <feature>zosconnect:zosConnectCommands-1.0</feature>
  <feature>apiDiscovery-1.0</feature>
<featureManager>
```

db2.xml (owned and managed by a DBA)

```
<server description="Db2 REST">
  <zosconnect_zosConnectServiceRestClientConnection id="Db2Conn" host="wg31.washington.ibm.com" port="2446" basicAuthRef="dsn2Auth" />
  <zosconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth" applName=DSN2APPL/>
</server>
```

cics.xml (owned and managed by a CICS administrator)

```
<server description="CICS">
  <featureManager> <feature>zosconnect:cicsService-1.0</feature> </featureManager>
  <zosconnect_cicsIpicConnection id="catalog" host="wg31" port="1491"/>
  <zosconnect_cicsIpicConnection id="cscvinc" host="wg31" port="1493"/>
</server>
```

imsDB.xml (owned and managed by a IMS administrator)

```
<server description="IMS DATABASE">
  <featureManager> <feature>zosconnect:dbService-1.0</feature> </featureManager>
  <connectionFactory id="DFSTIVPAComm" > <properties.imsudbJLocal databaseName="DFSTIVPA" datastoreName="IVP1" driverType="4" portNumber="5555" datastoreServer="wg31" user="USER1" password="USER1" flattenTables="True"/> </connectionFactory>
</server>
```

Nesting of an include file within a include file is possible



Review configuration conflicts

```
ÝAUDIT  " CWWKG0102I: Found conflicting settings for cscvincAPI instance of zosconnect_endpointConnection configuration.  
Property port has conflicting values:  
  Value 9443 is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value 9443 is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value 9463 is set in file:/var/zosconnect/servers/myServer/includes/oauth.xml.  
Property port will be set to 9463.  
Property host has conflicting values:  
  Value https://dvipa.washington.ibm.com is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value https://dvipa.washington.ibm.com is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value https://mpz3.washington.ibm.com is set in file:/var/zosconnect/servers/myServer/includes/oauth.xml.  
Property host will be set to https://mpz3.washington.ibm.com.  
Property authenticationConfigRef has conflicting values:  
  Value mySAFAuth is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value myoAuthConfig is set in file:/var/zosconnect/servers/myServer/includes/oauth.xml.  
Property authenticationConfigRef will be set to myoAuthConfig.
```

onConflict="MERGE" Conflicting elements will be merged, and the last value encountered will be used.

onConflict="REPLACE" When elements conflict, the element in the included file will be ignored

onConflict="IGNORE" Conflicting elements in the included file are ignored.

Tech-Tip: Understand the difference between a singleton XML configuration entry versus entries where multiple entries are allowed. During the XML parsing of the configuration files, multiple occurrences of singleton XML configuration entry are combined as they are encountered whereas multiple entries are kept separate based on the value of their *id*.



Use a *bootstrap.properties* file to help customize a server's XML configuration#

zceesrv1's bootstrap.properties

```
httpPort=9080
httpsPort=9443
ipicPort=1491
host=*
cicsHost=wg31.washington.ibm.com
network=ZOSCONN1
applid=ZOSCONN1
com.ibm.ws.zos.core.angelName=namedAngel
bootstrap.include=/var/liberty/common-bootstrap.properties
```

zceesrv2's bootstrap.properties

```
httpPort=9090
httpsPort=9453
ipicPort=1492
host=wg31.washington.ibm.com
cicsHost=wg31.washington.ibm.com
network=ZOSCONN2
applid=ZOSCONN2
com.ibm.ws.zos.core.angelName=namedAngel
bootstrap.include=/var/liberty/common-bootstrap.properties
```

server.xml

```
<!-- To access this server from a remote client, add a host attribute to the following
element, e.g. host="*" -->
<httpEndpoint id="defaultHttpEndpoint"
               host="${host}"
               httpPort ="${httpPort}"
               httpsPort ="${httpsPort}" />
```

ipicIDProp.xml

```
<zosconnect_cicsIpicConnection id="catalog"
                                host ="${cicsHost}" port ="${ipicPort}"
                                zosConnectNetworkid ="${network}"    zosConnectApplid ="${applid}"/>

<zosconnect_cicsIpicConnection id="cscvinc"
                                host ="${cicsHost}" port ="${ipicPort}"
                                zosConnectNetworkid ="${network}"    zosConnectApplid ="${applid}"/>

<zosconnect_cicsIpicConnection id="miniloan"
                                host ="${cicsHost}" port ="${ipicPort}"
                                zosConnectNetworkid ="${network}"    zosConnectApplid ="${applid}"/>
```

**N.B. Java directives
can also be provided.**

**N.B. Boot strap
properties can be
included from a
common boot strap
properties file**

Tech-Tip: A suggestion for managing the server.xml configuration file

Default server.xml configuration file

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
    <!-- Enable features -->
    <featureManager>
    - - - - -
    </featureManager>

    <!-- To access this server from a remote client add a host attribute
    <httpEndpoint id="defaultHttpEndpoint"
        host="*"
        httpPort="9080"
        httpsPort="9443" />
    <!-- add cors to allow cross origin access, e.g. when using swagger
    <cors id="defaultCORSConfig"
        domain="/"
        allowedOrigins="*"
        allowedMethods="GET, POST, PUT, DELETE, OPTIONS"
        allowedHeaders="Origin, Content-Type, Authorization, Cache-Control"
        allowCredentials="true"
        maxAge="3600"/>
    - - - - - 30 Line(s) not Displayed
    </server>
```

Modified server.xml configuration file

```
<server description="zCEE Server">
    <include location="${server.config.dir}/includes/safSecurity.xml"/>
    <include location="${server.config.dir}/includes/ipicIDProp.xml"/>
    <include location="${server.config.dir}/includes/keyring.xml"/>
    <include location="${server.config.dir}/includes/groupAccess.xml"/>
    <include location="${server.config.dir}/includes/shared.xml"/>
    <include location="${server.config.dir}/includes/apiRequesterHTTPS.xml"/>
    <include location="${server.config.dir}/includes/imsDatabase.xml"/>
    <!-- Enable features -->
    <featureManager>
    - - - - -
    </featureManager>
    <!-- To access this server from a remote client add a host attribute
    <httpEndpoint id="defaultHttpEndpoint"
        host="${host}"
        httpPort="${httpPort}"
        httpsPort="${httpsPort}" />
    - - - - - 36 Line(s) not Displayed
    </server>
```

The simplifies administration by :

- Using a *bootstrap.properties* file to customize the ports in the *server.xml* file.
- Using “include” statements to make further changes such as adding additional features and additional XML configuration elements.
- Review <https://www.ibm.com/docs/en/was-liberty/nd?topic=liberty-configuration-element-merging-rules> to understand merging rules.
- Consider providing configuration elements by placing server XML files in the .../configDropins/original subdirectory.

Tech-Tip: Sharing XML configuration files between servers using \${shared.config.dir}

Add an “includes” subdirectories {shared.config.dir} with a symbolic links to a common location. This common directory can be accessed from multiple servers on a single or from multiple LPARs. Additions and updates to the “include” files are then made in one single administrative directory.

Symbolic links from servers shared configuration \${shared.config.dir} to common directory

Symbolic links to a shared local LPAR directory

```
ln -s /var/shared/zosconnect/includes /var/zosconnect/shared/config
ln -s /var/shared/zosconnect/includes /var/ats/zosconnect/shared/config
ln -s /var/shared/zosconnect/includes /var/wsc/zosconnect/shared/config
```

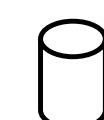
Symbolic links to a shared Sysplex directory *

```
ln -s /global/zosconnect/includes /var/zosconnect/shared/config
ln -s /global/zosconnect/includes /var/ats/zosconnect/shared/config
ln -s /global/zosconnect/includes /var/wsc/zosconnect/shared/config
```

```
<include location="${shared.config.dir}/safSecurity.xml"/>
<include location="${shared.config.dir}/ipicIDProp.xml"/>
<include location="${shared.config.dir}/keyringOutboundMutual.xml"/>
<include location="${shared.config.dir}/groupAccess.xml"/>
<include location="${shared.config.dir}/shared.xml"/>
<include location="${shared.config.dir}/db2.xml"/>
<include location="${shared.config.dir}/oauth.xml"/>
```



/var/shared/zosconnect/includes



/global/zosconnect/includes

Contents of the common “includes” directory

```
basicSecurity.xml
db2.xml
db2TLS.xml
groupAccess.xml
ipic.xml
ipicIDProp.xml
keyringInbound.xml
keystore.xml
keyringMutual.xml
keyringOutboundMutual.xml
safSecurity.xml
```

The server.xml file contains these “include” statements and the server XML is portable regardless of the underlying filesystem infrastructure.

For example, changing *basicSecurity.xml* to *safSecurity.xml* and refreshing the configuration changes security from basic to SAF

F ZCEESRVR ,REFRESH,CONFIG



Sharing XML configuration files – using ‘variables’ files

“variables” files whose names are based on the name of the server, e.g., \${wlp.server.name}

server.xml

```
<server description="new server">
<include location="${server.config.dir}/includes/safSecurity.xml"/>
<include location="${server.config.dir}/includes/${wlp.server.name}.xml"/>

    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:zosConnect-2.0</feature>
        <feature>zosconnect:zosConnectCommands-1.0</feature>
    </featureManager>
```

myServer.xml

```
<variable name="unauthenticatedUser" value="WSGUEST" />
<variable name="profilePrefix" value="BBGZDFLT" />
```

zceepid.xml

```
<variable name="unauthenticatedUser" value="ZCGUEST" />
<variable name="profilePrefix" value="EMJZDFLT" />
```

safSecurity.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="SAF security">

    <!-- Enable features -->
    <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>zosSecurity-1.0</feature>
    </featureManager>

    <webAppSecurity allowFailOverToBasicAuth="true" />
    <safRegistry id="saf" />
    <safAuthorization racRouteLog="ASIS" />
    <safCredentials unauthenticatedUser="${unauthenticatedUser}"
        profilePrefix="${profilePrefix}" />
</server>
```



Using the Liberty server's configuration drop-in's directories

Located in the same directory as the *server.xml* configuration file.

- Configuration files in the */default* directory provides defaults for configuration elements not present in *server.xml*
- Configuration files in the */overrides* directory adds to or replaces the configuration elements found in *server.xml*

```
 ${WLP_USER_DIR}
  /servers
    /serverName
      /apps
      /configDropins
        /default
        /overrides
    /logs
      /ffdc
      messages.log
    /resources
      /security
  server.xml
  /tranlog
  /workarea
```

```
commonFeatures.xml
<server description="Common Server Features">

  <!-- Enable features -->
  <featureManager>
    <feature>adminCenter-1.0</feature>
    <feature>restConnector-2.0</feature>
  </featureManager>

  <remoteFileAccess>
    <readDir>/var/zcee/includes</readDir>
    <readDir>/global/zosconnect/includes</readDir>
    <writeDir>${server.config.dir}</writeDir>
  </remoteFileAccess>

</server>
```

```
safSecurity.xml
<?xml version="1.0" encoding="UTF-8"?>
<server description="SAF security">

  <!-- Enable features -->
  <featureManager>
    <feature>appSecurity-2.0</feature>
    <feature>zosSecurity-1.0</feature>
  </featureManager>

  <webAppSecurity allowFailOverToBasicAuth="true" />
  <safRegistry id="saf" />
  <safAuthorization racRouteLog="ASIS" />
  <safCredentials unauthenticatedUser="${unauthenticatedUser}"
    profilePrefix="${profilePrefix}" />
</server>
```

Another directory that must be manually created.



IBM Washington Systems Center

Tech-Tip: Consider simplifying administration by sharing include files and by using server variables

Default server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
<include location="${server.config.dir}/includes/${wlp.server.name}.xml"/>

    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:zosConnect-3.0</feature>
        <feature>openapi-3.0</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute
    to the following element, e.g. host="*"
    <httpEndpoint id="defaultHttpEndpoint"
        host="*"
        httpPort="9080"
        httpsPort="9443" />
```

`${server.config.dir}/includes/${wlp.server.name}.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
<variable name="httpPort" value="9081"/>
<variable name="httpsPort" value="9445"/>
<variable name="hostName" value="*"/>
<variable name="CICS_HOST" value="wg31.washington.ibm.com"/>
<variable name="CICS_PORT" value="1491"/>
<variable name="DB2_HOST" value="wg31.washington.ibm.com"/>
<variable name="DB2_PORT" value="2446"/>
<variable name="DB2_USERNAME" value="USER2"/>
<variable name="DB2_PASSWORD" value="USER2"/>
<include location="${shared.config.dir}/safSecurity.xml"/>
<include location="${shared.config.dir}/httpEndpoint.xml"/>
<include location="${shared.config.dir}/db2.xml"/>
<include location="${shared.config.dir}/cics.xml"/>
<include location="${shared.config.dir}/keystore.xml"/>
</server>
```

`${server.config.dir}/includes/httpEndpoint.xml`

```
<${server.config.dir}/includes/httpEndpoint.xml"/>
<server description="basic security">
    <httpEndpoint id="defaultHttpEndpoint"
        host="${hostName}"
        httpPort="${httpPort}"
        httpsPort="${httpsPort}" />
</server>
```

`${server.config.dir}/includes/db2.xml`

```
<${server.config.dir}/includes/db2.xml"/>
<?xml version="1.0" encoding="UTF-8"?>
<server description="Default server">
    <featureManager>
        <feature>zosconnect:db2-1.0</feature>
    </featureManager>
    <zosconnect_credential user="${DB2_USERNAME}"
        password="${DB2_PASSWORD}" id="commonCredentials" />
    <zosconnect_db2Connection id="db2Conn" host="${DB2_HOST}"
        port="${DB2_PORT}" credentialRef="commonCredentials" />
</server>
```

`${server.config.dir}/includes/cics.xml`

```
<${server.config.dir}/includes/cics.xml"/>
<server description="CICS IPIC connections">
    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:cics-1.0</feature>
    </featureManager>
    <zosconnect_cicsIpicConnection id="cicsConn" host="${CICS_HOST}"
        port="${CICS_PORT}" />
</server>
```



For example, the MQ Console Liberty configuration uses a common R/O configuration file

An MQ Liberty server's `server.xml` only contain 'include' commands.

`server.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
This file defines the configuration for the mqweb server. Please do not
make changes to this file. Any changes made by users should be in the
referenced mqwebuser.xml file.
-->
<server>
    <include location="${wlp.install.dir}/mq/etc/mqweb.xml"/>
    <include location="mqwebuser.xml"/>
</server>
```

'Include' file `mqweb.xml` is provided by MQ and usually a read only file. It sets default values for variables and the required Liberty configuration elements.

'Include' file `mqwebuser.xml` is a read/write file and provides overrides for the default values of variables and any other Liberty configuration elements.



MQ Consoles sharing XML configuration files – using '*variables*' files

```
/usr/lpp/mqm/V9R3M4/web/mqm/etc/mqweb.xml

<?xml version="1.0" encoding="UTF-8"?>
<server>

    <featureManager>
        <feature>jaxrs-2.1</feature>
        <feature>ssl-1.0</feature>
        <feature>jndi-1.0</feature>
        <feature>concurrent-1.0</feature>
        <feature>websocket-1.0</feature>
        <feature>applicationMonitorMQ-1.0</feature>
    </featureManager>

    <!--
    Configurable properties. These can be overridden by setting appropriate
    -->
    <variable name="httpHost" value="localhost"/>
    <variable name="httpPort" value="-1"/>
    <variable name="httpsPort" value="9443"/>
    <variable name="mqRestRequestTimeout" value="30"/>
    <variable name="mqConsoleAutostart" value="true"/>
    <variable name="mqRestAutostart" value="true"/>
    <variable name="mqRestGatewayQmgr" value="" />
    <variable name="mqRestGatewayEnabled" value="true"/>
    <variable name="traceSpec" value="*=info"/>
    <variable name="maxTraceFileSize" value="20"/>
    <variable name="maxTraceFiles" value="2"/>
    <variable name="maxMsgTraceFileSize" value="200"/>
    <variable name="maxMsgTraceFiles" value="5"/>
    <variable name="ltpaExpiration" value="120"/>
    <variable name="ltpaCookieName" value="LtpaToken2_${httpsPort}"/>
    <variable name="secureLtpa" value="true"/>
    <variable name="mqRestCsrfValidation" value="true"/>
    <variable name="mqRestCorsAllowedOrigins" value="" />
    <variable name="mqRestCorsMaxAgeInSeconds" value="0"/>
    <variable name="mqRestMftCoordinationQmgr" value="" />
    <variable name="mqRestMftCommandQmgr" value="" />
    <variable name="mqRestMftEnabled" value="false"/>
    <variable name="mqRestMftReconnectTimeoutInMinutes" value="30"/>
    <variable name="mqRestMessagingEnabled" value="true"/>
    <variable name="mqRestMessagingMaxPoolSize" value="20"/>
```

mqwebuser.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
    <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>zosSecurity-1.0</feature>
        <feature>basicAuthenticationMQ-1.0</feature>
        <feature>apiDiscovery-1.0</feature>
    </featureManager>

    <enterpriseApplication id="com.ibm.mq.console"/>
    <enterpriseApplication id="com.ibm.mq.rest"/>

    <safAuthorization racRouteLog="ASIS"/>
    <safRegistry id="saf"/>
    <safAuthorization id="saf"
        reportAuthorizationCheckDetails="true" />
    <safCredentials profilePrefix="MQWEB" unauthenticatedUser="WSGUEST"/>

    <webAppSecurity allowFailOverToBasicAuth="true" />

    <variable name="httpsPort" value="1419"/>
    <variable name="httpHost" value="*"/>
    <variable name="mqRestMessagingEnabled" value="true"/>

    <b><config monitorInterval="5s" updateTrigger="mbean"/></b>

</server>
```



MQ Console/REST servers are created by invoking the MQ *crtmqweb* command

```
//*****  
///* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
// SET MQPATH='/usr/lpp/mqm/V9R3M4/web'  
// SET WLPUSER='/var/mqm/V9R3M4'  
// SET USER='MQSERV'  
// SET GROUP='MQGRP'  
//*****  
///* Step CRTMQWEB - Use the crtmqweb command  
//*****  
//CRTMQWEB EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
export JAVA_HOME=&JAVAHOME; +  
export WLP_USER_DIR=&WLPUSER; +  
&MQPATH/bin/crtmqweb &WLPUSER -p MQ; +  
chown -R &USER:&GROUP $WLP_USER_DIR/servers/&SERVER
```

- Set JCL symbols

- Export the JCL symbols passed to the OMVS environment
- Invoke the *crtmqweb* command to create the MQ Console and REST server. The default value for a MQ Console server is */var/mqm/web/installation1*
- Use the chown command to set the correct owner/group settings



CICS Liberty servers are created by defining and starting a CICS JVM Server resource

These are the Liberty sample JVM profiles provided by CICS that can be used as templates.

DFHWLP - The supplied sample profile for a Liberty JVM server.

EYUCMCIJ - The supplied sample profile for a CMCI Liberty JVM server within CICPlex SM.

EYUMSSJ - The supplied sample profile for a CMCI Liberty JVM server in a single CICS region.

These are the non-Liberty sample JVM profiles provided by CICS

DFHJVMAX.jvmprofile - The supplied sample profile for an Axis2 JVM server.

DFHJVMST.jvmprofile - The supplied sample profile for a JVM server for a Security Token Service.

DFHOSGI.jvmprofile - The supplied sample profile for an OSGi JVM server.

The screenshot shows a terminal window titled 'WG31# - 3270' running under 'CICS RELEASE = 0740'. The command entered is:

```
OVERTYPE TO MODIFY
CEDA ALTer JVmserver( DFHWLP )
JVmserver      : DFHWLP
Group          : DFH$WLP
DEscription   ==> CICS JVM server to run WLP samples
Status         ==> Enabled
                Enabled | Disabled
(Mixed Case)
Jvmprofile    ==> DFHWLP
Lerunopts     ==> DFHAXRO
Threadlimit   ==> 015
                  1-256
DEFINITION SIGNATURE
DEFinetime    : 07/26/22 17:35:22
CHANGETime   : 07/26/22 17:35:22
CHANGEUsrid   : JOHNSON
CHANGEAGEnt   : CSDBatch
CHANGEAGRel   : 0740
                CSDBatch | CSDBatch
```

At the bottom of the screen, the system ID and application ID are displayed as 'SYSID=CICS APPLID=CICS53Z' and 'DSN=CICSTS.CICS61Z.DFHCS'. The function key bar at the bottom includes PF 1 through PF 12 and CNCL, along with other standard keys like HELP, COM, END, and CRSR.



CICS Liberty sample JVM profiles are a starting point

DFHWLP.jvmpfile

```
JAVA_HOME=/usr/lpp/java/J8.0_6  
WORK_DIR=.  
WLP_INSTALL_DIR=&USSHOME;/wlp  
-Dcom.ibm.cics.jvmserver.wlp.a  
-Xms128M  
-Xmx256M  
-Xms01M  
-Xgcpolicy:gencon  
-Xscmx256M  
-Xshareclasses:name=cicsts.&AP  
-Xtune:virtualized  
-Dcom.ibm.tools.attach.enable=  
-Dfile.encoding=ISO-8859-1  
_BPXK_DISABLE_SHLIB=YES
```

CSCWLP.jvmpfile

```
PRINT_JVM_OPTIONS=YES  
JAVA_HOME=/usr/lpp/java/J17.0_64/  
WLP_INSTALL_DIR=&USSHOME;/wlp  
WORK_DIR=/var/cicsts  
WLP_OUTPUT_DIR=./&APPLID;/&JVM SERVER;  
-Dcom.ibm.cics.jvmserver.wlp.autoconfigure=true  
-Dcom.ibm.cics.jvmserver.controller.timeout=600000  
-Dcom.ibm.cics.jvmserver.wlp.server.host=*  
-Dcom.ibm.cics.jvmserver.wlp.server.http.port=9094  
-Dcom.ibm.cics.jvmserver.wlp.server.https.port=9497  
-Dcom.ibm.cics.jvmserver.wlp.saf.profilePrefix=ATSZDFLT  
-Dcom.ibm.cics.jvmserver.wlp.server.keystore.location=safkeyring:///Liberty.KeyRing 6.2  
-Dcom.ibm.cics.jvmserver.wlp.server.keystore.type=JCERACFKS 6.2  
-Djava.protocol.handler.pkgs=com.ibm.crypto.zsecurity.provider (IBM Semeru)  
-Dcom.ibm.ws.zos.core.anglName=CICSANGL  
STDOUT=/DD:JVMOUT  
STDERR=/DD:JVMERR  
JVMTRACE=/DD:JVMTRACE  
JVMLOG=/DD:JVMLOG  
-Xms128M  
-Xmx256M  
-Xms01M  
-Xgcpolicy:gencon  
-Xscmx256M  
-Xshareclasses:name=cicsts.&APPLID;,groupAccess,nonfatal  
-Xtune:virtualized  
-Dcom.ibm.tools.attach.enable=no  
-Dfile.encoding=ISO-8859-1  
_BPXK_DISABLE_SHLIB=YES
```

z/OSMF servers are started during an IPL or manually

SYS1.PARMLIB(IEASYS##)

IZU=00,

SYS1.PARMLIB(IZUPRM##)

HOSTNAME('*')
 HTTP_SSL_PORT(9433)
 JAVA_HOME('/usr/lpp/java/J11.0_64')
 KEYRING_NAME('IZUKeyring.IZUDFLT')
 ANGEL_PROC('IZUANG1')
 AUTOSTART('CONNECT')
 USER_DIR('/global/zosmf')
 UNAUTH_USER(IZUGUEST)

```
//IZUSVR1  PROC PARM='zosmfServer',      /* Server parms */
//                      ROOT='/usr/lpp/zosmf',   /* zOSMF install root */
//                      WLPDIR='/usr/lpp/zosmf/liberty', /* Liberty DIR */
//                      OUTCLS='*',           /* Sysout class */
//                      USERDIR='/global/zosmf', /* Config dir */
//                      TRACE='N',            /* Trace option */
//                      KCINDEX='N',          /* KC index rebuild flag */
//                      IZUPRM='##',           /* Parmlib suffixes or PREV */
//                      SERVER='AUTOSTART',   /* AUTOSTART server */
//                      Z='0',                /* Reserved for IBM */
//                      IZUMEM='NOLIMIT'       /* Server memlimit */

/*-----*/
/* Parse z/OSMF PARMLIB member
*-----*/
//ZPARM EXEC PGM=IZUPARMS,REGION=0M,
// PARM='/IZUPRM=&IZUPRM,TRACE=&TRACE,USERDIR=&USERDIR,SERVER=&SERVER,Z
//                   =&Z'
//DFLTCFG DD PATH='&ROOT./defaults/configuration.defaults'
//STDOUT  DD SYSOUT=&OUTCLS
//STDERR  DD SYSOUT=&OUTCLS

/*-----*/
/* Configure z/OSMF server
*-----*/
//CONFZMF EXEC PGM=BPXBATCH,REGION=0M,COND=(0,LT),
// PARM='SH &ROOT./bin/izuconfig.sh &ROOT &USERDIR &TRACE &KCINDEX'
//SYSPRINT DD SYSOUT=&OUTCLS
//SYSOUT   DD SYSOUT=&OUTCLS
```

z/OSMF servers are started during an IPL or manually

/global/zosmf/configuration/active_configuration

```
IZU_APPSERVER_HOSTNAME=WG31.WASHINGTON.IBM.COM
IZU_JWKS_HOSTNAME=WG31.WASHINGTON.IBM.COM
IZU_HTTP_SSL_PORT=9433
IZU_HTTP_PORT=-1
----- 44 Line(s) not Displayed
JAVA_HOME=/usr/lpp/java/J11.0_64
----- 10 Line(s) not Displayed
IZU_CONFIG_DIR=/global/zosmf/configuration
IZU_DATA_DIR=/global/zosmf/data
----- 74 Line(s) not Displayed
JVM_OPTIONS=-Xscmaxaot30m
```

```
/*
-----*
/* Start the Websphere Liberty Profile server
/*
/* WLPUDIR - PATH DD that points to the Liberty Profile's "user"
directory. If the DD is not allocated, the user
directory location defaults to the wlp/usr directory
in the install tree.
/* STDOUT - Destination for stdout (System.out)
/* STDERR - Destination for stderr (System.err)
/* STDENV - Initial Unix environment - read by the system. The
installation default and server specific server
environment files will be merged into this environment
before the JVM is launched.
```

/global/zosmf/configuration/servers/zosmfServer

```
bootstrap.properties
izu.config.prop
jvm.options
server.env
server.xml
```

```
//ZOSMF EXEC PGM=BPXBATSL,REGION=0M,COND=(0,LT),
// MEMLIMIT=&IZUMEM.,TIME=NOLIMIT,
// PARM='PGM &WLPDIR./lib/native/zos/s390x/bbgzsrv --clean &PARMS'
//WLPUDIR DD PATH='&USERDIR./configuration'
//STDOUT DD SYSOUT=&OUTCLS
//STDERR DD SYSOUT=&OUTCLS
```

z/OS Connect servers are created using the z/OS Connect `zosconnect` command



To create a z/OS Connect server, use the `zosconnect` command using one of these templates, as in:

`zosconnect create serverName --template=templateName`

Where *templateName* can be:

- **`zosconnect:apiRequester`** for an OpenAPI2 z/OS Connect API requester enabled server
- **`zosconnect:default`** template for base OpenAPI2 z/OS Connect servers
- **`zosconnect:openApi3`** template for base OpenAPI3 z/OS Connect native servers
- **`zosconnect:openApi3Requester`** template for base OpenAPI3 z/OS Connect native API requester servers

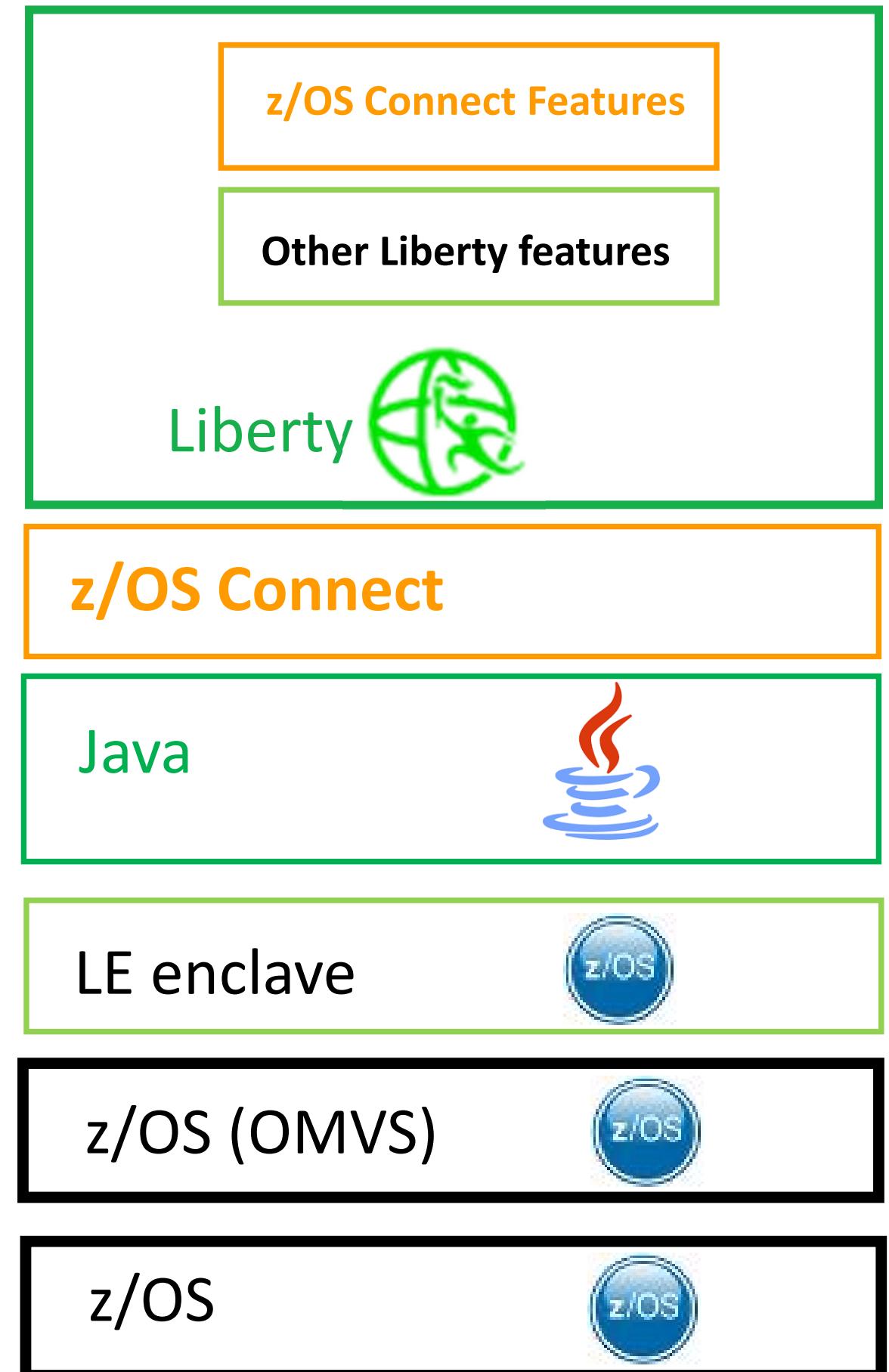
- Where *serverName* is any value you wish, such as `zceesrvr` or `zCEEServer`, and this value will be the name of the server instance. The templates can be found in directory `/usr/lpp/IBM/zosconnect/v3r0/runtime/templates/servers`.
- Environment variable **`WLP_USER_DIR`** will be used to set the location of the configuration directory and files created by this command, default location is `/var/zosconnect/servers` where `/var/zosconnect` is default value for `WLP_USER_DIR` for z/OS Connect.
- The `zosconnect:openApi3` template installs feature **`zosconnect:zosConnect-3.0`**. z/OS Connect service provider features, e.g., `zosconnect:cics-1.0`, `zosconnect:mqService-1.0`, `zosconnect:dbService-1.0` and `imsmobile:imsmobile-2.0` have dependencies on feature **`zosconnect:zosConnect-2.0`** and are not compatible with feature **`zosconnect:zosConnect-3.0`**.
- The `zosconnect:openApi3Requester` template installs feature **`zosconnect:oasRequester-1.0`**.
- See URL <https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=reference-configuration-elements> for a current list of OpenAPI2 configuration elements and URL <https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0?topic=reference-configuration-elements> for the current list of OpenAPI3 configuration elements.



Adding z/OS Connect to a Liberty server adds an additional software layer

z/OS Connect is both a Java application and a set of Liberty features

- The z/OS Connect Java application provides code that spawns a Liberty process *
- The z/OS Connect Liberty features provides a REST interface to user application artifacts that access to common z/OS subsystems, e.g., CICS, Db2, IMS, MQ, etc. and external REST endpoints.



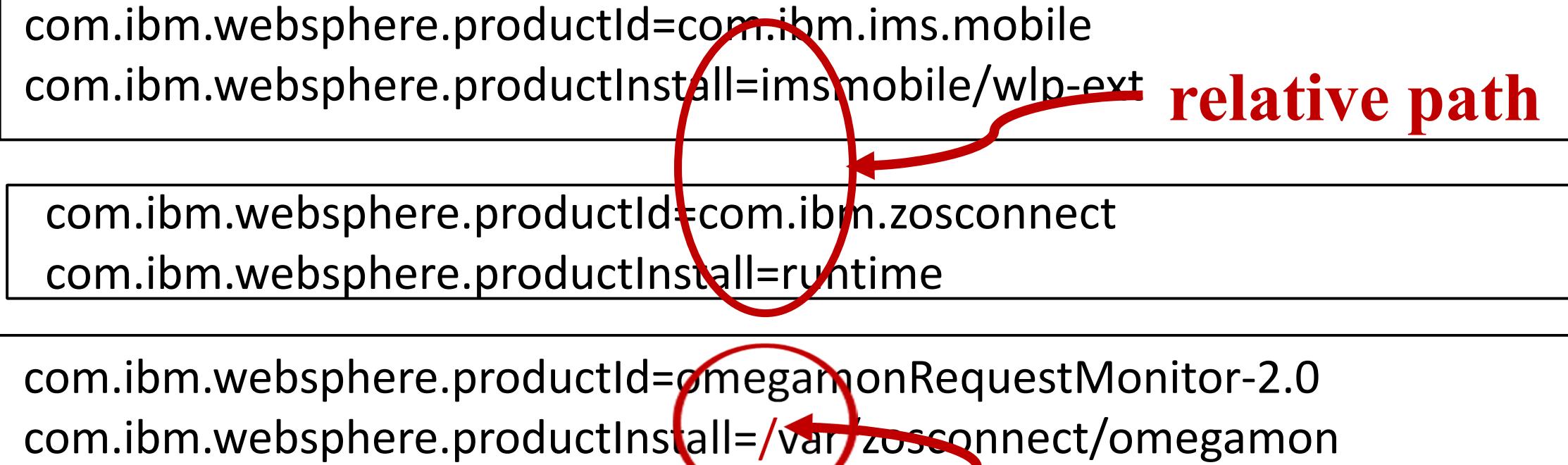
* z/OS Connect starts a Liberty process using a system programming interface (SPI). See the Note regarding environment variables and *jvm.options* and *server.env* files at URL <https://www.ibm.com/docs/en/was-liberty/zos?topic=liberty-embedding-server-in-your-applications> regarding restrictions in this environment.

Tech-Tip: The z/OS Connect `zconsetup` script must be invoked once per LPAR per install path

The `zconsetup` script creates a symbolic link from the z/OS Connect's WLP `..v3r0/wlp/etc` directory in the product path (normally R/O) to a local R/W directory (creating a default configuration and local extension directories). This provides a means for administrator to configure the image.

```
JOHNSON:/usr/lpp/IBM/zosconnect/v3r0/wlp/etc: ls -al
total 32
drwxrwxr-x  2 OMVSKERN 0          8192 Jun 24 10:24 .
drwxrwxr-x 10 OMVSKERN 0          8192 Jun 24 10:24 ..
lrwxrwxrwx  1 990023 0          31 Jul 27 2020 extensions -> /var/zosconnect/v3r0/extensions
```

```
/var/zosconnect
  /servers
  /v3r0
    /extensions
      imsmobile.properties
      zosconnect.properties
      filemanager.properties
      omegamon.properties
```



```
com.ibm.websphere.productId=com.ibm.ims.mobile
com.ibm.websphere.productInstall=imsmobile/wlp-ext relative path

com.ibm.websphere.productId=com.ibm.zosconnect
com.ibm.websphere.productInstall=runtime

com.ibm.websphere.productId=omegamonRequestMonitor-2.0
com.ibm.websphere.productInstall=/var/zosconnect/omegamon absolute path
```

- This directory structure and contents is created by invoking the `zconsetup` script and **must be created on each LPAR** on which z/OS Connect will execute. This is how the z/OS Connect Liberty server locates service provider executables. Note: the `com.ibm.websphere.productInstall` directive value that is **relative** to directory `/usr/lpp/IBM/zosconnect/v3r0`.
- Not creating this link will cause messages `CWWKF0001E: A feature definition could not be found for zosconnect:....` or `CWWKE0054E: Unable to open /usr/lpp/IBM/zosconnect/v3r0/wlp/etc/extensions/zosconnect.properties`

Tech-Tip: Verify the `zconsetup` script has been executed. My recommendation is to execute this script in the SMP/E target environment, otherwise it will be lost when service is applied and propagated to other images.

Tech Tip: Use multiple mount points and dedicated ZFS file systems

- Create mount points in the “administrative” directory for shared r/w directories
- Avoid creating directories and files in the root file system.
- Use a common or shared mount point
 - Use /var mount point for local read/write file systems
 - Use /global for sharing a mount point across multiple LPARs
- Use ZFS filesystems and use AGGRGROW to allow R/W ZFS filesystems to automatically go into extents (>16).

SYS1.PARMLIB (BPXPRM##)

```

MOUNT FILESYSTEM('OMVS LIBERTY ZFS')
  MOUNTPOINT('/var/liberty')
  TYPE(ZFS) MODE(READ)
MOUNT FILESYSTEM('OMVS LIBERTY SERVERS ZFS')
  MOUNTPOINT('/var/liberty/servers')
  TYPE(ZFS) PARM('AGGRGROW') MODE(RDWR)
MOUNT FILESYSTEM('OMVS LIBERTY GROUP1 ZFS')
  MOUNTPOINT('/var/zosconnect/group1')
  TYPE(ZFS) PARM('AGGRGROW') MODE(RDWR)
MOUNT FILESYSTEM('OMVS LIBERTY GROUP2 ZFS')
  MOUNTPOINT('/var/liberty/group2')
  TYPE(ZFS) PARM('AGGRGROW') MODE(RDWR)
MOUNT FILESYSTEM('OMVS LIBERTY GROUP3 ZFS')
  MOUNTPOINT('/var/liberty/group3')
  TYPE(ZFS) PARM('AGGRGROW') MODE(RDWR)

```

- Create a dedicated filesystem for the root z/OS Connect /var directory, e.g., /var/liberty/v3r0/extensions. This directory structure can not be changed. This provides portability for migrations and system upgrades. Note: MODE(READ) will apply to /var/liberty/servers.

- Create a dedicated filesystem for each set or groups of servers. These filesystems will contain the server configuration directories for 1 or more servers.
- Each server's WLP_USER_DIR environment variable will be set to the mount point, e.g., *WLP_USER_DIR=/var/liberty/group1* when the server is created and in the server's startup JCL.

df -P | grep /var/liberty

Filesystem	512-blocks	Used	Available	Capacity	Mounted on
OMVS.LibVar.ZFS	69120	68658	462	100%	/var/liberty
OMVS.Liberty.Servers.ZFS	159120	76455	82665	48%	/var/liberty/servers
OMVS.Liberty.Group1.ZFS	135360	1506	133854	2%	/var/liberty/group1
OMVS.Liberty.Group2.ZFS	4059360	2591284	1468076	64%	/var/liberty/group2
OMVS.Liberty.Group3.ZFS	135360	17858	117502	14%	/var/liberty/group3

Tech-Tip: Consider using symbolic links as an administrative shortcut

- Create an “administration” subdirectory, e.g., *LibertyConfigs* in directory */var*
- Then create a symbolic link in the “administration” directory to each Liberty server’s configuration directory and other frequently accessed directories.

```
ls -al /var/LibertyConfigs
drwxrwxrwx  4 JOHNSON  SYS1          8192 Aug 16 12:23 .
drwxrwxrwt 25 OMVSKERN SYS1          8192 Aug 16 11:56 ..
lrwxrwxrwx  1 JOHNSON  SYS1          57 Aug 16 12:22 CSCWLP -> /var/wlp/cics/CICS53Z/CSCWLP/wlp/usr/servers/defaultServer
lrwxrwxrwx  1 JOHNSON  SYS1          57 Aug 16 12:22 CICSWLP -> /var/wlp/cics/CICS53Z/CICSWLP/wlp/usr/servers/cicswlp
drwxrwxrwx  2 JOHNSON  SYS1          8192 Aug 16 15:30 hcd
lrwxrwxrwx  1 JOHNSON  SYS1          27 Jun 10 15:55 includes -> /global/zosconnect/includes
lrwxrwxrwx  1 JOHNSON  SYS1          28 Aug 16 10:12 mqweb -> /var/mqm/mqweb/servers/mqweb
lrwxrwxrwx  1 JOHNSON  SYS1          32 Jun  4 12:49 myServer -> /var/zosconnect/servers/myServer
drwxr-xr-x  2 JOHNSON  SYS1          8192 Aug 16 13:14 properties
lrwxrwxrwx  1 JOHNSON  SYS1          18 Aug 17 12:47 shared -> /var/shared/zosconnect/resources/zosconnect
lrwxrwxrwx  1 JOHNSON  SYS1          24 May 13 2020 walop3a -> /var/wlp/servers/walop3a
lrwxrwxrwx  1 JOHNSON  SYS1          24 May 13 2020 walrp3a -> /var/wlp/servers/walrp3a
lrwxrwxrwx  1 JOHNSON  SYS1          31 May 14 2020 wazs34a -> /var/zosconnect/servers/wazs34a
lrwxrwxrwx  1 JOHNSON  SYS1          24 Aug 16 10:32 wlphats -> /var/wlp/servers/wlphats
lrwxrwxrwx  1 JOHNSON  SYS1          36 Aug 16 10:31 zceearpir -> /var/ats/zosconnect/servers/zceearpir
lrwxrwxrwx  1 JOHNSON  SYS1          39 Aug 16 10:18 zceecics -> /var/cicsts/zosconnect/servers/zceecics
lrwxrwxrwx  1 JOHNSON  SYS1          35 Aug 16 10:31 zceedvm -> /var/ats/zosconnect/servers/zceedvm
lrwxrwxrwx  1 JOHNSON  SYS1          32 Jun 10 15:54 zceepid -> /var/zosconnect/servers/zceepid
lrwxrwxrwx  1 JOHNSON  SYS1          36 Aug 16 10:14 zceesrvr -> /var/ats/zosconnect/servers/zceesrvr
lrwxrwxrwx  1 JOHNSON  SYS1          44 Aug 16 11:57 zosmfServer -> /var/zosmf/configuration/servers/zosmfServer
```

There are various Liberty servers shown here. There are CICS Liberty servers, z/OS Connect servers, a MQ Web Console Liberty server, a zOSMF Liberty server, a HATS Liberty server and a few standard Liberty servers for Java applications.

One administration directory to manage them all!



Tech-Tip: Combine these techniques to standardized the creation and customization of new servers

```
//*****  
///* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
// SET ZCEEPATH='/usr/lpp/IBM/zosconnect/v3r0'  
// SET SERVER='zceesrvr'  
// SET TEMPLATE='zosconnect:default'  
// SET WLPUSER='/var/ats/zosconnect'  
// SET USER='ATSSERV'  
// SET GROUP='ATSGRP'  
//*****  
///* Step ZCEESRVR - Use the zosconnect command to create a server  
//*****  
//ZCEESRVR EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
export JAVA_HOME=&JAVAHOME; +  
export WLP_USER_DIR=&WLPUSER; +  
&ZCEEPATH/bin/zosconnect create &SERVER +  
--template=&TEMPLATE; +  
ln -s $WLP_USER_DIR/servers/&SERVER /var/liberty/&SERVER; +  
cp /var/liberty/properties/bootstrap.properties +  
/var/liberty/&SERVER; +  
cp /var/liberty/properties/server.xml +  
/var/liberty/&SERVER; +  
chown -R &USER:&GROUP $WLP_USER_DIR/servers/&SERVER
```

- Set and export environment variables
- Create the server and its configuration directories, etc.
- Create a symbolic link from the server's configuration directory to common administration directory
- Copy a default bootstrap.properties and server.xml files into the server's configuration directory
- Change owner and group of the server's configuration directories and files to the SAF identity under which the server will run



Finally, be aware of the defaults for update polling (updateTrigger=mbean)

```
<!-- Application Monitoring      -->
<applicationMonitor updateTrigger="polled" dropinsEnabled="true"/>

<!-- config requires updateTrigger="mbean" for REFRESH command support -->
<config updateTrigger="polled" monitorInterval="500ms"/>

<!-- zosConnect APIs -->
<zosconnect_zosConnectAPIs pollingRate="5s" updateTrigger="disabled" />

<!-- zosConnect API requesters -->
<zosconnect_apiRequesters updateTrigger="disabled" pollingRate="5s"/>

<!-- zosConnect Services -->
<zosconnect_services pollingRate="5s" updateTrigger="disabled"/>

<!-- zosConnect policies -->
<zosconnect_policy pollingRate="1m" updateTrigger="disabled"/>

<!-- zosConnect data transformer -->
<zosconnect_zosConnectDataXform pollingrate="2s" updateTrigger="polled"/>

<!--A security certificate repository -->
<keystore pollingrate="500ms" updateTrigger="mbean"/>
```

For our purposes, a Mbean (managed bean) is a Java object that can perform a function on a resource or configuration update using MVS modify commands.

mitchj@us.ibm.com

© 2017, 2025 IBM Corporation
Slide 45

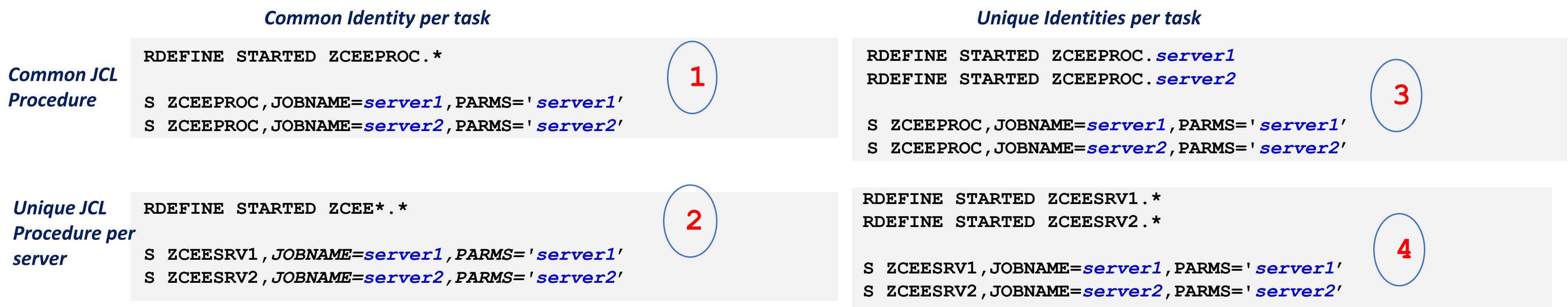
Liberty and z/OS Security

Overview

z/OS Security: Assigning ID to started tasks: SAF STARTED class

The first question here is whether you wish to have a common started task ID that is shared among servers, or if you wish each server to have a unique ID

Then the second question is whether servers under a WLP_USER_DIR will share a common JCL start proc, or use unique start procs for each server



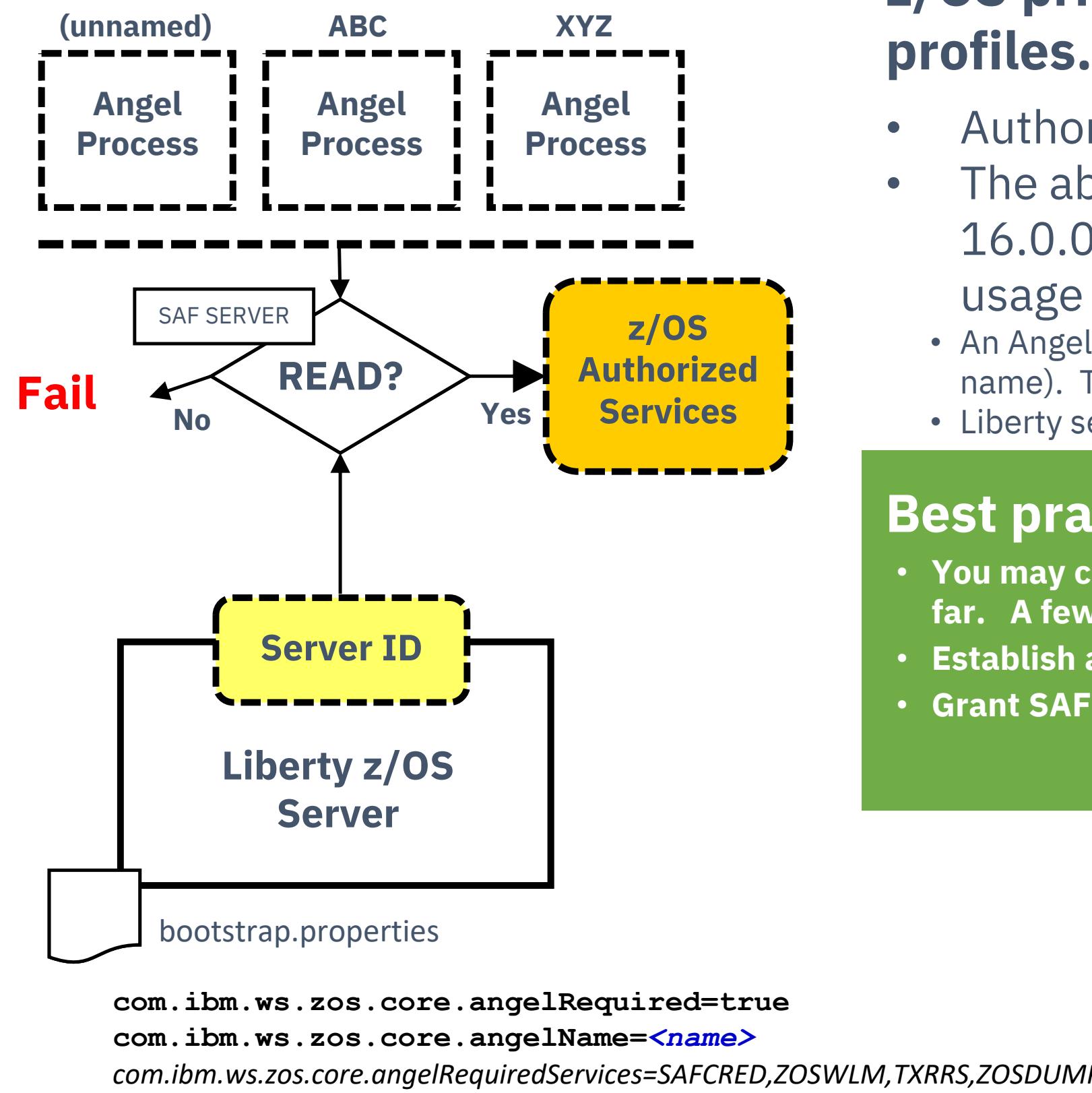
Note: Using unique JCL procedure eliminates the need to specify PARMS on the start commands

1. The same identity is used for all servers using a RACF STARTED class resource where the JCL procedure is discrete, and the job name is generic.
2. The same identity is used for all servers using a RACF STARTED class resource where both the JCL procedure and job names are generic
3. Different identities are used for each server using a RACF STARTED class resource where both the JCL procedure and job name are discrete.
4. Different identities are used for each server using a RACF STARTED class resource where the JCL procedure is discrete and job name is generic.

It's possible to use a combination of the above, even under the same WLP_USER_DIR. So there's no "one best answer" here. What's best is what's best for you.



z/OS Security: The Angel process – what is this about?



The Angel Process is a started task that is used to protect access to z/OS privileged or authorized services. This is done with SAF SERVER profiles.

- Authorized services include: WOLA, SAF, WLM, RRS, DUMP
- The ability to start multiple Angel processes on an LPAR was introduced in 16.0.0.4. This is called "Named Angels". It provides a way to separate Angel usage between Liberty servers:
- An Angel process can be started with a NAME='<name>' parameter (or it can be started as a "default" without a name). The name may be up to 54 characters.
- Liberty servers can be pointed at a specific Angel with a bootstrap property

Best practice:

- You may create separate named Angels for isolation of Test and Production, but do far. A few Angels, yes; dozens, no.
- Establish automation routines to start the Angels at IPL
- Grant SAF GROUP access to the SERVER profiles, then connect server IDs as needed

List of current Liberty Features

https://www.ibm.com/support/knowledgecenter/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/rwlp_feat.html

z/OS Security: SAF SERVER profiles related to the Angel



Best practice:

- Establish all the SERVER profiles ahead of time. Existence of profile does not grant access; READ access does.
- Determine what access a server needs and grant only that; check "is available" messages in messages.log to verify

Tech/Tip: The SAFLOG parameter was added in a recent Liberty service. If this parameter is set to Y, additional security related messages will be written to the JES messages and console if a Liberty server does not have authorization to use an angel-controlled privileged function. See URL

https://www.ibm.com/support/knowledgecenter/SS7K4U_liberty/com.ibm.websphere.wlp.zseries.doc/ae/rwlp_newinrelease.html

Liberty 21.0.6 add a new property to identify required services, com.ibm.ws.zos.core.angelRequiredServices, for more details see URL

<https://www.ibm.com/docs/en/was-liberty/zos?topic=overview-process-types-zos>

z/OS Authorized service security: Angel Required Services

To use z/OS authorized services, you must have a Liberty Angel process and grant access for your Liberty server's SAF identity to use these services.

- LOCALCOM - Required to use *WebSphere Optimized Local Adapters* (WOLA).
- PRODMGR - Required to use IFAUSAGE services for SMF reporting.
- SAFCRED - Required to use SAF authorized user registry services and SAF authorization services.
- TXRRS - Required by the IBM® MQ resource adapter when the connection to IBM MQ is made in BINDINGS mode
- WOLA - Required to use *WebSphere Optimized Local Adapters* (WOLA).
- ZOSAIO - Required to use AsyncIO on z/OS. **It is a good practice to enable this service**
- ZOSDUMP - Only required if asked to obtain an SVC dump by IBM service. It provides access to SVCDUMP services.
- ZOSWLM - Required to use WLM services. For more information, see [Measuring API workloads with WLM](https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=considerations-measuring-api-workloads-wlm) at URL <https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=considerations-measuring-api-workloads-wlm>

- When a Liberty server connects to an angel process during server startup, it checks that the server's identity has access to the z/OS authorized services. By default, access checks are performed for all authorized services.
- You can restrict the Liberty server to check and use only the authorized services it requires, which then makes other authorized services unavailable by using property, **com.ibm.ws.zos.core.angelRequiredServices**
- The value for this property, **com.ibm.ws.zos.core.angelRequiredServices**, must be a comma-separated list of valid angel process services, as described above. This property must be specified with the **com.ibm.ws.zos.core.angelRequired** property set to **true**. Only these services, when properly specified, are the ones used by the server.

Lack of access to the angel process itself or any of these listed required services will cause a server startup failure.



Tech-Tip: SAF APPL and EJBRole Resources

Connect z/OS Connect users to a common group

CONNECT (FRED,USER1,JOHNSON) GROUP(ZCEEUSR)

Define a APPL profile for the server's SAF profilePrefix and permits access to the Liberty server

RDEFINE APPL BBGZDFLT UACC(NONE) OWNER(SYS1)

**PERMIT BBGZDFLT CLASS(APPL) ACCESS(READ) ID(WSGUEST#, ZCEEUSR)
SETROPTS RACLIST(APPL) REFRESH**

Example of defining an EJBROLE profile for the server's SAF profilePrefix and permits access to Liberty APIs

RDEFINE EJBROLE BBGZDFLT.zos.connect.access.roles.zosConnectAccess OWNER(SYS1) UACC(NONE)

**PERMIT BBGZDFLT.zos.connect.access.roles.zosConnectAccess +
CLASS(EJBROLE) ID(ZCEEUSR) ACCESS(READ)
SETROPTS RACLIST(EJBROLE) REFRESH**

```
<safCredentials unauthenticatedUser="WSGUEST" profilePrefix="BBGZDFLT"/>  
<safAuthorization racRouteLog=ASIS reportAuthorizationCheckDetails="true"/>
```

- # https://www.ibm.com/support/knowledgecenter/SS7K4U_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp_config_security_saf.html
<https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=registry-saf-unauthenticated-user-id>

Example: Using the *bootstrap.properties* file to specify the required z/OS privileges



zceesrv1's bootstrap.properties

```
httpPort=9080
httpsPort=9443
ipicPort=1491
host=*
cicsHost=wg31.washington.ibm.com
network=ZOSCONN1
applid=ZOSCONN1
com.ibm.ws.zos.core.angelName=namedAngel
com.ibm.ws.zos.core.angelRequired=true
com.ibm.ws.zos.core.angelRequiredServices=SAFCRED,ZOSWLM,PRODMGR,ZOSAIO,TXRRS,LOCALCOM
```

zceesrv2's bootstrap.properties

```
httpPort=9090
httpsPort=9453
ipicPort=1492
host=wg31.washington.ibm.com
cicsHost=wg31.washington.ibm.com
network=ZOSCONN2
applid=ZOSCONN2
com.ibm.ws.zos.core.angelName=namedAngel
com.ibm.ws.zos.core.angelRequired=true
com.ibm.ws.zos.core.angelRequiredServices=SAFCRED,ZOSWLM,PRODMGR,ZOSAIO,TXRRS,LOCALCOM
```

Useful Liberty functions/features and MVS commands



Liberty feature: restConnector-2.0

A secure, REST administrative connector that provides real time access to configuration settings from remote access from a Java client or Web browser (GET only) or directly through an HTTPS call to the current runtime configuration.

The screenshot shows the 'Server Config' interface with the title 'restConnector.xml'. It has tabs for 'Design' and 'Source'. The 'Source' tab is selected, displaying XML code for a server named 'REST Connector' that includes a 'featureManager' section with a single feature 'restConnector-2.0'. To the right of the code, a list of URLs is shown, each starting with 'https://wg31.washington.ibm.com:9443/ibm/api/config/' followed by a specific endpoint name.

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="REST Connector">
  <featureManager>
    <feature>restConnector-2.0</feature>
  </featureManager>
</server>
```

URI Path is the concatenation of the path /ibm/api/config with the server XML configuration element and any optional query strings.

- <https://wg31.washington.ibm.com:9443/ibm/api/config/jmsQueue>
- https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_cicsIpicConnection?port=1491
- https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectServiceRestClientConnection
- https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_cicsIpicConnection?id=miniloan
- <https://wg31.washington.ibm.com:9443/ibm/api/config/safCredentials>
- <https://wg31.washington.ibm.com:9443/ibm/api/config/connectionFactory>
- https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectManager
- <https://wg31.washington.ibm.com:9443/ibm/api/config/keyStore>
- <https://wg31.washington.ibm.com:9443/ibm/api/config/ssl>
- <https://wg31.washington.ibm.com:9443/ibm/api/config/sslDefault>
- https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectManager
- https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectAPIs
- https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_services
- https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_apiRequesters

```
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.Administrator OWNER(SYS1) UACC(NONE)
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.Reader OWNER(SYS1) UACC(NONE)
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers OWNER(SYS1) UACC(NONE)
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.Administrator CLASS(EJBROLE) ID(ZCEEUSRS) ACCESS(READ)
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.Reader CLASS(EJBROLE) ID(ZCEEUSRS) ACCESS(READ)
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers CLASS(EJBROLE) ID(ZCEEUSRS) ACCESS(READ)
SETR RACLIST(EJBROLE) REFRESH
```



Liberty feature: restConnector-2.0 – featureManager example

https://wg31.washington.ibm.com:9443/ibm/api/config/featureManager

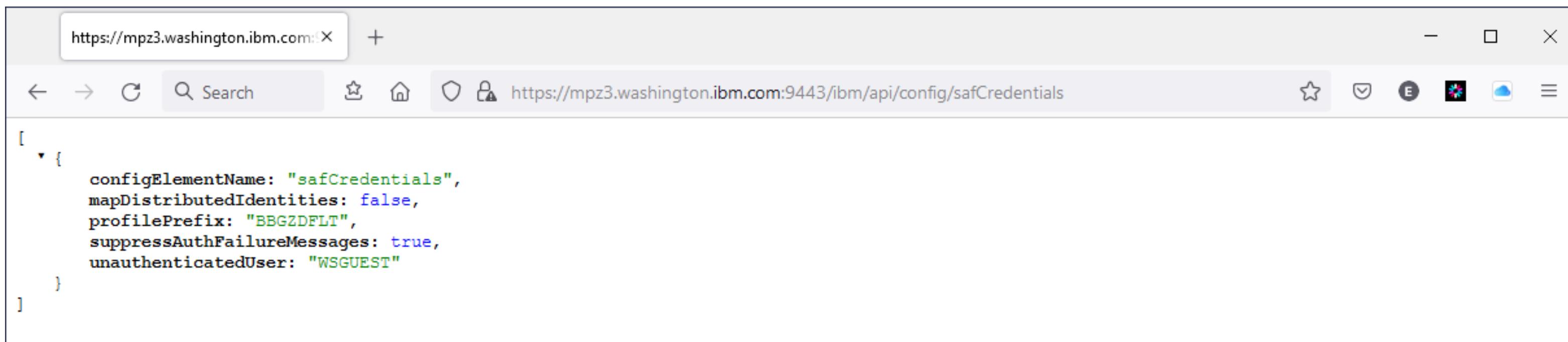
The screenshot shows a browser window with the URL <https://wg31.washington.ibm.com:9443/ibm/api/config/featureManager>. The page displays a JSON object representing the configuration for the featureManager. The JSON structure is as follows:

```
[{"configElementName": "featureManager", "feature": ["appSecurity-2.0", "zosSecurity-1.0", "zosconnect:cicsService-1.0", "transportSecurity-1.0", "zosconnect:apiRequester-1.0", "zosconnect:apiRequester-1.0", "zosconnect:mqService-1.0", "zosWlm-1.0", "restConnector-2.0", "monitor-1.0", "zosRequestLogging-1.0", "adminCenter-1.0", "apiDiscovery-1.0", "zosconnect:zosConnect-2.0", "zosconnect:zosConnectCommands-1.0", "imsmobile:imsmobile-2.0"], "onError": "WARN"}]
```

Liberty feature: restConnector-2.0 – safCredentials/safAuthorization examples



https://wg31.washington.ibm.com:9443/ibm/api/config/safCredentials



```
[{"configElementName": "safCredentials", "mapDistributedIdentities": false, "profilePrefix": "BBGZDFLT", "suppressAuthFailureMessages": true, "unauthenticatedUser": "WSGUEST"}]
```

https://wg31.washington.ibm.com:9443/ibm/api/config/safAuthorization



```
[{"configElementName": "safAuthorization", "uid": "safAuthorization[default-0]", "enableDelegation": false, "racRouteLog": "ASIS", "reportAuthorizationCheckDetails": false, "roleMapper": "com.ibm.ws.security.authorization.saf.internal.SAFRoleMapperImpl"}]
```

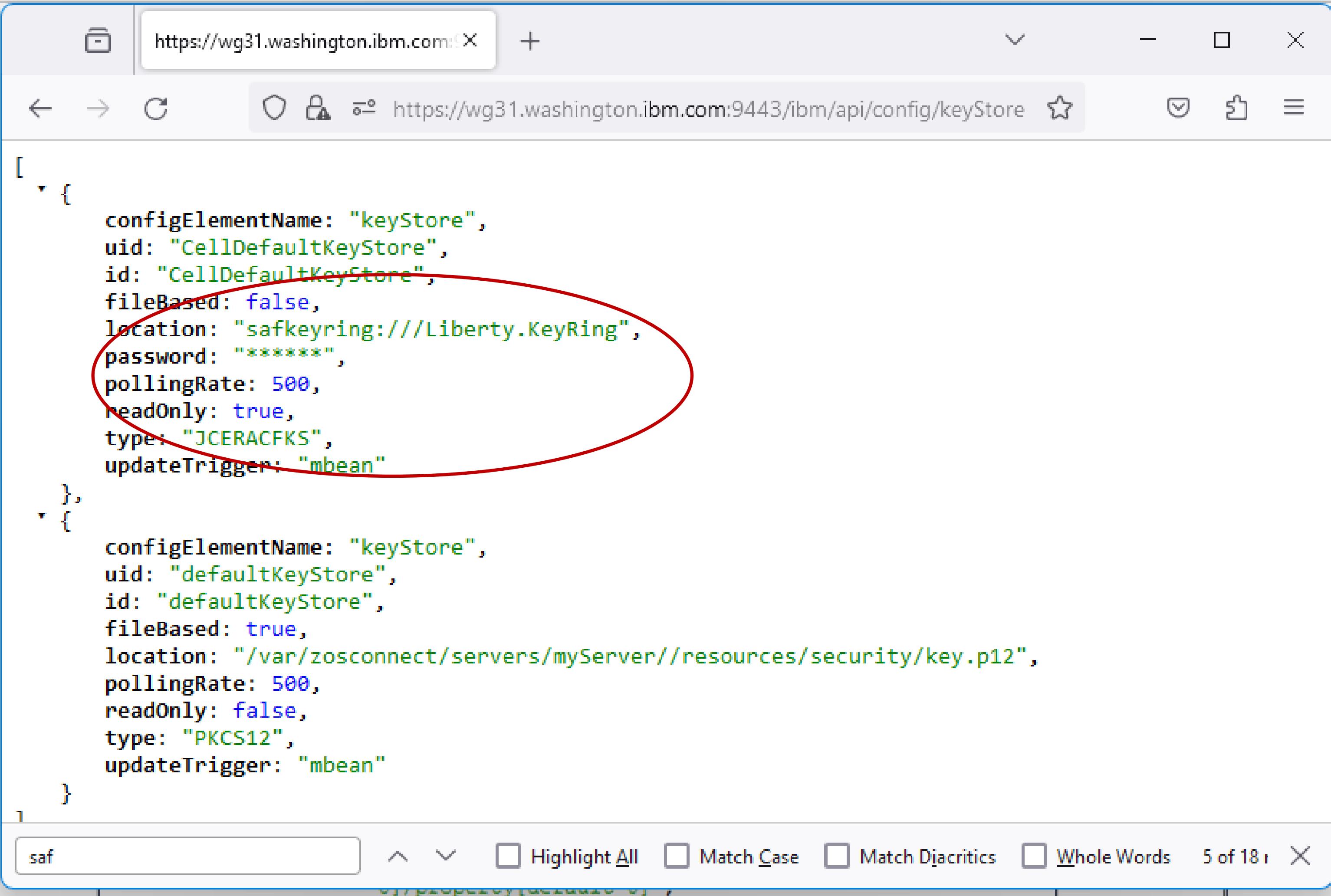
saf

Highlight All Match Case Match Diacritics Whole Words 5 of 18 matches



Liberty feature: restConnector-2.0 – keystore example

https://wg31.washington.ibm.com:9443/ibm/api/config/keyStore

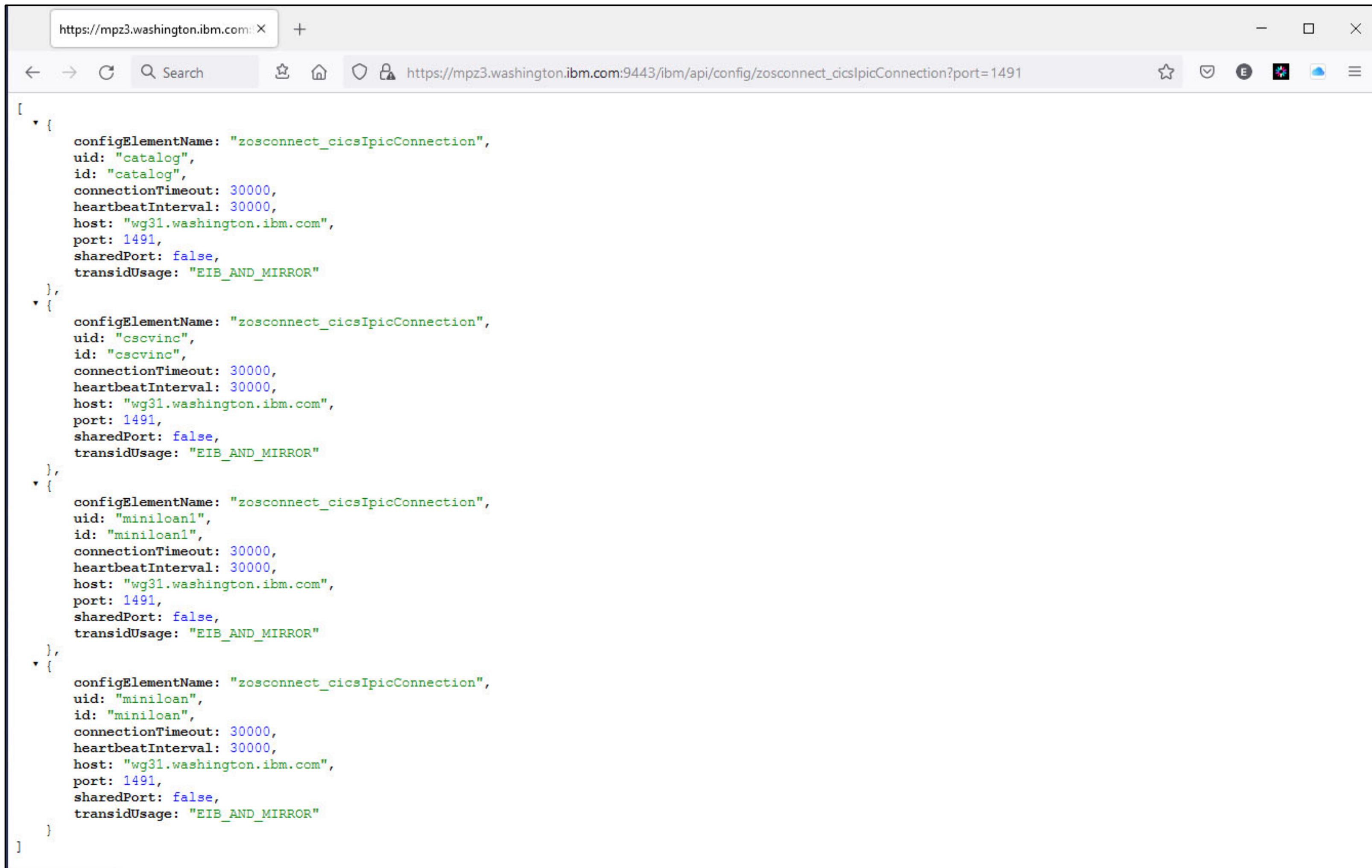


```
[  
  {  
    configElementName: "keyStore",  
    uid: "CellDefaultKeyStore",  
    id: "CellDefaultKeyStore",  
    fileBased: false,  
    location: "safkeyring:///Liberty.KeyRing",  
    password: "*****",  
    pollingRate: 500,  
    readOnly: true,  
    type: "JCERACFKS",  
    updateTrigger: "mbean"  
  },  
  {  
    configElementName: "keyStore",  
    uid: "defaultKeyStore",  
    id: "defaultKeyStore",  
    fileBased: true,  
    location: "/var/zosconnect/servers/myServer//resources/security/key.p12",  
    pollingRate: 500,  
    readOnly: false,  
    type: "PKCS12",  
    updateTrigger: "mbean"  
  }  
]
```



Liberty feature: restConnector-2.0 – zosconnect_cicsIpicConnection example

https://wg31.washington.ibm.com:9443/ibm/api/config/zosconnect_cicsIpicConnection?port=1491



```
[{"configElementName": "zosconnect_cicsIpicConnection", "uid": "catalog", "id": "catalog", "connectionTimeout": 30000, "heartbeatInterval": 30000, "host": "wg31.washington.ibm.com", "port": 1491, "sharedPort": false, "transidUsage": "EIB_AND_MIRROR"}, {"configElementName": "zosconnect_cicsIpicConnection", "uid": "cscvinc", "id": "cscvinc", "connectionTimeout": 30000, "heartbeatInterval": 30000, "host": "wg31.washington.ibm.com", "port": 1491, "sharedPort": false, "transidUsage": "EIB_AND_MIRROR"}, {"configElementName": "zosconnect_cicsIpicConnection", "uid": "minilcan1", "id": "minilcan1", "connectionTimeout": 30000, "heartbeatInterval": 30000, "host": "wg31.washington.ibm.com", "port": 1491, "sharedPort": false, "transidUsage": "EIB_AND_MIRROR"}, {"configElementName": "zosconnect_cicsIpicConnection", "uid": "minilcan", "id": "minilcan", "connectionTimeout": 30000, "heartbeatInterval": 30000, "host": "wg31.washington.ibm.com", "port": 1491, "sharedPort": false, "transidUsage": "EIB_AND_MIRROR"}]
```

Liberty feature: restConnector-2.0 – CICS ECI connection factory example



<https://wg31.washington.ibm.com:9443/ibm/api/config/connectionFactory>

The screenshot shows a web browser window displaying a JSON configuration object. The URL in the address bar is <https://wg31.washington.ibm.com:9443/ibm/api/config/connectionFactory>. The JSON content is as follows:

```
[{"configElementName": "connectionFactory", "uid": "ECI", "id": "ECI", "jndiName": "ECI", "properties.eciResourceAdapter": {"TPNName": "", "applid": "", "applidQualifier": "", "cipherSuites": "", "clientSecurity": "", "connectionURL": "tcp://wg31.washington.ibm.com", "keyRingClass": "", "keyRingPassword": "*****", "password": "*****", "portNumber": "2006", "requestExits": "", "serverName": "CICS62", "serverSecurity": "", "socketConnectTimeout": "0", "traceLevel": 3, "tranName": "", "userName": ""}, "api": ["/ibm/api/validation/connectionFactory/ECI"]}]
```

A red rectangular box highlights the "properties.eciResourceAdapter" section of the JSON, which contains the connection parameters for the CICS ECI resource adapter.

Liberty feature: restConnector-2.0 – IMS connection factory example

https://wg31.washington.ibm.com:9453/ibm/api/config/imsmobile_imsConnection

The image shows two side-by-side browser windows displaying JSON configuration data for IMS connection factories.

Left Browser Window: The URL is https://wg31.washington.ibm.com:9453/ibm/api/config/imsmobile_imsConnection. The JSON content describes an 'imsmobile_imsConnection' configuration element. A red box highlights the 'properties.gmoa' section, which includes fields like 'CMODedicated', 'IMSConnectName', 'SSLEnabled', and 'SSLProtocol'. The 'api' section lists the endpoint `/ibm/api/validation/connectionFactory/IVP1`.

```
[{"configElementName": "imsmobile_imsConnection", "uid": "IMSCONN", "id": "IMSCONN", "comment": "", "connectionFactoryRef": {"configElementName": "connectionFactory", "uid": "IVP1", "id": "IVP1", "containerAuthDataRef": {"configElementName": "authData", "uid": "Connection1_Auth", "id": "Connection1_Auth", "password": "*****", "user": "USER1"}, "properties.gmoa": {"CMODedicated": false, "IMSConnectName": "", "SSLEnabled": false, "SSLProtocol": "TLSv1.0", "SSLTrustStoreName": "", "SSLTrustStorePassword": "*****", "applicationName": "", "dataStoreName": "myDStrNm", "groupName": "", "hostName": "wg31.washington.ibm.com", "password": "*****", "passwordPhrase": "*****", "portNumber": 4000, "traceLevel": 1, "userName": ""}, "api": ["/ibm/api/validation/connectionFactory/IVP1"]}, "connectionTimeout": "-1", "connectionType": "IMSCONNECT", "pingIMSCollectionInvoke": false}]
```

Right Browser Window: The URL is <https://wg31.washington.ibm.com:9453/>. The JSON content describes an 'imsmobile_interaction' configuration element. A search bar at the bottom contains the text 'imsinter'. The configuration includes fields like 'ackNakProvider', 'commitMode', 'configSchemaVersion', 'imsConnectCodepage', 'imsConnectTimeout', and 'imsConnectUserMessageExitIdentifier'.

```
[{"configElementName": "imsmobile_interaction", "uid": "IMSINTER", "id": "IMSINTER", "ackNakProvider": 0, "comment": "", "commitMode": 1, "configSchemaVersion": 1, "imsConnectCodepage": "Cp1047", "imsConnectTimeout": 30000, "imsConnectUserMessageExitIdentifier": "*SAMPL1*", "imsDatastoreName": "IVP1", "inputMessageDataSegmentsIncludeLlzzAndTrancode": true, "interactionTimeout": -1, "interactionTypeDescription": "SENDRECV", "ltermOverrideName": "", "outputMessageDataSegmentsIncludeLlzz": true, "propagateNetworkSecurityCred": true, "propertyType": "TRAN", "purgeUndeliverableOutput": true, "rerouteUndeliverableOutput": false, "resumeTpipeProcessing": 16, "returnMfsModname": true, "syncLevel": 0, "useCM0AckNoWait": true}]
```

Liberty feature: restConnector-2.0 – IMS connection factory example



https://wg31.washington.ibm.com:9453/ibm/api/config/zosconnect_imsConnection

The screenshot shows a browser window with the URL https://wg31.washington.ibm.com:9445/ibm/api/config/zosconnect_imsConnection. The page displays a JSON object representing the configuration of an IMS connection factory. A red rectangular box highlights the 'properties.gmoa' section, which contains sensitive information like user names and passwords. The JSON structure includes fields for connection factories, authentication data, properties, and APIs.

```
[{"configElementName": "zosconnect_imsConnection", "uid": "imsConn", "id": "imsConn", "commitMode": 1, "connectionFactoryRef": {"configElementName": "connectionFactory", "uid": "imsConnectionFactory", "id": "imsConnectionFactory", "containerAuthDataRef": {"configElementName": "authData", "uid": "IMSCredentials", "id": "IMSCredentials", "password": "*****", "user": "${IMS_USER}"}, "properties.gmoa": {"CMODedicated": false, "IMSConnectName": "", "SSLEnabled": false, "SSLEncryptionType": "Weak", "SSLKeyStoreName": "", "SSLKeyStorePassword": "*****", "SSLProtocol": "TLSv1.0", "SSLTrustStoreName": "", "SSLTrustStorePassword": "*****", "applicationName": "", "dataStoreName": "myDStrNm", "groupName": "", "hostName": "wg31.washington.ibm.com", "password": "*****", "passwordPhrase": "*****", "portNumber": 4000, "traceLevel": 1, "userName": ""}, "api": ["/ibm/api/validation/connectionFactory/imsConnectionFactory"]}, "imsConnectCodepage": "Cp1047", "imsConnectTimeout": 0, "imsDatastoreName": "IVP1", "interactionTimeout": 0, "pingIMSConnectOnInvoke": false, "propagateNetworkSecurityCred": true, "syncLevel": 0, "tranExpiration": false}]
```

mitchj@us.ibm.com

© 2017, 2025 IBM Corporation
Slide 61

Liberty feature: restConnector-2.0 – jmsConnectionFactory example



<https://wg31.washington.ibm.com:9453/ibm/api/config/jmsConnectionFactory>

```
[{"configElementName": "jmsConnectionFactory", "uid": "qmgrCf", "id": "qmgrCf", "jndiName": "jms/qmgrCf", "connectionManagerRef": {"configElementName": "connectionManager", "uid": "ConMgr1", "id": "ConMgr1", "agedTimeout": -1, "autoCloseConnections": true, "connectionTimeout": 30, "enableContainerAuthForDirectLookups": false, "enableSharingForDirectLookups": true, "maxIdleTime": 1800, "maxInUseTime": -1, "maxPoolSize": 5, "purgePolicy": "EntirePool", "reapTime": 180}, "properties.wmqJms": {"CCSID": 819, "cleanupLevel": "SAFE", "cloneSupport": "DISABLED", "failIfQuiesce": true, "headerCompression": "NONE", "messageCompression": "NONE", "messageSelection": "CLIENT", "port": 1414, "providerVersion": "unspecified", "pubAckInterval": 25, "queueManager": "QM1", "rescanInterval": 5000, "shareConvAllowed": true, "sparseSubscriptions": false, "sslResetCount": 0, "statusRefreshInterval": 60000, "subscriptionStore": "BROKER", "targetClientMatching": true, "transportType": "BINDINGS", "wildcardFormat": "TOPIC"}, "api": ["/ibm/api/validation/jmsConnectionFactory/qmgrCf"]}]
```

wmqj

Highlight All Match Case Match Diacritics Whole Words 1 of 6 matches



Liberty feature: **apiDiscovery-1.0** or **OpenAPI-3.0** to display/execute APIs

The screenshot shows a browser window titled "IBM REST API Documentation" with the URL <https://mpz3.washington.ibm.com:9443/api/explorer/#/cscvinc>. The page displays a list of REST APIs under the heading "Liberty REST APIs".

cscvinc

- POST** /cscvinc/employee
- DELETE** /cscvinc/employee/{employee}
- GET** /cscvinc/employee/{employee}
- PUT** /cscvinc/employee/{employee}

db2employee

filemgr

imsPhoneBook

jwtlvpDemoApi

miniloancics

mqapi

phonebook

Each API entry includes "Show/Hide", "List Operations", and "Expand Operations" buttons.



Use the **apiDiscovery-1.0** or **OpenAPI-3.0** features to execute RESTful APIs directly

The screenshot shows a browser window titled "IBM REST API Documentation" with the URL <https://mpz3.washington.ibm.com:9443/api/explorer/#/cscvinc>. The page displays the "Liberty REST APIs" section, specifically for the "cscvinc" service. It lists several API operations:

- cscvinc**:
 - POST /cscvinc/employee
 - DELETE /cscvinc/employee/{employee}
 - GET /cscvinc/employee/{employee}
 - PUT /cscvinc/employee/{employee}
- db2employee**:
 - Show/Hide | List Operations | Expand Operations
- filemgr**:
 - Show/Hide | List Operations | Expand Operations
- imsPhoneBook**:
 - Show/Hide | List Operations | Expand Operations
- jwtlvpDemoApi**:
 - Show/Hide | List Operations | Expand Operations
- miniloancics**:
 - Show/Hide | List Operations | Expand Operations
- mqapi**:
 - Show/Hide | List Operations | Expand Operations
- phonebook**:
 - Show/Hide | List Operations | Expand Operations

mitchj@us.ibm.com



IBM MQ Administrative REST API

qmgr		Show/Hide List Operations Expand Operations
GET	/ibmmq/rest/v1/admin/qmgr	Retrieves details of all queue managers in the IBM MQ installation.
GET	/ibmmq/rest/v1/admin/qmgr/{qmgr}	Retrieves details of a specific queue manager in the IBM MQ installation.
*	GET /ibmmq/rest/v2/admin/qmgr	Retrieves details of all queue managers in the IBM MQ installation.
*	GET /ibmmq/rest/v2/admin/qmgr/{qmgr}	Retrieves details of a specific queue manager in the IBM MQ installation.
qmgr : action		Show/Hide List Operations Expand Operations
POST	/ibmmq/rest/v1/admin/action/qmgr/{qmgrName}/mqsc	Runs an MQSC command.
*	POST /ibmmq/rest/v2/admin/action/qmgr/{qmgrName}/mqsc	Runs an MQSC command.
queue		Show/Hide List Operations Expand Operations
GET	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue	Retrieves details of all queues.
POST	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue	Creates a queue.
DELETE	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue/{qName}	Deletes a queue.
GET	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue/{qName}	Retrieves details of a specific queue.
PATCH	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue/{qName}	Modifies a queue.
subscription		Show/Hide List Operations Expand Operations
GET	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/subscription	Retrieves details of all subscriptions.
GET	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/subscription/{name}	Retrieves details of a specific subscription.

* If you are accessing a version earlier than V9.1.5 you must use v1

IBM MQ Messaging REST API Support



messaging

Show/Hide | List Operations | Expand Operations

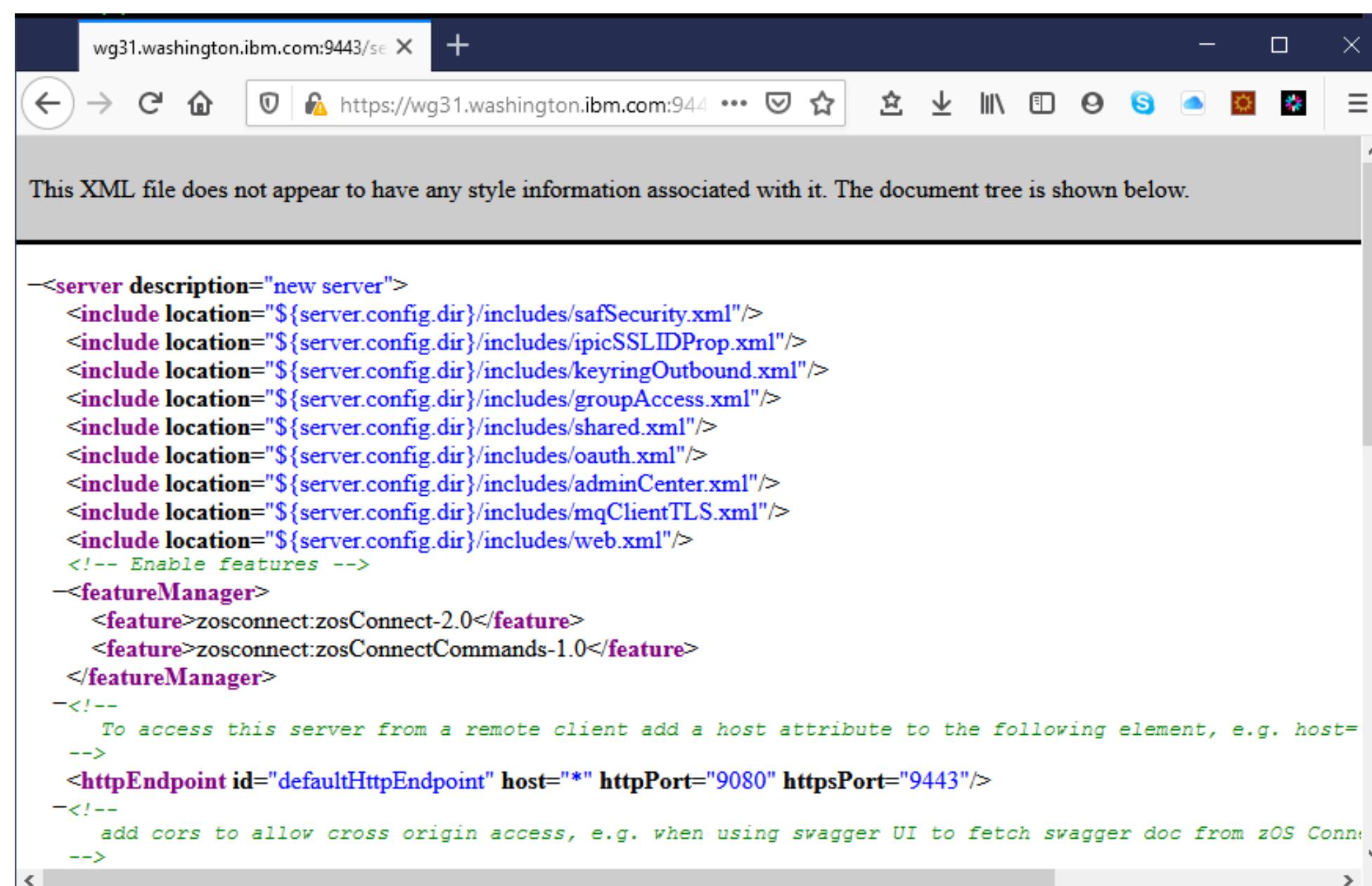
DELETE	/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{qName}/message	Retrieves the next message from a specified queue.
GET	/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{qName}/message	Browses the next message from a specified queue.
POST	/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{qName}/message	Sends a message to a specified queue.
GET	/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{qName}/messagelist	Browses messages from a specified queue.
* DELETE	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{qName}/message	Retrieves the next message from a specified queue.
* GET	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{qName}/message	Browses the next message from a specified queue.
* POST	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{qName}/message	Sends a message to a specified queue.
* GET	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{qName}/messagelist	Browses messages from a specified queue.
* POST	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/topic/{topicString}/message	Publishes a message to a specified topic.

* If you are accessing a version earlier than V9.1.5 you must use v1 rather than v2



Liberty: Enable as a file server to provide remote access to configuration/logs

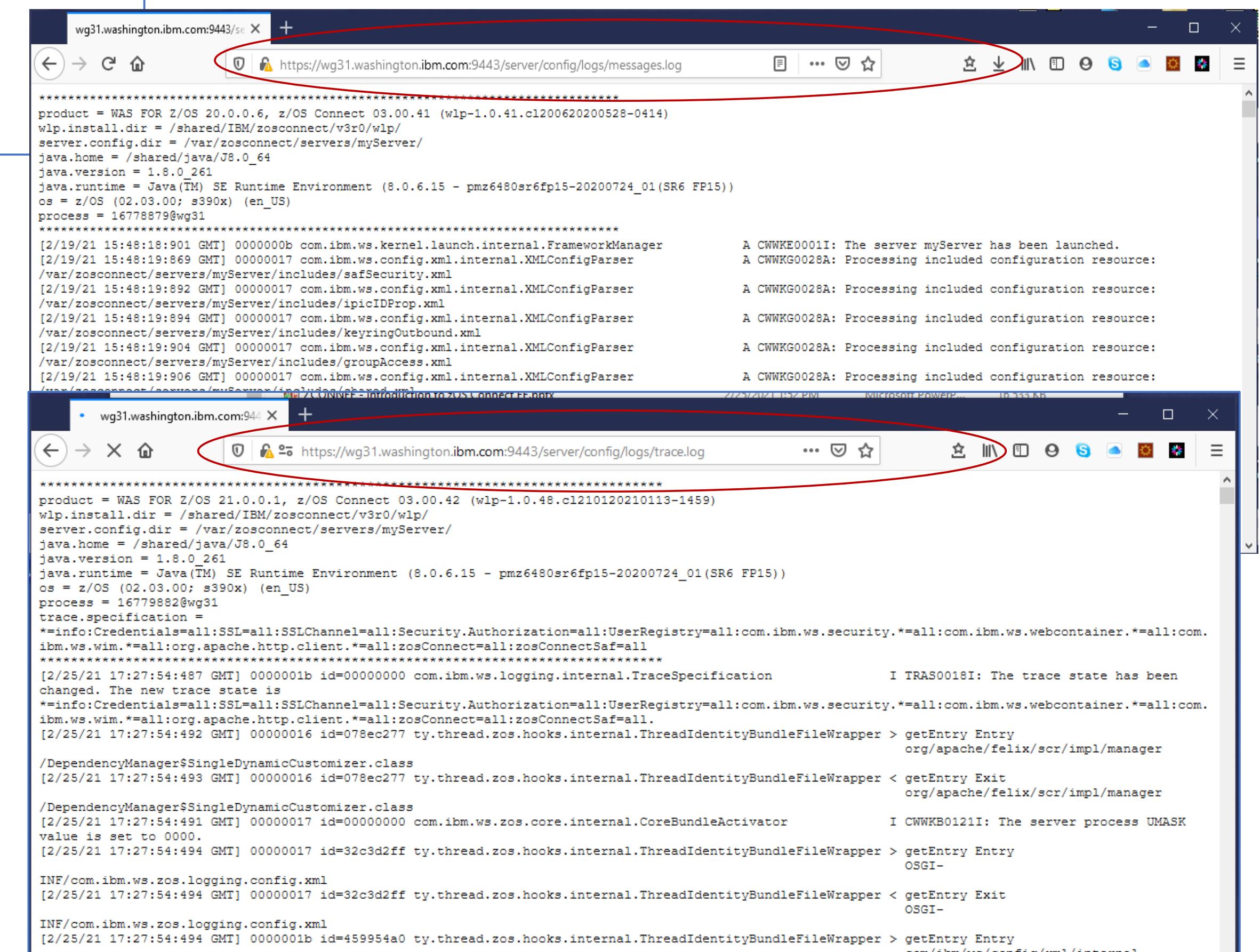
```
<webApplication id="serverConfig-location" name="serverConfig"
  location="${server.config.dir}">
  <web-ext context-root="/server/config"
    enable-file-serving="true" enable-directory-browsing="true">
    <file-serving-attribute name="extendedDocumentRoot"
      value="${server.config.dir}" />
  </web-ext>
</webApplication>
```



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<server description="new server">
  <include location="${server.config.dir}/includes/safSecurity.xml"/>
  <include location="${server.config.dir}/includes/ipcSSLIDProp.xml"/>
  <include location="${server.config.dir}/includes/keyringOutbound.xml"/>
  <include location="${server.config.dir}/includes/groupAccess.xml"/>
  <include location="${server.config.dir}/includes/shared.xml"/>
  <include location="${server.config.dir}/includes/oauth.xml"/>
  <include location="${server.config.dir}/includes/adminCenter.xml"/>
  <include location="${server.config.dir}/includes/mqClientTLS.xml"/>
  <include location="${server.config.dir}/includes/web.xml"/>
  <!-- Enable features -->
  <featureManager>
    <feature>zosconnect:zosConnect-2.0</feature>
    <feature>zosconnect:zosConnectCommands-1.0</feature>
  </featureManager>
  <!--
    To access this server from a remote client add a host attribute to the following element, e.g. host=...
  -->
  <httpEndpoint id="defaultHttpEndpoint" host="*" httpPort="9080" httpsPort="9443"/>
  <!--
    add cors to allow cross origin access, e.g. when using swagger UI to fetch swagger doc from zOS Connect
  -->

```



The browser windows show the following log content:

messages.log

```
*****
product = WAS FOR Z/OS 20.0.0.6, z/OS Connect 03.00.41 (wlp-1.0.41.c1200620200528-0414)
wlp.install.dir = /shared/IBM/zosconnect/v3r0/wlp/
server.config.dir = /var/zosconnect/servers/myServer/
java.home = /shared/java/J8_0_64
java.version = 1.8.0_261
java.runtime = Java(TM) SE Runtime Environment (8.0.6.15 - pmz6480sr6fp15-20200724_01(SR6 FP15))
os = z/OS (02.03.00; s390x) (en_US)
process = 16778879@wg31
*****
[2/19/21 15:48:18:901 GMT] 0000000b com.ibm.ws.kernel.launch.internal.FrameworkManager
[2/19/21 15:48:18:869 GMT] 00000017 com.ibm.ws.config.xml.internal.XMLConfigParser
/var/zosconnect/servers/myServer/includes/safSecurity.xml
[2/19/21 15:48:18:892 GMT] 00000017 com.ibm.ws.config.xml.internal.XMLConfigParser
/var/zosconnect/servers/myServer/includes/ipcIDProp.xml
[2/19/21 15:48:18:894 GMT] 00000017 com.ibm.ws.config.xml.internal.XMLConfigParser
/var/zosconnect/servers/myServer/includes/keyringOutbound.xml
[2/19/21 15:48:19:904 GMT] 00000017 com.ibm.ws.config.xml.internal.XMLConfigParser
/var/zosconnect/servers/myServer/includes/groupAccess.xml
[2/19/21 15:48:19:906 GMT] 00000017 com.ibm.ws.config.xml.internal.XMLConfigParser
/var/zosconnect/servers/myServer/includes/web.xml
[2/19/21 15:48:19:907 GMT] 00000017 com.ibm.ws.config.xml.internal.XMLConfigParser
*****
```

trace.log

```
*****
product = WAS FOR Z/OS 21.0.0.1, z/OS Connect 03.00.42 (wlp-1.0.48.c12101201013-1459)
wlp.install.dir = /shared/IBM/zosconnect/v3r0/wlp/
server.config.dir = /var/zosconnect/servers/myServer/
java.home = /shared/java/J8_0_64
java.version = 1.8.0_261
java.runtime = Java(TM) SE Runtime Environment (8.0.6.15 - pmz6480sr6fp15-20200724_01(SR6 FP15))
os = z/OS (02.03.00; s390x) (en_US)
process = 16779882@wg31
trace.specification =
*=info:Credentials=all:SSL=all:SSLChannel=all:Security.Authorization=all:UserRegistry=all:com.ibm.ws.security.*=all:com.ibm.ws.wim.*=all:org.apache.http.client.*=all:zosConnect=all:zosConnectSaf=all
*****
[2/25/21 17:27:54:487 GMT] 0000001b id=00000000 com.ibm.ws.logging.internal.TraceSpecification
I TRAS0018I: The trace state has been changed. The new trace state is
*=info:Credentials=all:SSL=all:SSLChannel=all:Security.Authorization=all:UserRegistry=all:com.ibm.ws.security.*=all:com.ibm.ws.wim.*=all:org.apache.http.client.*=all:zosConnect=all:zosConnectSaf=all.
[2/25/21 17:27:54:492 GMT] 00000016 id=078ec277 ty.thread.zos.hooks.internal.ThreadIdentityBundleFileWrapper > getEntry Entry org/apache/felix/scr/impl/manager
/DependencyManager$SingleDynamicCustomizer.class
[2/25/21 17:27:54:493 GMT] 00000016 id=078ec277 ty.thread.zos.hooks.internal.ThreadIdentityBundleFileWrapper < getEntry Exit org/apache/felix/scr/impl/manager
/DependencyManager$SingleDynamicCustomizer.class
[2/25/21 17:27:54:491 GMT] 00000017 id=00000000 com.ibm.ws.zos.core.internal.CoreBundleActivator
I CWWK0012I: The server process UMASK value is set to 0000.
[2/25/21 17:27:54:494 GMT] 00000017 id=32c3d2ff ty.thread.zos.hooks.internal.ThreadIdentityBundleFileWrapper > getEntry Entry OSGI-
INF/com.ibm.ws.zos.logging.config.xml
[2/25/21 17:27:54:494 GMT] 00000017 id=32c3d2ff ty.thread.zos.hooks.internal.ThreadIdentityBundleFileWrapper < getEntry Exit OSGI-
INF/com.ibm.ws.zos.logging.config.xml
[2/25/21 17:27:54:494 GMT] 0000001b id=459954a0 ty.thread.zos.hooks.internal.ThreadIdentityBundleFileWrapper > getEntry Entry com/ibm/ws/config/xml/internal
```

Liberty MVS Commands

F BAQSTRT,CACHE,CLEAR,AUTH

Clears all users that are cached in the Liberty authentication cache

F BAQSTRT,REFRESH,CONFIG

Process pending configuration updates. Configuration processing applies to the server.xml file, any files it includes

F BAQSTRT,REFRESH,APPS

Process pending application updates. (Applicable to OpenAPI 3 servers only)

F BAQSTRT,REFRESH,KEYSTORE

Use the command to refresh the keystore instorage profiles for the server.

F BAQSTRT,REFRESH,KEYSTORE, ID=*OutboundKeyRing*

To refresh a specific keystore defined in the server XML with ID=OutboundKeyRing.

F BAQSTRT,CACHE,CLEAR,AUTH

Clears all users that are cached in the Liberty authentication cache.

F BAQSTRT,PAUSE

To pause the server

F BAQSTRT,STATUS

To display the current status of a server

F BAQSTRT,RESUME

To resume the server

For more details, see URL <https://www.ibm.com/docs/en/was-liberty/zos?topic=zos-modify-commands>

Liberty MVS Angel Commands

F BAQZANGL,DISPLAY,SERVERS

Displays a list of servers currently connected to the angel

F BAQZANGL,DISPLAY,SERVERS,PID

Displays a list of servers currently connected to the angel code along with the server's PIDs.

```
CWWKB0067I ANGEL DISPLAY OF ACTIVE SERVERS
CWWKB0080I ACTIVE SERVER ASID 4d JOBNAME ZCEEAPIR PID 16777398
CWWKB0080I ACTIVE SERVER ASID 4b JOBNAME ZCEEDVM PID 50331780
CWWKB0080I ACTIVE SERVER ASID 4f JOBNAME WLPRPSRV PID 138
CWWKB0080I ACTIVE SERVER ASID 4a JOBNAME ZCEESRVR PID 50331815
CWWKB0080I ACTIVE SERVER ASID 50 JOBNAME ZCEEOPID PID 33554605
CWWKB0080I ACTIVE SERVER ASID 4c JOBNAME ZCEEHATS PID 143
CWWKB0080I ACTIVE SERVER ASID 4e JOBNAME WLPOPSRV PID 33554565
CWWKB0080I ACTIVE SERVER ASID 58 JOBNAME MQWEBS PID 152
```

F BAQZANGL,VERSION

Displays the version level of the angel

z/OS Connect MVS Commands (OpenAPI 2)

```
<feature>zosconnect:zosConnectCommands-1.0</feature>
```

F BAQSTRT,ZCON,REFRESH

All updated z/OS Connect artifacts (APIs, services, and API Requesters) are reloaded.

F BAQSTRT,ZCON,CLEARTOKENCACHE

Clears all OAuth 2.0 access tokens and JWTs from the cache. The token cache is only applicable for OAuth 2.0 access tokens and JWTs that were generated either locally or by an external authentication server, when invoking API requesters.

F BAQSTRT,ZCON,CLEARSAFCACHE

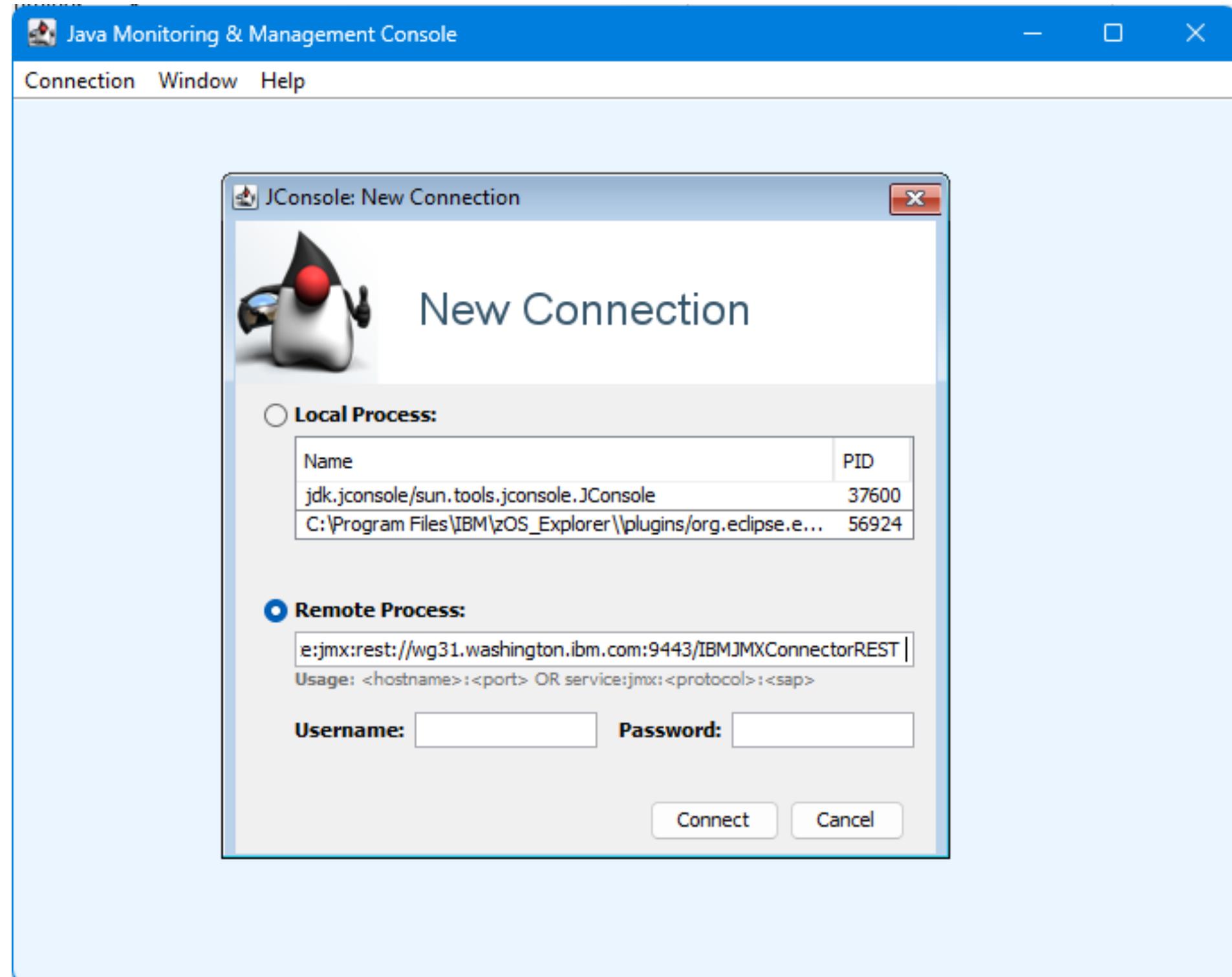
Clears the SAF cache. The SAF cache contains SAF user IDs and any associated RACF groups in which the user ID resides. The SAF cache is only applicable to API requester, and only when ID assertion is enabled.

F BAQSTRT,REFRESH,APPS

Monitoring Java and Liberty



Using Java's jConsole with Liberty features restConnector-2.0 and monitor-1.0

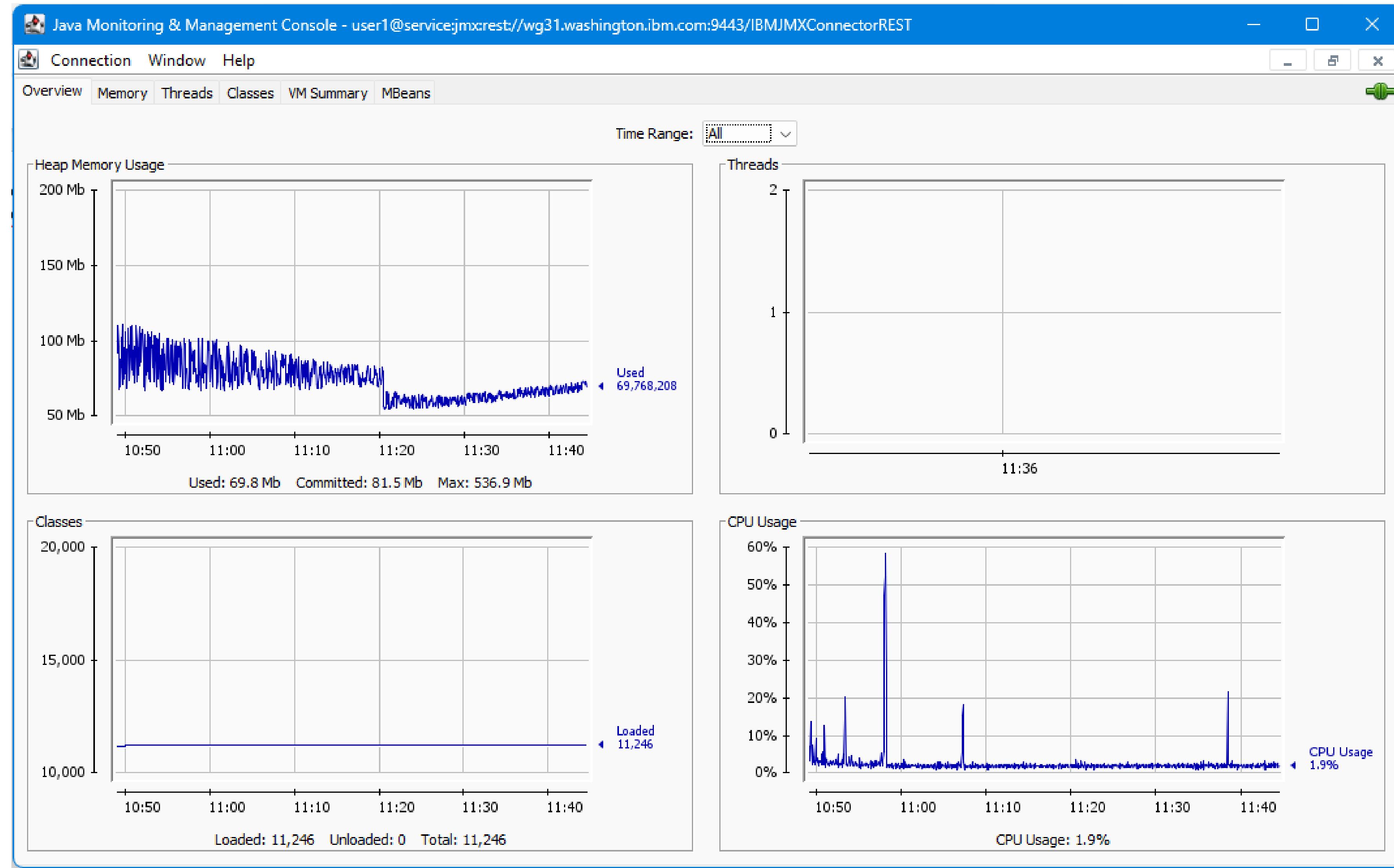


```
<server description="Additional Features">  
    <!-- Enable features -->  
    <featureManager>  
        <feature>adminCenter-1.0</feature>  
        <feature>restConnector-2.0</feature>  
        <feature>monitor-1.0</feature>  
    </featureManager>  
  
    <remoteFileAccess>  
        <readDir>/var/zcee/includes</readDir>  
        <readDir>/global/zosconnect/includes</readDir>  
        <writeDir>${server.config.dir}</writeDir>  
    </remoteFileAccess>  
  
</server>
```

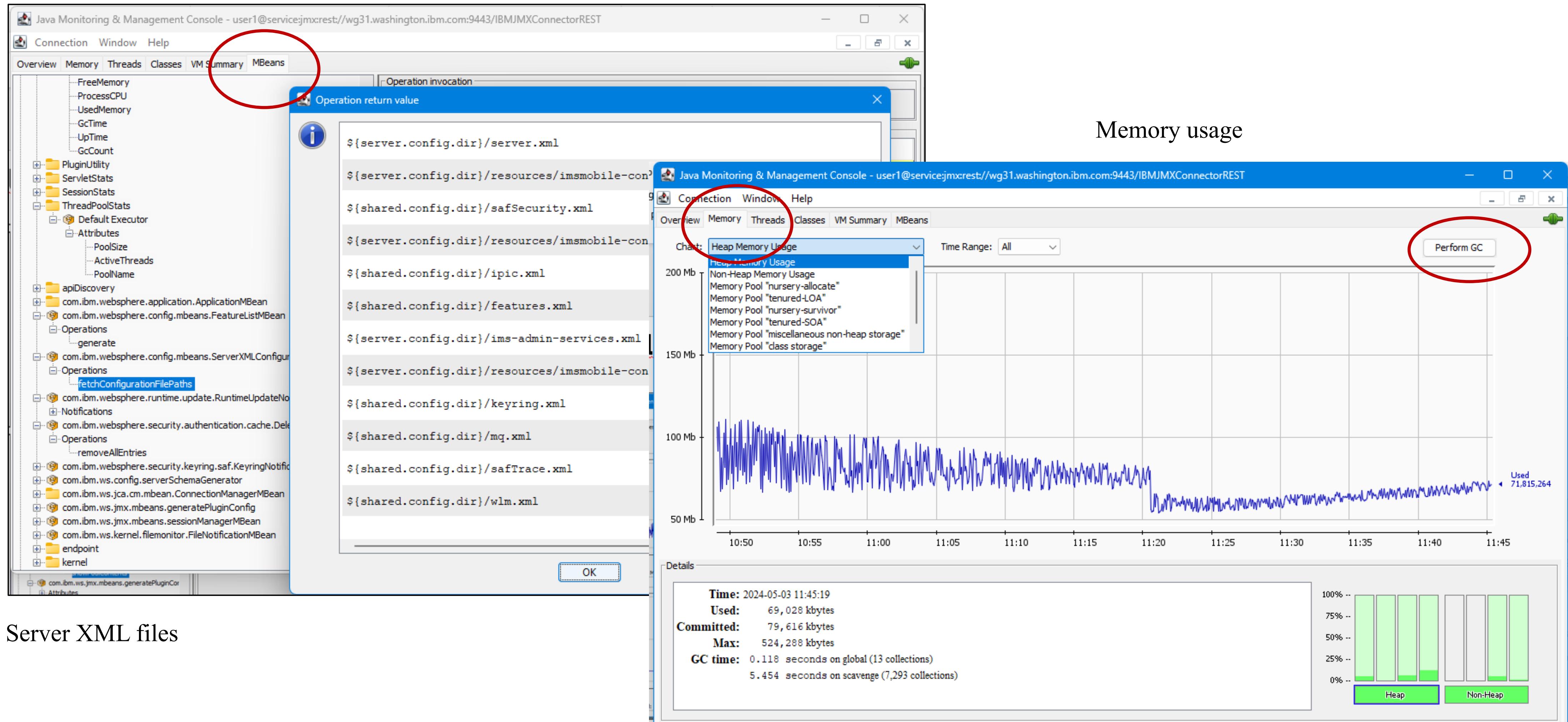
Monitoring a z/OS Liberty server using JMX and REST Clients: <https://ibm.biz/BdahXK>

```
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.Administrator OWNER(SYS1) UACC(NONE)  
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.Reader OWNER(SYS1) UACC(NONE)  
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers OWNER(SYS1) UACC(NONE)  
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.Administrator CLASS(EJBROLE) ID(ZCEEUSRS) ACCESS(READ)  
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.Reader CLASS(EJBROLE) ID(ZCEEUSRS) ACCESS(READ)  
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers CLASS(EJBROLE) ID(ZCEEUSRS) ACCESS(READ)  
SETR RACLIST(EJBROLE) REFRESH
```

Using Java's jConsole with Liberty features restConnector-2.0 and monitor-1.0



Using Java's jConsole with Liberty features restConnector-2.0 and monitor-1.0



Server XML files

Using Java's jConsole with Liberty features restConnector-2.0 and monitor-1.0



Review connection pool utilization

The screenshot shows the Java Monitoring & Management Console interface. The title bar reads "Java Monitoring & Management Console - Fred@service:jmx:rest://wg31.washington.ibm.com:9443/IBMJMXConnectorREST". The menu bar includes Connection, Window, Help, Overview, Memory, Threads, Classes, VM Summary, and MBeans. A red circle highlights the "MBeans" tab. The left pane displays a tree view of MBeans, including categories like IBM MQ, JMIImplementation, WebSphere, and various stats and configuration beans. The right pane shows the "Operation invocation" section with a call to "java.lang.String showPoolContents ()". Below it is the "MBeanOperationInfo" table:

Name	Value
Operation:	
Name	showPoolContents

A modal dialog titled "Operation return value" is open, displaying the result of the operation:

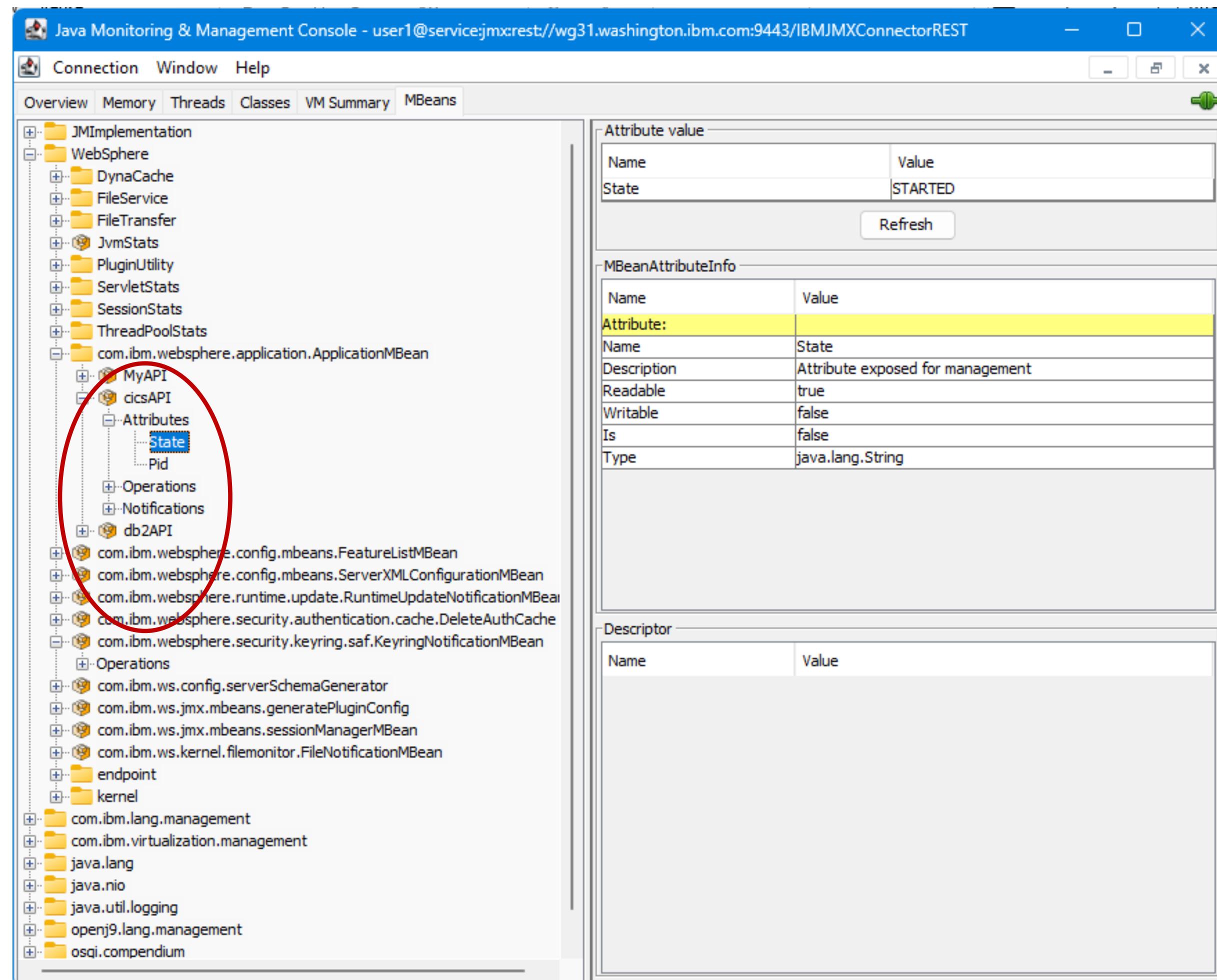
```
PoolManager@1badcb0
name=WebSphere:type=com.ibm.ws.jca.cm.mbean.ConnectionManager
jndiName=none
maxPoolSize=555
size=2
waiting=0
unshared=0
shared=0
available=2
ManagedConnection@475ca2d6=Reusable
ManagedConnection@92aa45f3=Reusable
```

At the bottom of the dialog is an "OK" button.

Using Java's jConsole with Liberty features restConnector-2.0 and monitor-1.0



Displayed and manage z/OS Connect OpenAPI3 APIs



Workload Manager - Definitions

WLM Report Classes

mpz3

Report-Class View Notes Options Help

Report Class Selection List Row 1 to 12 of 12

Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete, /=Menu Bar

-- Last Change --

Action	Name	Description	User	Date
BAQSTC			JOHNSON	2021/09/04
WMQFTE			JOHNSON	2011/08/31
WMQFTER			JOHNSON	2011/08/31
WMQFTEZ			JOHNSON	2011/08/31
ZCEEADM			JOHNSON	2021/08/02
ZCEEAPIR			JOHNSON	2021/08/05
ZEECICS			JOHNSON	2021/08/05
ZCEEDB2			JOHNSON	2021/08/05
ZCEEIMS			JOHNSON	2021/08/05
ZCEEMQ			JOHNSON	2021/08/05
ZCEEOTHR			JOHNSON	2021/08/02
ZCEESTC			JOHNSON	2021/09/02

***** Bottom of data *****

10/004

Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23

WLM Service Classes

mpz3

Service-Class Xref Notes Options Help

Modify a Service Class Row 1 to 2 of 2

Command ==> _____

Service Class Name : OPS_HIGH

Description System Tasks Velocity 70

Workload Name STC_WKL (name or ?)

Base Resource Group (name or ?)

Cpu Critical NO (YES or NO)

I/O Priority Group NORMAL (NORMAL or HIGH)

Honor Priority DEFAULT (DEFAULT or NO)

Specify BASE GOAL information. Action Codes: I=Insert new period, E>Edit period, D=Delete period.

-- Period -- Goal --

Action	#	Duration	Imp.	Description
	1		1	Execution velocity of 70

***** Bottom of data *****

19/004

Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23

mitchj@us.ibm.com

WLM "CB" Classification Rules

mpz3

Subsystem-Type Xref Notes Options Help

Modify Rules for the Subsystem Type Row 1 to 8 of 16

Command ==> _____

Subsystem Type . . . CB Fold qualifier names? N (Y or N)

Description WLP/zCEE Transactions

Action codes: A=After C=Copy M=Move I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
More ==>

Action	Type	Name	Start
1	CN	myServer	_____
2	TC	TCADM	_____
2	TC	TCAPIR	_____
2	TC	TCCICS	_____
2	TC	TCDB2	_____
2	TC	TCIMS	_____
2	TC	TCMQ	_____
2	TC	TCOTHR	_____

-----Class-----

Service	Report
OPS_HIGH	ZCEEOTHR
OPS_HIGH	BAQSTC
OPS_HIGH	ZCEEADM
OPS_HIGH	ZCEEAPIR
OPS_HIGH	ZEECICS
OPS_HIGH	ZCEEDB2
OPS_HILO	ZCEEIMS
OPS_MED	ZCEEMQ
OPS_LOW	ZCEEOTHR

07/021

mpz3

File Edit Settings View Communication Actions Window Help

Subsystem-Type Xref Notes Options Help

Modify Rules for the Subsystem Type Row 9 to 16 of 16

Command ==> _____

Subsystem Type . . . CB Fold qualifier names? N (Y or N)

Description WLP/zCEE Transactions

Action codes: A=After C=Copy M=Move I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
More ==>

Action	Type	Name	Start
1	CN	zceex	_____
2	TC	TCADM	_____
2	TC	TCAPIR	_____
2	TC	TCDB2	_____
2	TC	TCCICS	_____
2	TC	TCIMS	_____
2	TC	TCMQ	_____
2	TC	TCOTHR	_____

-----Class-----

Service	Report
OPS_HIGH	ZCEEOTHR
OPS_HIGH	ZCEESTC
OPS_HIGH	ZCEEADM
OPS_HIGH	ZCEEAPIR
OPS_HIGH	ZEECICS
OPS_HILO	ZCEEDB2
OPS_HILO	ZCEEIMS
OPS_MED	ZCEEMQ
OPS_HILO	ZCEEOTHR

07/021

mpz3

File Edit Settings View Communication Actions Window Help

Subsystem-Type Xref Notes Options Help

Modify Rules for the Subsystem Type Row 9 to 16 of 16

Command ==> _____

Subsystem Type . . . CB Fold qualifier names? N (Y or N)

Description WLP/zCEE Transactions

Action codes: A=After C=Copy M=Move I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
More ==>

Action	Type	Name	Start
1	CN	zceex	_____
2	TC	TCADM	_____
2	TC	TCAPIR	_____
2	TC	TCDB2	_____
2	TC	TCCICS	_____
2	TC	TCIMS	_____
2	TC	TCMQ	_____
2	TC	TCOTHR	_____

-----Class-----

Service	Report
OPS_HIGH	ZCEEOTHR
OPS_HIGH	ZCEESTC
OPS_HIGH	ZCEEADM
OPS_HIGH	ZCEEAPIR
OPS_HIGH	ZEECICS
OPS_HILO	ZCEEDB2
OPS_HILO	ZCEEIMS
OPS_MED	ZCEEMQ
OPS_HILO	ZCEEOTHR

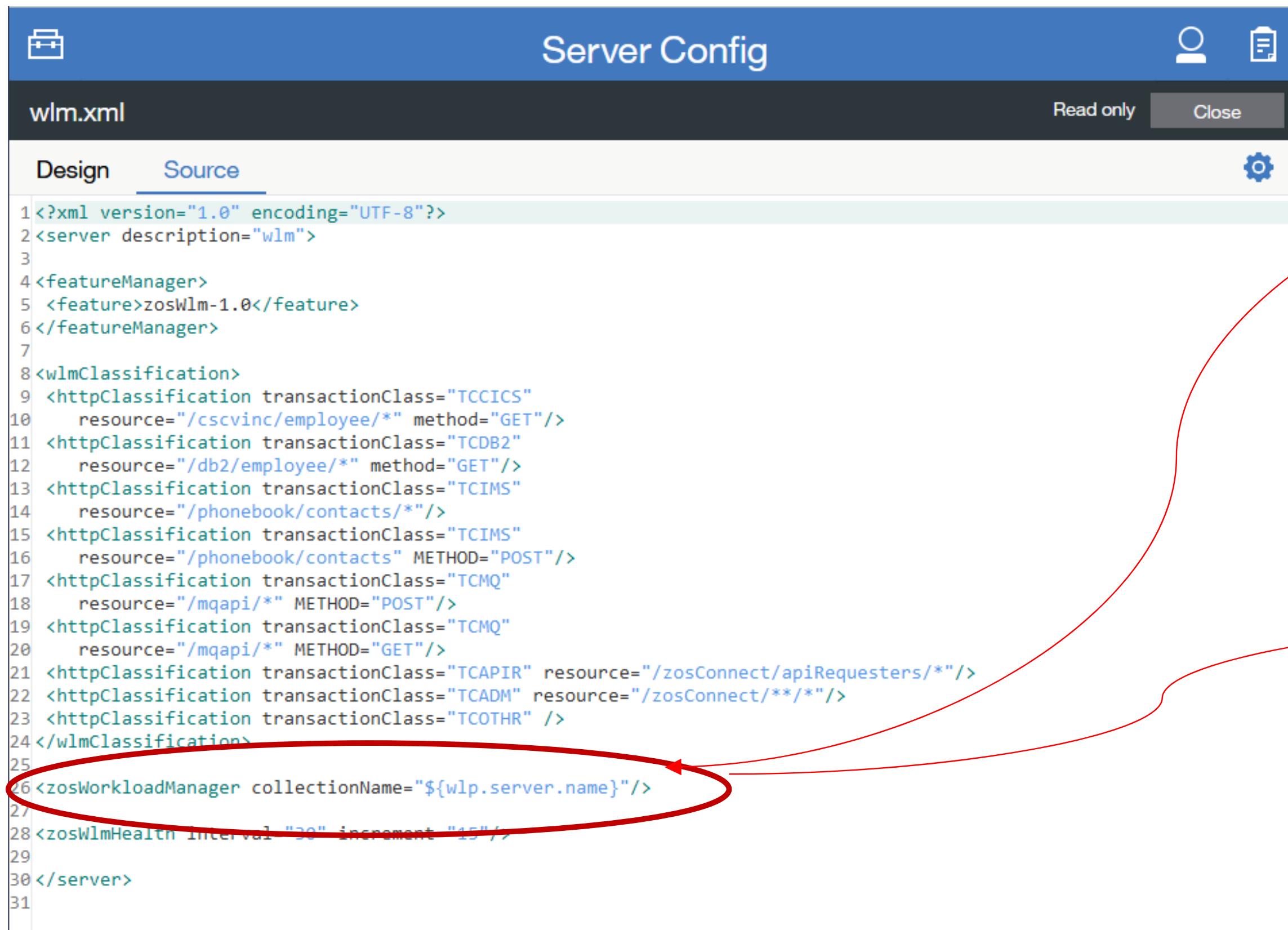
07/021

Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23

Workload Manager – WLM Classification server XML

The corresponding required server XML configuration

- Based on HTTP path matching (port and/or method can also be specified)
- The default value for the *wlmClassification* name is the name of the server
- See URL <https://www.ibm.com/docs/en/was-liberty/zos?topic=zos-wlm-classification> for more information
- The *transactionClass* attribute is required to ensure an enclave is created.

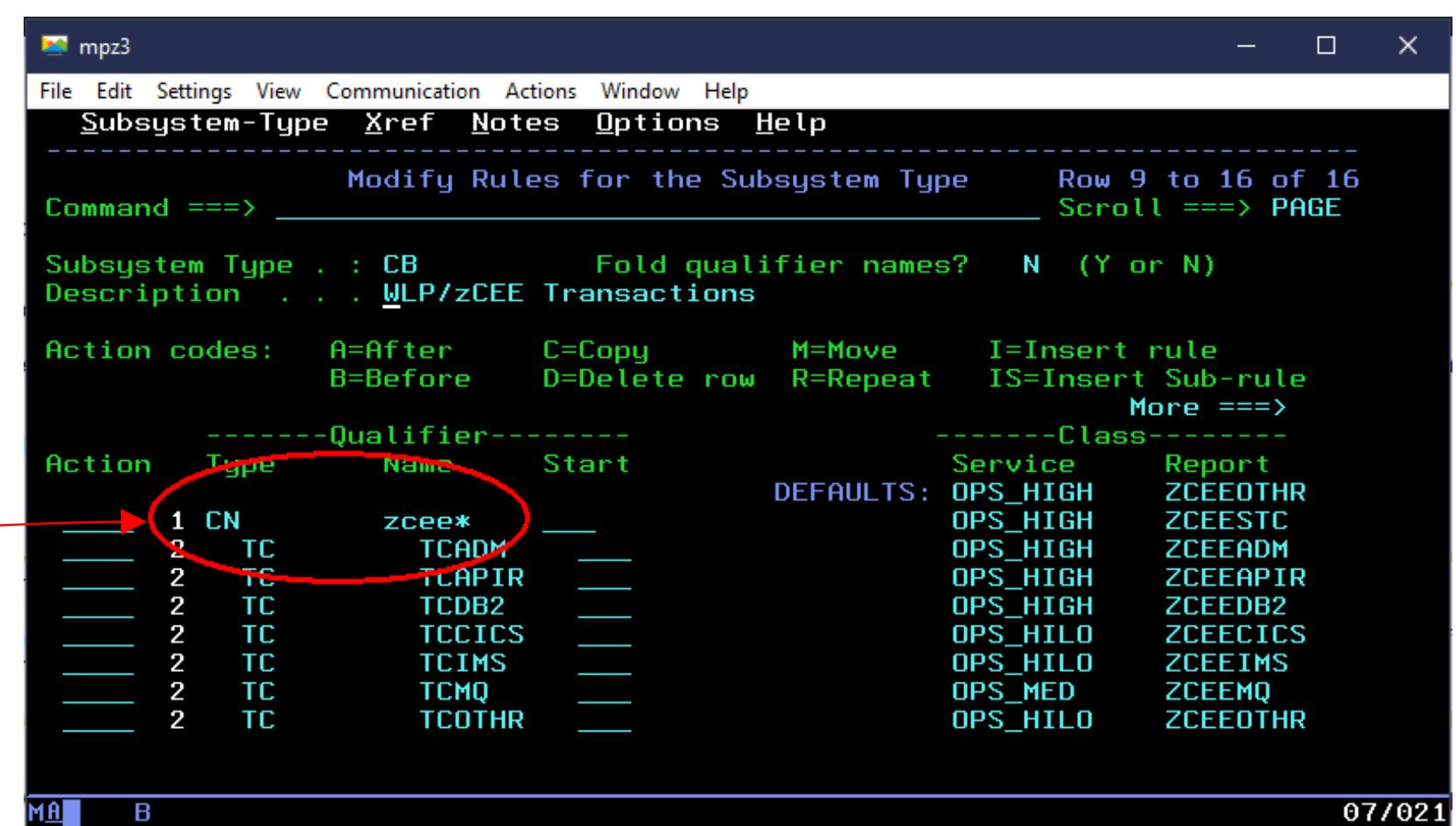


```

<?xml version="1.0" encoding="UTF-8"?>
<server description="wlm">
  <featureManager>
    <feature>zosWlm-1.0</feature>
  </featureManager>
  <wlmClassification>
    <httpClassification transactionClass="TCCICS"
      resource="/cscvinc/employee/*" method="GET"/>
    <httpClassification transactionClass="TCDB2"
      resource="/db2/employee/*" method="GET"/>
    <httpClassification transactionClass="TCIMS"
      resource="/phonebook/contacts/*"/>
    <httpClassification transactionClass="TCIMS"
      resource="/phonebook/contacts" METHOD="POST"/>
    <httpClassification transactionClass="TCMQ"
      resource="/mqapi/*" METHOD="POST"/>
    <httpClassification transactionClass="TCMQ"
      resource="/mqapi/*" METHOD="GET"/>
    <httpClassification transactionClass="TCAPIR" resource="/zosConnect/apiRequesters/*"/>
    <httpClassification transactionClass="TCADM" resource="/zosConnect/**/*"/>
    <httpClassification transactionClass="TCOTHR" />
  </wlmClassification>
  <zosWorkloadManager collectionName="${wlp.server.name}"/>
  <zosWlmHealth interval "30" increment "15"/>
</server>

```

Related to WLM CN name.



Action	Type	Name	Start	Service	Report
1	CN	zceex*		OPS_HIGH	ZCEEOTHR
2	TC	TCADM		OPS_HIGH	ZCEEESTC
2	TC	TCDB2		OPS_HIGH	ZCEEADM
2	TC	TCCICS		OPS_HILO	ZCEEAPIR
2	TC	TCIMS		OPS_HILO	ZCEEIMS
2	TC	TCMQ		OPS_MED	ZCEEMQ
2	TC	TCOTHR		OPS_HILO	ZCEEOTHR



Tech-Tip: Generate WLM Workload Activity Reports

```
//JOHNSONS JOB (ACCOUNT), NOTIFY=&SYSUID, REGION=0M,  
// CLASS=A, MSGCLASS=H, MSGLEVEL=(1, 1)  
//DELETE EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
    DELETE JOHNSON.DUMPSMF.SORT  
//RMFSORT EXEC PGM=SORT, REGION=0M  
//SORTIN DD DISP=SHR, DSN=MPZ3.DUMPSMF  
//SORTOUT DD DISP=(,CATLG), DSN=JOHNSON.DUMPSMF.SORT,  
//           SPACE=(CYL,(100,50),RLSE), UNIT=SYSDA  
//SORTWK01 DD DISP=(NEW,DELETE), UNIT=SYSDA, SPACE=(CYL,(100))  
//SORTWK02 DD DISP=(NEW,DELETE), UNIT=SYSDA, SPACE=(CYL,(100))  
//SORTWK03 DD DISP=(NEW,DELETE), UNIT=SYSDA, SPACE=(CYL,(100))  
//SORTWK04 DD DISP=(NEW,DELETE), UNIT=SYSDA, SPACE=(CYL,(100))  
//SORTWK05 DD DISP=(NEW,DELETE), UNIT=SYSDA, SPACE=(CYL,(100))  
//SORTWK06 DD DISP=(NEW,DELETE), UNIT=SYSDA, SPACE=(CYL,(100))  
//SORTWK07 DD DISP=(NEW,DELETE), UNIT=SYSDA, SPACE=(CYL,(100))  
//SYSPRINT DD SYSOUT=(,)  
//SYSOUT DD SYSOUT=(,)  
//SYSIN DD *  
    SORT FIELDS=(11,4,CH,A,7,4,CH,A), EQUALS  
    MODS E15=(ERBPPE15,36000,,N), E35=(ERBPPE35,3000,,N)  
//RMFPP EXEC PGM=ERBRMFPP, REGION=0M  
//SYSUDUMP DD SYSOUT=*  
//STEPLIB DD DSN=SYS1.COMBINED.LINKLIB, DISP=SHR  
//MFPIINPUT DD DISP=SHR, DSN=JOHNSON.DUMPSMF.SORT  
//MFPMMSGDS DD SYSOUT=*  
//SYSIN DD *  
    SYSOUT(O)  
    SYSRPTS(WLMGL(RCPER)) /*WORKLOAD ACTIVITY REPORT */
```

RMF SMF Type 72 Service Class Reports

mpz3

File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help

SDSF OUTPUT DISPLAY JOHNSONR JOB12740 DSID 112 LINE CHARS 'CICS' FOUND
COMMAND INPUT ===>
POLICY=WSCPOL

REPORT CLAS

- TRANSACTIONS --		TRANS-TIME	HHH.MM.SS.FFFFFFF	TRA
Avg	0.02	ACTUAL	108891	TOT
MPL	0.02	EXECUTION	108856	MOB
ENDED	96	QUEUED	34	CAT
END/S	0.16	R/S AFFIN	0	CAT
#SWAPS	0	INELIGIBLE	0	
EXCTD	0	CONVERSION	0	
		STD DEV	762583	
----SERVICE----	SERVICE TIME	--APPL %--	--P	
IOC	0 CPU	1.967 CP	0.02 AAP	BLK
CPU	1739K SRB	0.000 IIPCP	0.02 ENQ	
MSO	0 RCT	0.000 IIP	0.31 CRM	
SRB	0 IIT	0.000 AAPCP	0.00 LCK	
TOT	1739K HST	0.000 AAP	N/A SUP	
/SEC	2898 IIP	1.844		
ABSRPTN	166K AAP	N/A		
TRX SERV	166K			

MA A
Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23

mpz3

File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help

SDSF OUTPUT DISPLAY JOHNSONR JOB12740 DSID 112 LINE CHARS 'APIR' FOUND
COMMAND INPUT ===>
SCROLL ==> PAGE
POLICY=WSCPOL

REPORT CLASS=ZCEEAPIR PERIOD=1

- TRANSACTIONS --		TRANS-TIME	HHH.MM.SS.FFFFFFF	TRANS-APPL%-----CP-IIPCP/AAPCP-IIP/AAP	---ENCLAVES---	
Avg	0.14	ACTUAL	424835	TOTAL 0.12 0.12 0.73	Avg Enc 0.14	
MPL	0.14	EXECUTION	424707	MOBILE 0.00 0.00 0.00	Rem Enc 0.00	
ENDED	200	QUEUED	126	CATEGORYA 0.00 0.00 0.00	MS Enc 0.00	
END/S	0.33	R/S AFFIN	0	CATEGORYB 0.00 0.00 0.00		
#SWAPS	0	INELIGIBLE	0			
EXCTD	0	CONVERSION	0			
		STD DEV	1.381943			
----SERVICE----	SERVICE TIME	--APPL %--	--PROMOTED--	--DASD I/O--	----STORAGE----	-PAGE-IN RATES-
IOC	0 CPU	5.073 CP	0.12	BLK 0.000 SSCHRT 2.4	Avg 0.00 SINGLE 0.0	
CPU	4485K SRB	0.000 IIPCP	0.12	ENQ 0.000 RESP 0.4	Total 0.00 BLOCK 0.0	
MSO	0 RCT	0.000 IIP	0.73	CRM 0.000 CONN 0.3	Shared 0.00 Shared 0.0	
SRB	0 IIT	0.000 AAPCP	0.00	LCK 0.000 DISC 0.0		
TOT	4485K HST	0.000 AAP	N/A SUP	0.000 Q+PEND 0.0	HSP 0.0	
/SEC	7474 IIP	4.363		IOSQ 0.0		
ABSRPTN	53K AAP	N/A				
TRX SERV	53K					

MA A
Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23



Liberty SMF 120 Subtype 11

WebSphere Liberty Profile (WLP) can generate various types of SMF 120 records. Support for a SMF 120 record relevant for z/OS Connect was added in WLP V16.0.0.2. This record, a SMF 120 Subtype 11, is generated for each HTTP request received by the Liberty server. For more details and a description of the contents of this record, see URL <https://www.ibm.com/support/pages/liberty-zos-smf-120-11-version-2>



The screenshot shows the 'Server Config' interface with a blue header bar. On the left is a briefcase icon, in the center is the title 'Server Config', and on the right are user and save icons. Below the header is a dark navigation bar with the text 'smf.xml' on the left and 'Read only' and 'Close' buttons on the right. The main area contains two tabs: 'Design' and 'Source'. The 'Source' tab is selected, showing XML code. The code is numbered from 1 to 10 and defines a server configuration:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <server description="SMF">
4   <featureManager>
5     <feature>monitor-1.0</feature>
6     <feature>zosRequestLogging-1.0</feature>
7   </featureManager>
8
9 </server>
10
```

Useful Plug-ins for WAS z/OS SMF 120.9 Browser

<https://www.ibm.com/support/pages/node/6355403>



Tech-Tip: BBOSMFV (Extract Liberty SMF 120 Subtype 11 records)

```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=JOHNSON,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1, 1)  
//EXPORT EXPORT SYMLIST=(*)  
// SET REPORT='LibertyExport'  
//JAVA EXEC PROC=JVMPRC86,  
// JAVAACLs='com.ibm.ws390.sm.smfview.JclSmf'  
//STDENV DD DISP=SHR,DSN=JOHNSON.JCLLIB.CNTL(STDENV)  
//SMFDATA DD DISP=SHR,DSN=MPZ3.DUMPSMF  
//SMFENV DD *,SYMBOLS=EXECSYS  
# Specify the plugin to use  
plugin=&REPORT  
# Specify where the output goes  
output=/u/johnson/&REPORT..csv  
# Uncomment (and change the value as appropriate) to filter  
#matchServer=BAQSTRT
```

```
JOHNSON.JCLLIB.CNTL(STDENV)  
. /etc/profile  
export JAVA_HOME=/usr/lpp/java/J8.0_64  
export PATH=/bin:"${JAVA_HOME}"/bin  
  
LIBPATH=/lib:/usr/lib:"${JAVA_HOME}"/bin  
LIBPATH="$LIBPATH":"${JAVA_HOME}"/lib/s390x  
LIBPATH="$LIBPATH":"${JAVA_HOME}"/lib/s390x/j9vm  
LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin/classic  
export LIBPATH="$LIBPATH":  
  
# Customize your CLASSPATH here  
APP_HOME=$JAVA_HOME  
CLASSPATH=$APP_HOME:"${JAVA_HOME}"/lib:"${JAVA_HOME}"/lib/ext  
CLASSPATH=/u/johnson/lib/bbosmfv.jar:$CLASSPATH  
CLASSPATH=/u/johnson/lib/WP102312_Plugins.jar:$CLASSPATH  
  
# Add Application required jars to end of CLASSPATH  
for i in "${APP_HOME}"/*.jar; do  
    CLASSPATH="$CLASSPATH":$i"  
done  
export CLASSPATH="$CLASSPATH":  
  
# Configure JVM options  
IJO="-Xms16m -Xmx128m"  
export IBM_JAVA_OPTIONS="$IJO "
```



Liberty SMF 120 Subtype 11 – WP102312 Plugin

AutoSave (Off) Mitch Johnson MJ

File Home Insert Page Layout Formulas Data Review View Help ACROBAT

Cut Copy Format Painter Paste Share Comments

Font Alignment Number Styles Cells Editing Ideas Sensitivity

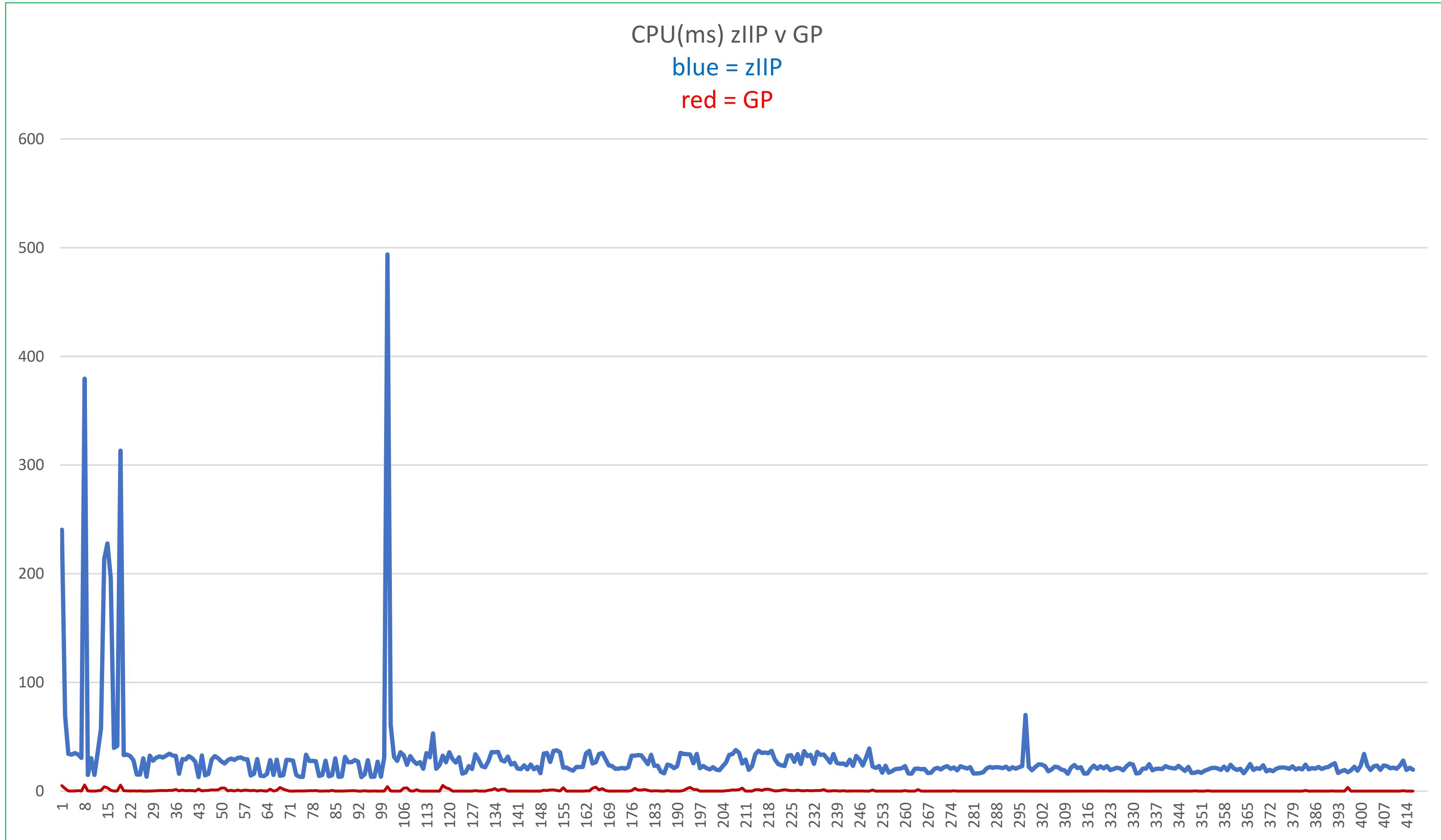
AS9 : 166

LibertyExport.csv

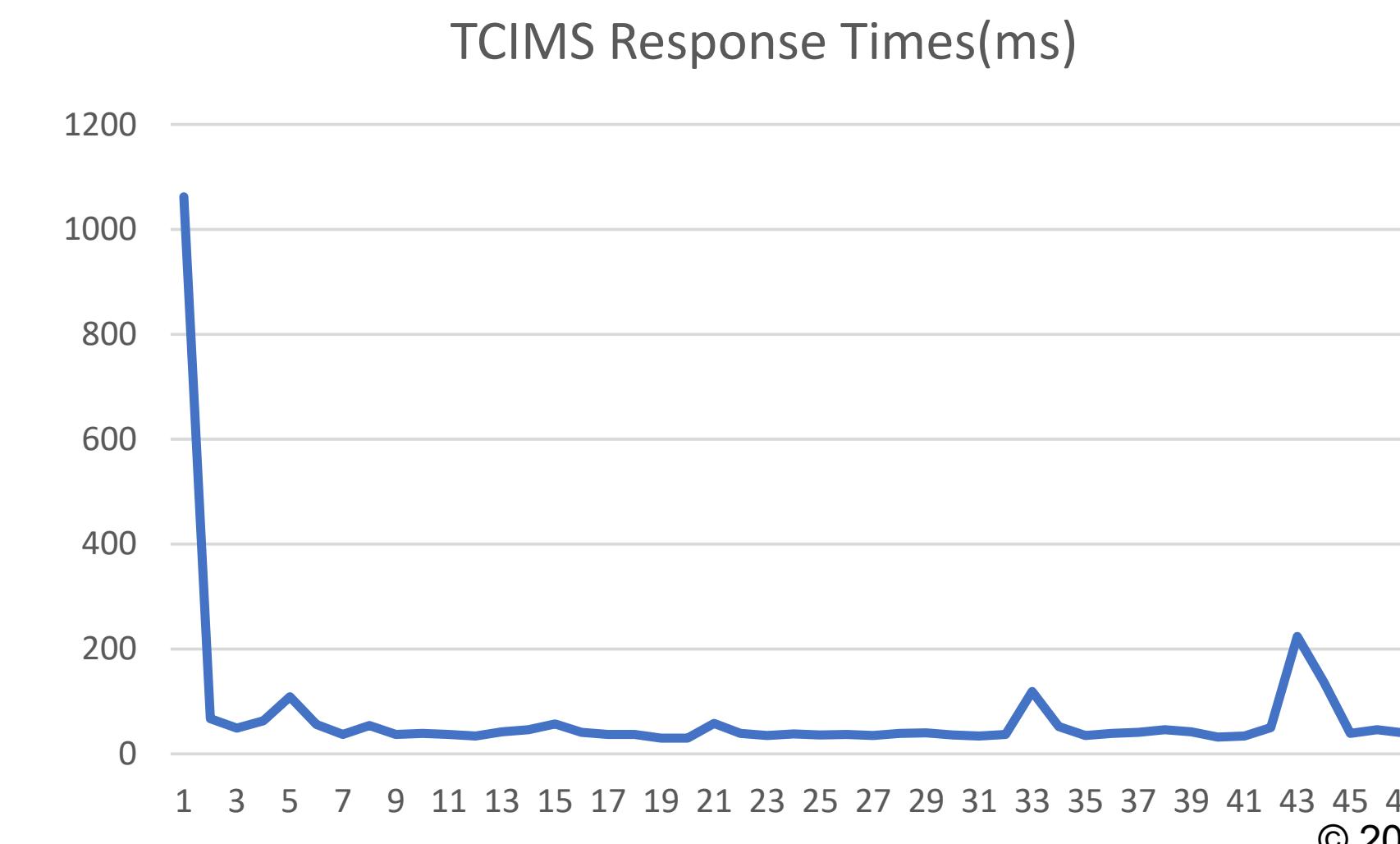
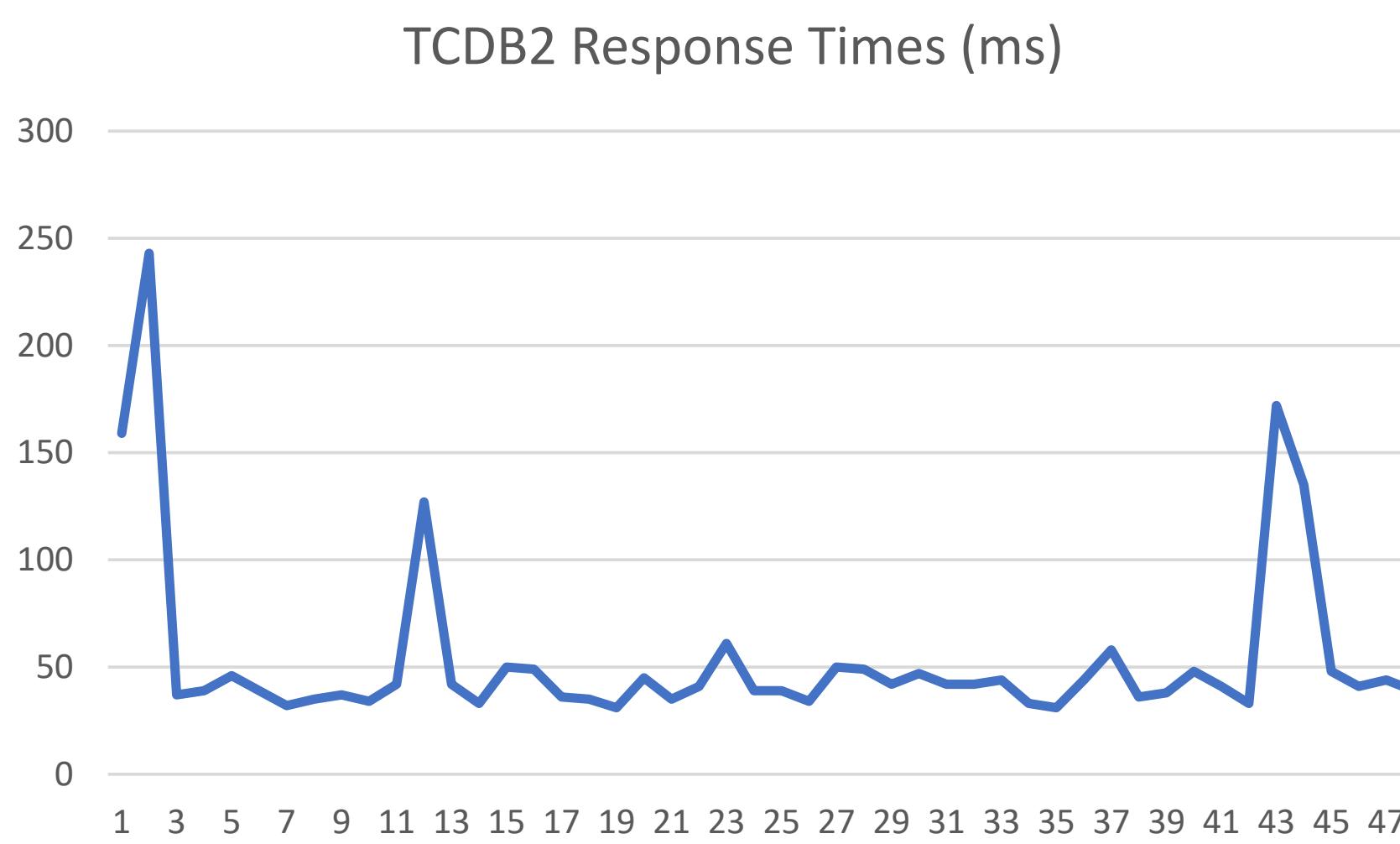
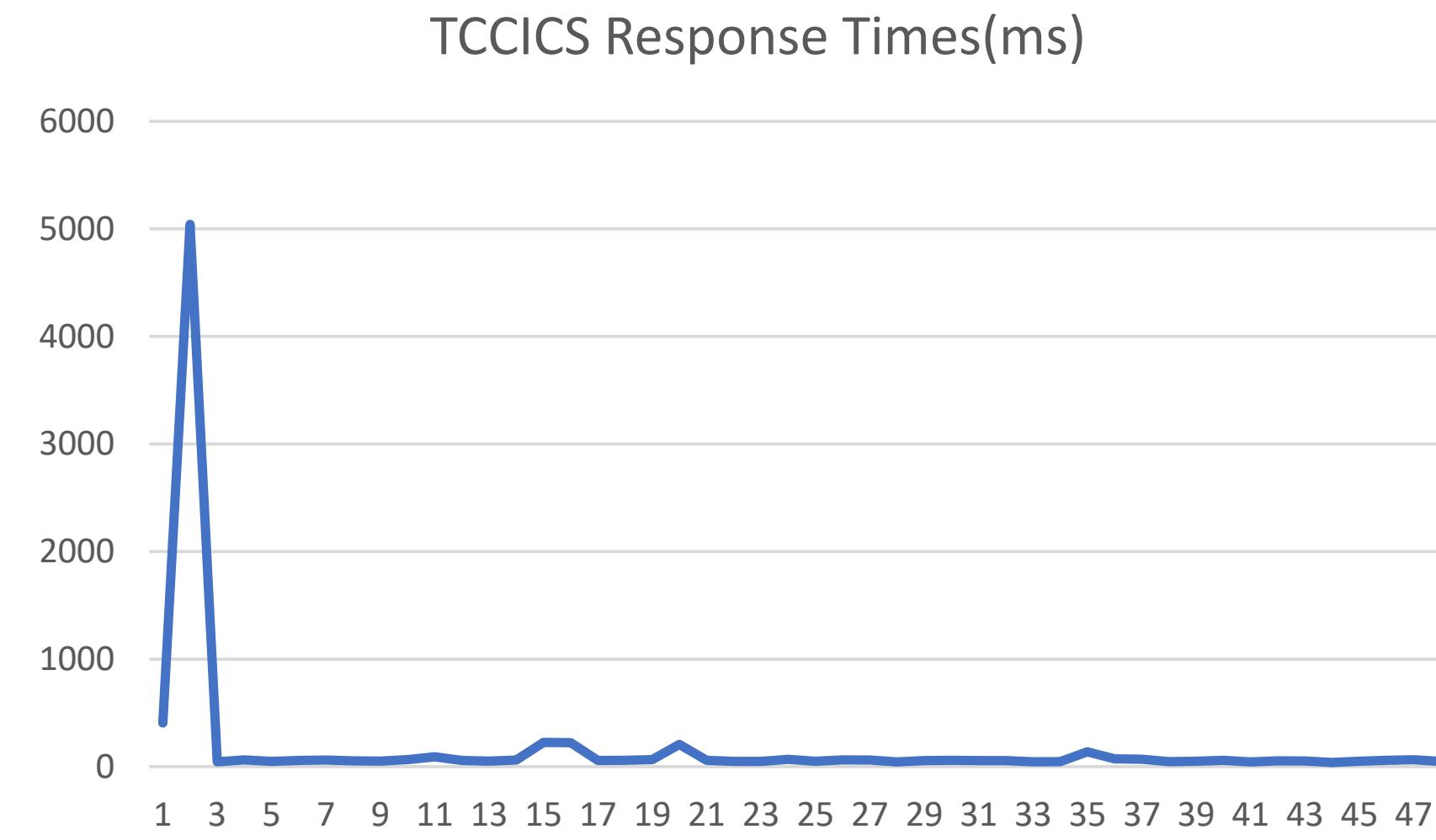
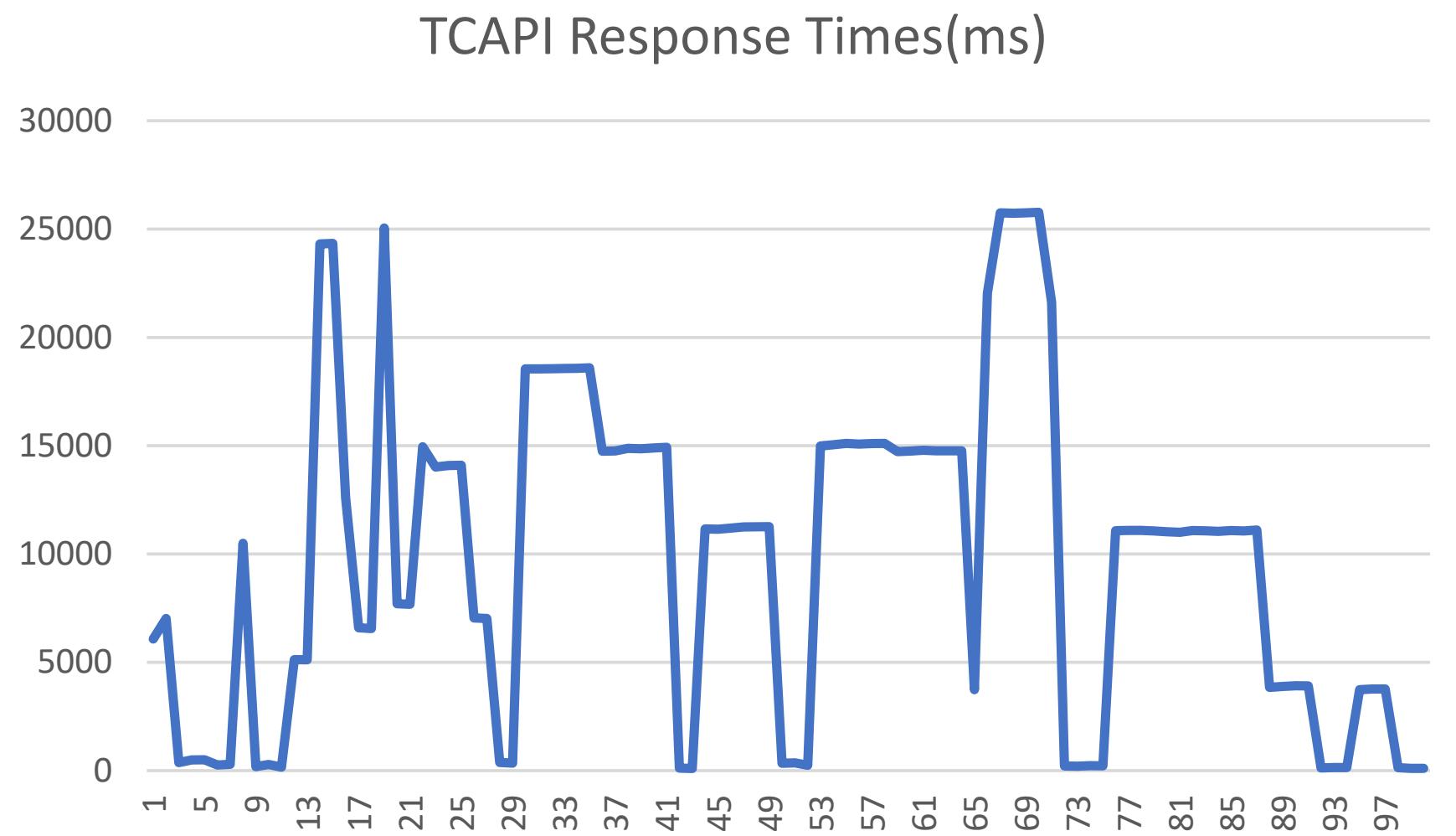
Search

B	C	E	P	Q	R	S	T	U	V	W	Z	AA	AB	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	
1	SystemName	SysplexName	JobName	StartTime	StartTime	EndTime	EndTime	Respo	TranClass	TotalCPUStart	TotalCPUEnd	TotalCPU(ms)	TotalGP(ms)	TotalOffload(ms)	userid	MappedUser	requestUser	host	port	uri	response	targetPort	remotePort	RemoteAddr	
2	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	6080	TCAPIR	3314772936	4.32E+09	245.5195	5.0110927	240.508381	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4283	192.168.17.243
3	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7030	TCAPIR	178821759	471750165	71.51572	2.334169	69.18156	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4286	192.168.17.243
4	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	374	TCAPIR	4327455460	4.469E+09	34.44008	0.10757129	34.332504	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4301	192.168.17.243
5	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	495	TCAPIR	2762287407	2.9E+09	33.65053	0.057430662	33.5931	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4304	192.168.17.243
6	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	500	TCAPIR	4484655211	4.629E+09	35.15451	0.12540185	35.020004	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4303	192.168.17.243
7	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	262	TCAPIR	4637789017	4.777E+09	34.10823	0.42818993	33.680042	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4305	192.168.17.243
8	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	293	TCAPIR	542458283	668050357	30.66213	0.053870115	30.608257	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4306	192.168.17.243
9	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	10493	TCAPIR	3802597962	5.38E+09	385.0374	5.576215	379.46115	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4285	192.168.17.243
10	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	185	TCAPIR	5384541333	5.446E+09	15.04486	0.15656103	14.888303	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4308	192.168.17.243
11	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	282	TCAPIR	1028119195	1.153E+09	30.38298	0.04661279	30.336363	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4309	192.168.17.243
12	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	163	TCAPIR	901260513	962209631	14.88016	0	14.8801565	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4310	192.168.17.243
13	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	5126	TCAPIR	3137255105	3.284E+09	35.92899	0.33009765	35.598892	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4313	192.168.17.243
14	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	5122	TCAPIR	4890213483	5.128E+09	58.01673	0.61064285	57.40609	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4314	192.168.17.243
15	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	24315	TCAPIR	13036032356	1.393E+10	217.4406	4.0119	213.4287	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4282	192.168.17.243
16	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	24338	TCAPIR	1463812131	2.41E+09	230.9845	3.1036336	227.8809	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4284	192.168.17.243
17	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	12587	TCAPIR	1160912461	1.967E+09	196.8579	0.7669092	196.09096	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4315	192.168.17.243
18	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	6599	TCAPIR	5303866625	5.467E+09	39.78177	0.020269532	39.761494	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4316	192.168.17.243
19	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	6565	TCAPIR	6143860672	6.315E+09	41.86705	0.16208105	41.704967	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4317	192.168.17.243
20	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	25052	TCAPIR	2622790027	3.928E+09	318.7149	5.489483	313.22546	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4281	192.168.17.243
21	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7709	TCAPIR	4477460136	4.615E+09	33.52233	0.35891944	33.163406	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4322	192.168.17.243
22	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7682	TCAPIR	1973032107	2.112E+09	33.81701	0.19548193	33.621525	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4321	192.168.17.243
23	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	14950	TCAPIR	458083508	590213570	32.25832	0.0489917	32.209324	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4323	192.168.17.243
24	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	14016	TCAPIR	61401222	178390269	28.56178	0.2347461	28.327032	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4325	192.168.17.243
25	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	14088	TCAPIR	86069826	148846164	15.32625	0.0541626	15.272091	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4326	192.168.17.243
26	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	14097	TCAPIR	5471350509	5.535E+09	15.43587	0.21740967	15.218459	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4324	192.168.17.243
27	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7051	TCAPIR	5358173556	5.482E+09	30.16547	0.001757324	30.163715	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4328	192.168.17.243
28	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7029	TCAPIR	2281578411	2.336E												

Liberty SMF 120 type 11 – GP v zIIP comparison example



Liberty SMF 120 type 11 – Response times comparisons example





Liberty Connection Management



Inbound persistent connections

- Persistent connections can be used to avoid too many handshakes
- Configured by setting the `keepAliveEnabled` attribute on the `httpOptions` element to **true**
- Example setting `server.xml` file

```
<httpEndpoint host="*" httpPort="80" httpsPort="443" id="defaultHttpEndpoint"
httpOptionsRef="httpOpts"/>

<httpOptions id="httpOpts" keepAliveEnabled="true" maxKeepAliveRequests="500"
persistTimeout="1m" />
```

- This sets the connection timeout to **1 minute** (default is 30 seconds) and sets the maximum number of persistent requests that are allowed on a single HTTP connection to **500**
- It is recommended to set a maximum number of persistent requests when connection workload balancing is configured
- It is also necessary to configure the client to support persistent connections



TLS sessions

- When connections timeout, it is still possible to avoid the impact of full handshakes by reusing the TLS session id
- Configured by setting the `sslSessionTimeout` attribute on the `sslOptions` element to an amount of time
- Example setting `server.xml` file

```
<httpEndpoint host="*" httpPort="80" httpsPort="443" id="defaultHttpEndpoint"  
httpOptionsRef="httpOpts" sslOptionsRef="mySSLOptions"/>  
  
<httpOptions id="httpOpts" keepAliveEnabled="true" maxKeepAliveRequests="100"  
persistTimeout="1m"/>  
  
<sslOptions id="mySSLOptions" sslRef="DefaultSSLSettings"  
sslSessionTimeout="10m"/>
```

- This sets the timeout limit of an TLS session to **10 minutes** (default is 8640ms)



Liberty Connection Management

Liberty default connection pool management <connectionManager>

- agedTimeout The number of seconds before a physical connection can be discarded by the pool maintenance thread. The default value of -1 disables this timeout.
- connectionTimeout Amount of time after which a connection request will time out with an exception when no connections are available.
- maxIdleTime Amount of time a connection can be unused or idle until it can be discarded during pool maintenance.
- maxPoolSize Maximum number of physical connections for a pool.
- minPoolSize Minimum number of physical connections to maintain in the pool.
- purgePolicy Specifies which connections to destroy when a “stale” connection is detected in a pool (EntirePool, FailingConnectionOnly or ValidateAllConnections)
- reapTime Amount of time between runs of the pool maintenance thread. Should be less than agedTimeout and maxIdleTime.

```
<connectionManager id="ConMgr1">  
    agedTimeout=-1  
    connectionTimeout=30s  
    maxIdleTime=1800s  
    maxPoolSize=50  
    minPoolSize=0  
    purgePolicy= "EntirePool"  
    reapTime=180/>
```



Connection Management for Db2

Use the connectionManagerRef attribute in a *datasource* to provide a connection pool for connections to Db2.

```
<library id="jdbcLib">
  <fileset dir="/usr/lpp/db2/jdbc/classes"
    includes="db2jcc4.jar db2jcc_license_cisuz.jar" />
  <fileset dir="/usr/lpp/db2/jdbc/lib"
    includes="libdb2jcctzos4_64.so" />
</library>
<connectionManager id="Db2ConnMgr" agedTimeout=-1 connectionTimeout=30 maxIdleTime=1800 maxPoolSize=50
minPoolSize=0 purgePolicy="EntirePool" reapTime=180/>

<dataSource id="db2type4" jndiName="jdbc/sample4" connectionManagerRef="Db2ConnMgr"
  type="javax.sql.XADataSource">
  <jdbcDriver libraryRef="jdbcLib" />
  <properties.db2.jcc driverType="4" databaseName="DNS2LOC" currentSchema="DSN81210"
    serverName="wg31.washington.ibm.com" portNumber="2446" user="USER1" password="user1" />
</dataSource>

<dataSource id="db2type2" jndiName="jdbc/sample2" connectionManagerRef="Db2ConnMgr"
  transactional="false">
  <jdbcDriver libraryRef="jdbcLib" />
  <properties.db2.jcc driverType="2" currentSchema="DSN81210" />
</dataSource>
```



IMS Connect TCPIP configuration parameters

In the IMS Connect **TCPIP** configuration statement, you can set the following limits on the amount of time IMS Connect waits in the following stages of communication:

- The **TIMEOUT** parameter determines how long IMS Connect keeps a connection open if the client does not send any input after the connection is **first** established.
- The **IDLETO** parameter determines how long IMS Connect keeps a connection that is in **RECV** state open **after** the prior client interaction completes.
- The **TIMEOUT** parameter also determines how long IMS Connect waits for a response from IMS before IMS Connect notifies the client of the timeout and returning the socket connection to a **RECV** state.

```
HWS=(ID=IMS15HWS,XIBAREA=100,RACF=Y,RRS=Y)
TCPIP=(HOSTNAME=TCPIP,
       ECB=Y,           This allows TCP/IP to post an ECB into IMS Connect to improve performance
       PORT=(ID=4000, IDLETO=4500), Idle timeout value set to 4500 hundredths of a second
       PORT=(ID=4001),           Idle timeout value is set to global default
       PORT=(ID=4002, IDLETO=3600), Idle timeout value set to 3600 hundredths of a second
       PORTID=(4003),           Idle timeout value is set to global default
       IDLETO=5500,             Global idle timeout value set to 5500 hundredths of a second
       RACFID=JOHNSON,
       TIMEOUT=5000)
```



IMS Connection Management Attributes

IMS connection profiles (imsmobile_imsConnection)

- **connectionFactoryRef** – Set this file to the name (ID) of the ConnectionFactory configuration element

IMS interaction profiles (imsmobile_interaction)

- **imsConnectTimeout** - Specify the time in milliseconds to wait for a reply after sending a message to IMS Connect.
General guidelines for setting the imsConnectTimeout value:
 - This value should be equal or larger than the value for interactionTimeout.
 - This value should be at least 5 seconds shorter than the value for the *asyncRequestTimeout* attribute of the *zosconnect_zosConnectManager* element in server.xml.
- **interactionTimeout** - Specify the time in milliseconds for the transaction to be processed by IMS. After sending a message to IMS, IMS Connect waits for a reply from IMS until this timeout value is reached..
 - Valid values are -1, 0, or between 1 and 3600000 (one hour), inclusively.
 - A value of 0 means that the timeout value is determined by IMS Connect.
 - A value of -1 (the default) means to wait indefinitely.
- **transExpiration** - Indicates to IMS OTMA, when the execution timeout value is reached, whether the transaction is considered expired, and OTMA no longer needs to process it. When this property is set to true, the IMS TM resource adapter client application indicates to OTMA that the transaction can be discarded after the execution times out. This function relieves OTMA from having to process unnecessary messages. The default is false.

Tip: The imsConnectTimeout value should be equal or larger than the value for interactionTimeout.



Connection Management for IMS TM

Use the connectionManagerRef attribute in an IMS ConnectionFactory to provide a connection pool for connections to IMS Connect.

```
<connectionManager id="IMSTMConnMgr1" agedTimeout=-1 connectionTimeout=30 maxIdleTime=1800 maxPoolSize=50  
minPoolSize=0 purgePolicy="EntirePool" reapTime=180/>  
<connectionManager id="IMSTMConnMgr2" agedTimeout=-1 connectionTimeout=30 maxIdleTime=1800 maxPoolSize=200  
minPoolSize=0 purgePolicy="EntirePool" reapTime=180/>  
  
<imsmobile_imsConnection id="IMSCONN1" connectionFactoryRef="IMSCF1"/>  
<connectionFactory id="IMSCF1" connectionManagerRef="IMSTMConnMgr1" containerAuthDataRef="Connection1_Auth" >  
    <properties.gmoa hostName="wg31.washington.ibm.com" portNumber="4000" applicationName="IMSTMPL"/>  
</connectionFactory>  
  
<imsmobile_imsConnection id="IMSCONN2" connectionFactoryRef="IMSCF2"/>  
<connectionFactory id="IMSCF2" connectionManagerRef="IMSTMConnMgr2" containerAuthDataRef="Connection1_Auth" >  
    <properties.gmoa hostName="wg31.washington.ibm.com" portNumber="4000" applicationName="IMSTMPL"/>  
</connectionFactory>  
  
<imsmobile_interaction id="IMSINTER1" imsConnectTimeout="30000"  
                      interactionTimeout="20000" ... />  
<imsmobile_interaction id="IMSINTER2" imsConnectTimeout="20000"  
                      interactionTimeout="15000" ... />
```

The total of *maxPoolSize* in the *connectionManager* configuration elements should not exceed the value of the IMS Connect *MAXSOC* attribute – 1,

- The imsConnectTimeout value is the time the service provider waits for a reply after sending a message to IMS Connect
- The interactionTimeout value is passed to IMS Connect. IMS Connect sends the message to IMS and then waits that long for a reply. If there is none there is a timeout in IMS Connect and IMS Connect sends a timeout to the service provider.



TCP/IP considerations with IMS Connect

On the Liberty TCP/IP environment, ensure:

- **TCPNODELAY=DISABLE**. This allows optimization of transmission but depends on the client environment.
Allows for multiple writes and waits for the buffer to be filled before sending.
- **SO_Linger=Y,VALUE=10** ensures no loss of data. The close of the socket is blocked until ACK is received or 10 seconds, whichever comes first.

In PROFILE.TCPIP configuration on the IMS Connect endpoint, ensure:

- IMS Connect PORT set to NODELAYACKS. This allows ACKS to be sent immediately.
- Specify SHAREPORT, which allows IMS Connect PORTS to be shared by multiple IMS Connect instances on the same stack.
- TCPCONFIG INTERVAL or KEEPALIVEOPTIONS INTERVAL allows TCP/IP to maintain a connection that can be inactive for long periods of time.
- SOMAXCONN must be defined large enough for maximum concurrent connections.

From Redbook *IMS Performance and Tuning Guide*, SG24-7324-00



Connection Management for IMS DB

Use the connectionManagerRef attribute in an IMS ConnectionFactory to provide a connection pool for connections to IMS Connect.

```
<connectionFactory id="DFSIVPACConn" connectionManagerRef="IMSDBConnMgr" >
<properties.imsudbJLocal
    databaseName="DFSIVPA"
    datastoreName="IVP1"
    datastoreServer="wg31.washington.ibm.com"
    driverType="4"
    portNumber="5555"
    user="USER1"
    password="USER1"
    flattenTables="True"/>
</connectionFactory>

<connectionManager id="IMSDBConnMgr" agedTimout=-1 connectionTimeout=30
maxIdleTime=1800 maxPoolSize=50 minPoolSize=0 purgePolicy="EntirePool"
reapTime=180/>
```

The *maxPoolSize* in the *connectionManager* configuration element should not exceed the value of the IMS Connect *MAXSOC* attribute.



Connection Management for MQ

Use the connectionManagerRef attribute in a JMS ConnectionFactory to provide a connection pool for connections to a queue manager.

```
<jmsConnectionFactory id="qmgrCf" jndiName="jms/qmgrCf"  
connectionManagerRef="MQConnMgr">  
  <properties.wmqJMS transportType="CLIENT"  
    queueManager="ZMQ1"  
    channel="LIBERTY.DEF.SVRCONN"  
    hostName="wg31.washington.ibm.com"  
    port="1422" />  
</jmsConnectionFactory>  
  
<connectionManager id="MQConnMgr" agedTimout=-1 connectionTimeout=30  
  maxIdleTIme=1800 maxPoolSize=50 minPoolSize=0 purgePolicy="EntirePool"  
reapTIme=180/>
```

The *maxPoolSize* in the *connectionManager* configuration element should not exceed the value of the *MAXINST* or *MAXINSTC* attributes of the queue manager's server-connection channel.



Connection Management for outbound HTTP request, e.g., Db2, etc.

Outbound connections to Db2, authorization servers, API requesters servers are managed by z/OS Connect code (as is any endpoint configured by the use of a z/OS Connection configuration element).

Connections are managed and/or configured by the use of Java system parameters (-D) *http.maxConnections* and *http.keepAlive*.

- Dhttp.maxConnections=5
- Dhttp.keepAlive=true



Thank you for listening and your questions.

z/OS Connect Wildfire Github Site

<https://ibm.biz/BdPRGD>



The screenshot shows a GitHub repository page with a red box highlighting several files in the file list. The highlighted files are:

- README.md
- WebSphere Liberty on zOS Managi...
- WebSphere Liberty on zOS Configu...
- WebSphere Liberty on zOS Introduc...
- zOS Connect EE V3 Advanced Topic...
- ZOS Connect EE v3 Getting Started....

The GitHub interface includes sections for Releases, Packages, Contributors, Deployments, and Languages. The Languages section shows COBOL at 100.0%.

File List:

File	Action	Last Updated
archive	Add files via upload	last year
containers	Delete containers/readme	9 months ago
misc	Add files via upload	4 months ago
Accessing REST APIs from zOS usin...	Add files via upload	4 days ago
Accessing zOS resources with REST ...	Add files via upload	9 months ago
README.md	Update README.md	3 years ago
WebSphere Liberty on zOS Managi...	Add files via upload	25 minutes ago
WebSphere Liberty on zOS Configu...	Add files via upload	25 minutes ago
WebSphere Liberty on zOS Introduc...	Add files via upload	25 minutes ago
zOS Connect EE V3 Advanced Topic...	Add files via upload	3 years ago
ZOS Connect EE v3 Getting Started....	Add files via upload	3 years ago

README Content:

This repository contains material from the z/OS Connect EE Wildfire workshops run by the IBM Washington Systems Center. It is should be referenced frequently for updates to the presentations, exercises, samples and other material.

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

mitchj@us.ibm.com

Thank you for listening and your questions.

© 2017, 2025 IBM Corporation
Slide 99

Call for survey participation!

A next-generation z/OS experience for the next generation of System Programmers.

What parts of the z/OS experience should be simplified, more intuitive, and user-friendly?

ibm.biz/nextgen-zos-survey

Please forward this survey to your System Programmer colleagues and friends, too!



Miscellaneous Tech-Tips

Important - verify the Java and OMVS environments are ready*

Basic system configuration settings which have more than once caused issues

- Prevent out-of-memory or other storage issues:
 - Verify the Java runtime is not being limited by system parameters, e.g., *MAXASSIZE* (2 147 483 647), *MAXTHREADS*, etc., for details see *BPXPRM setting* at URL https://www.ibm.com/docs/en/sdk-java-technology/8?topic=SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/j9_configure_zos_bpxprm.html
 - Check the value of *ASSIZEMAX* in the OMVS segments of the identities involved and ensure it is adequate, see *MAXASSIZE* above.
 - Exclude OMVS from any IEFUSI exit, SUBSYS(OMVS,NOEXITS) in PARMLIB member *SMFRPMxx*.
 - Verify the JCL MEMLIMIT parameter is within reason for your system.
- Start an OMVS shell session and verify that Java is fully operational by entering command ***java –version***, you see should results like this:

```
java version "1.8.0_301"
Java(TM) SE Runtime Environment (build 8.0.6.35 - pmz6480sr6fp35-20210714_01(SR6 FP35))
IBM J9 VM (build 2.9, JRE 1.8.0 z/OS s390x-64-Bit Compressed References 20210622_7763 (JIT enabled, AOT
enabled)
OpenJ9   - b1f3adb
OMR      - c2f4a18
IBM      - c24a144)
JCL - 20210625_01 based on Oracle jdk8u301-b09
```

- Verify that RACF identities associated with started tasks have OMVS segments with UIDs and GIDs and valid HOME directories and that the identities can invoke Java commands.
- Verify the *zconsetup* script has been executed. My recommendation is to execute this script in the SMP/E target environment, otherwise it will be lost when service is applied and propagated to other images.



Tech/Tip: Sample RACF Commands for SERVER resources

```
RDEFINE SERVER BBG.ANGEL.angelName UACC(NONE) OWNER(SYS1)
PERMIT BBG.ANGEL.angelName CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.SAFCRED UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.SAFCRED CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSWLM UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSWLM CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.TXRRS UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.TXRRS CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSDUMP UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSDUMP CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.SECPFX.BBGZDFLT UACC(NONE)
PERMIT BBG.SECPFX.BBGZDFLT CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.WOLA UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.WOLA CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.LOCALCOM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.LOCALCOM CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSCFM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSCFM CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSCFM.WOLA UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSCFM.WOLA CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.PRODMGR UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.PRODMGR CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSAIO UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSAIO CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
SETROPTS RACLIST(SERVER) REFRESH
```

Tech-Tip: Executing the Liberty *productInfo* command

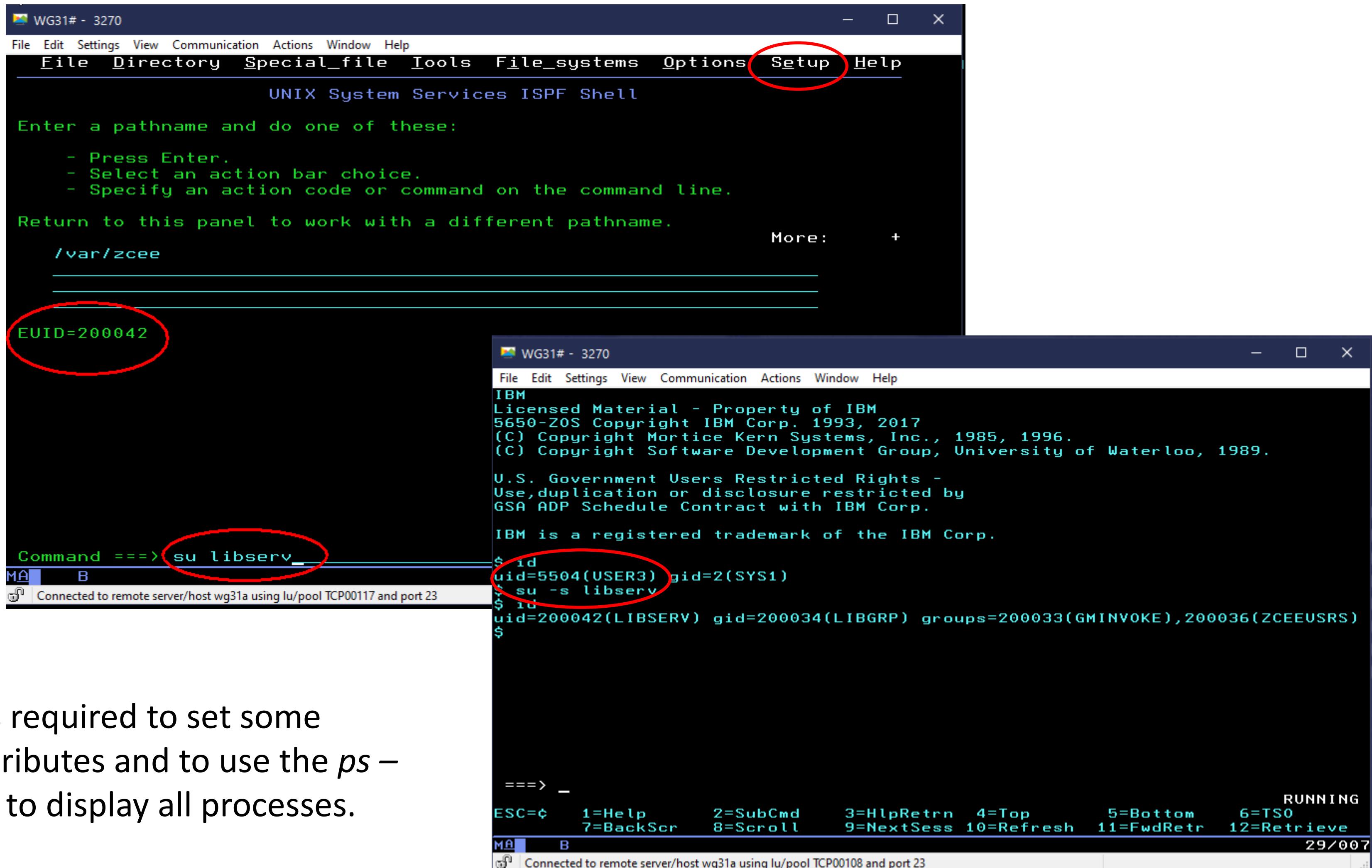
```
//*****
// * SET SYMBOLS
// ****
//EXPORT EXPORT SYMLIST=(*)
// SET WLPDIR='/usr/lpp/IBM/zosconnect/v3r0/wlp'
//PRODINFO EXEC PGM=IKJEFT01,REGION=0M,MEMLIMIT=4G
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
Export WLPDIR=&WLPDIR; +
$WLPDIR/bin/productInfo version; +
$WLPDIR/bin/productInfo featureInfo | grep cics; +
$WLPDIR/bin/productInfo featureInfo | grep mq; +
$WLPDIR/bin/productInfo featureInfo | grep ims; +
$WLPDIR/bin/productInfo validate | grep 'Product validation'
```

```
productInfo featureInfo
productInfo version
productInfo validate
```

Product name: z/OS Connect
 Product version: 03.00.48
 Product edition: z/OS Connect Enterprise Edition

cicsService-1.0 ÿ1.0.0"
 wmqJmsClient-1.1 ÿ1.0.0"
 wmqJmsClient-2.0 ÿ1.0.0"
 Product Extension: mqzosconnect
 mqService-1.0 ÿ1.0.0"
 Product Extension: imsmobile
 imsmobile-2.0 ÿ2.0.0.202108160933"
 Product validation completed successfully.

Tech/Tip: z/OS : ISPF/OMVS examples of using SURROGAT access



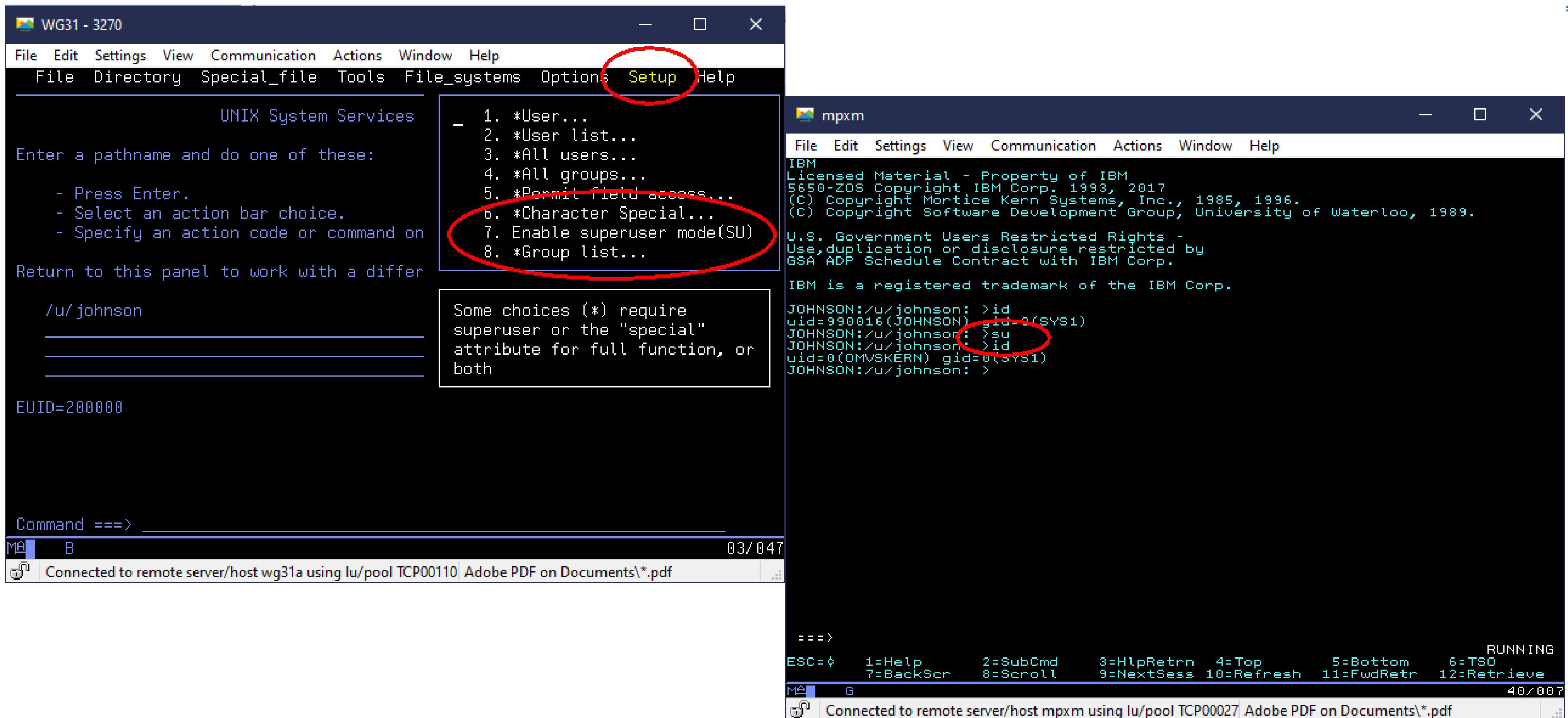
The image contains two side-by-side screenshots of the ISPF/OMVS interface.

Screenshot 1: This screenshot shows the ISPF Shell. The menu bar is visible with options like File, Directory, Special_file, Tools, File_systems, Options, Setup (which is circled in red), and Help. Below the menu, it says "UNIX System Services ISPF Shell". A message on the screen reads: "Enter a pathname and do one of these:" followed by three bullet points: "- Press Enter.", "- Select an action bar choice.", "- Specify an action code or command on the command line.". Below this, another message says "Return to this panel to work with a different pathname." At the bottom, the path "/var/zcee" is shown. In the bottom left corner, the text "EUID=200042" is displayed, also circled in red.

Screenshot 2: This screenshot shows a terminal window with the following text:
 IBM
 Licensed Material - Property of IBM
 5650-ZOS Copyright IBM Corp. 1993, 2017
 (C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
 (C) Copyright Software Development Group, University of Waterloo, 1989.
 U.S. Government Users Restricted Rights -
 Use, duplication or disclosure restricted by
 GSA ADP Schedule Contract with IBM Corp.
 IBM is a registered trademark of the IBM Corp.
 Command ==> su libserv
 MA B
 Connected to remote server/host wg31a using lu/pool TCP00117 and port 23
 \$ id
 uid=5504(USER3) gid=2(SYS1)
 \$ su -s libserv
 \$ id
 uid=200042(LIBSERV) gid=200034(LIBGRP) groups=200033(GMINVOKE),200036(ZCEEUSRS)
 \$
 ==> _
 ESC=< 1=Help 2=SubCmd 3=HlpRetrn 4=Top 5=Bottom 6=TSO
 7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
 MA B
 Connected to remote server/host wg31a using lu/pool TCP00108 and port 23 29/007

Super user is required to set some extended attributes and to use the *ps -ef* command to display all processes.

Tech/Tip: z/OS : Switching to root authority



Tech-Tip: Super user is required to set the program control extended attribute (`extattr +p`) bit for the Java shared object ***ifaedjreg64.so***. This extended attribute must be set for identity assertion in certain situations.



Tech-Tip: Identity assertion and/or JWT generation Extended Attribute Requirement

As root or superuser, set the *libifaedjreg64.so* program control extended attribute bit

- *Permit the server's identity to the required FACILITY resource*

PERMIT BPX.SERVER CLASS(FACILITY) ID(*LIBSERV*) ACCESS(READ)

SETROPTS RACLIST(FACILITY) REFRESH

- *Define a SURROGAT profile for the asserted identity and permit access to connection identity*

RDEFINE SURROGAT *clientID.BAQASSRT* UACC(NONE) OWNER(SYS1)

PERMIT *clientID.BAQASSRT* CLASS(SURROGAT) ACCESS(READ) ID(*zCEEID*)

OR

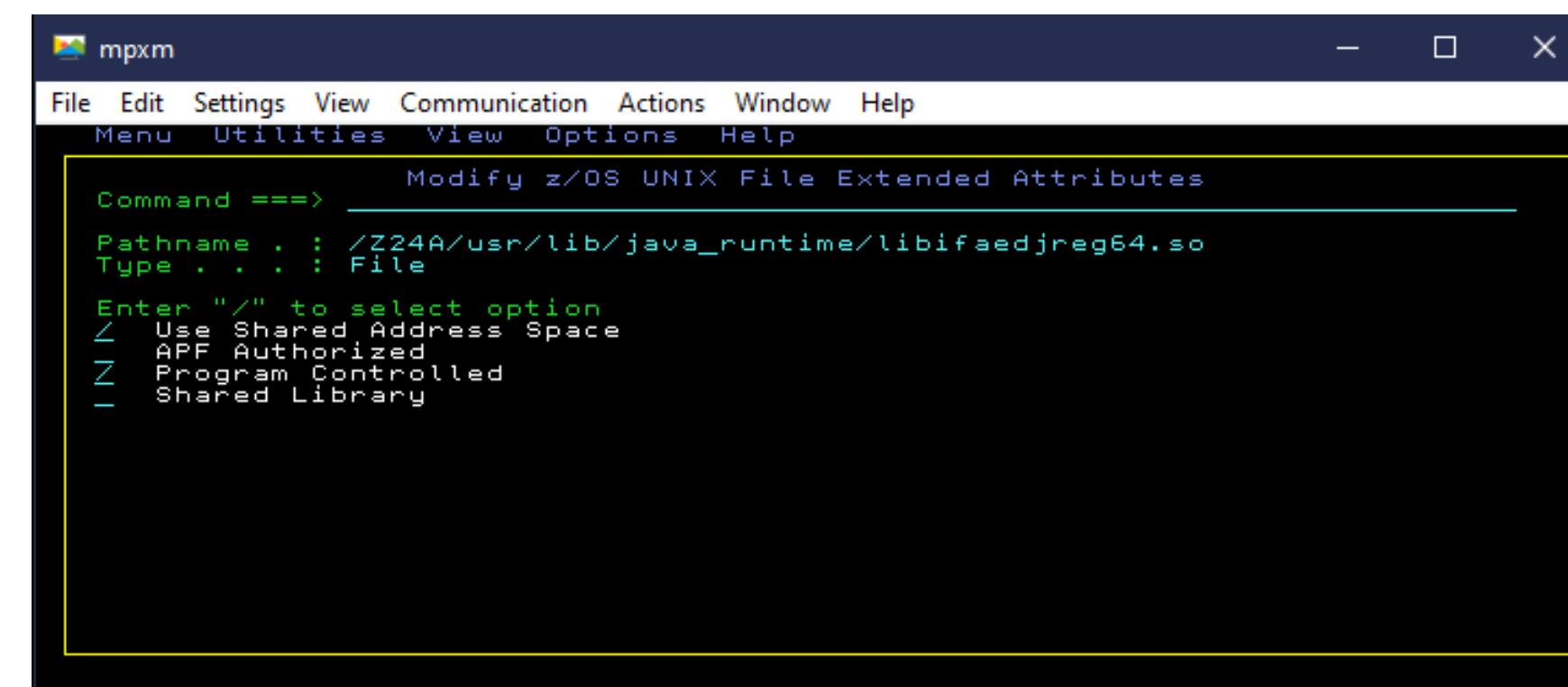
RDEFINE SURROGAT *.BAQASSRT UACC(NONE) OWNER(SYS1)

PERMIT *.BAQASSRT CLASS(SURROGAT) ACCESS(READ) ID(*zCEEID*)

SETROPTS RACLIST(SURROGAT) REFRESH

- *Enable the program control bit for Java shared object ifaedjreg64*

```
su  
cd /usr/lib/java_runtime  
extattr +p libifaedjreg64.so
```





Tech-Tip: Use Symbolic links to simplify commands used in OMVS and JCL

Performing commands:

```
ln -s /global/zosconnect/includes /var/zcee/includes  
ln -s /var/zosconnect/servers/zceesrv1 /var/zcee/zceesrv1  
ln -s /var/zosconnect/servers/zceesrv2 /var/zcee/zceesrv2
```

Will change these OMVS commands from:

```
ln -s /global/zosconnect/includes /var/zosconnect/servers/zceesrv1/includes  
ln -s /global/zosconnect/includes /var/zosconnect/servers/zceesrv2/includes
```

To simpler (and shorter) OMVS commands:

```
ln -s /var/zcee/includes /var/zcee/zceesrv1/includes  
ln -s /var/zcee/includes /var/zcee/zceesrv2/includes
```

Directory Shortcuts

- Create a shortcut from the shared administrative *include* directory to the Sysplex or LPAR shared directory
- Create shortcuts from the server's administrative directories to each server's configuration directory.

N.B. These are symbolic links to symbolic links.

ln -s oldname newname

These symbolic links can be used as JCL symbols

```
//EXPORT EXPORT SYMLIST=(*)
// SET SERVER= 'zceesrv1'
// SET SHARED='/var/zcee/includes'
// SET WLPUSER='var/zosconnect'
//ZCEELN EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
ln -s &SHARED /var/zcee/&SERVER/includes
instead of entering the full directory names as in
ln -s /global/zosconnect/includes +
&WLPUSER/servers/&SERVER/includes
```

And added as exports to */u/home/.profile* or */etc/profile* files

```
export serverName=zceesrv1
export shared=/var/zcee/includes
export WLP_USER_DIR=/var/zosconnect
```

```
//ZCEELN EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *
BPXBATCH SH +
ln -s $shared /var/zcee/$serverName/includes
instead of entering the full directory names as in
ln -s /global/zosconnect/includes +
$WLPUSER/servers/$serverName/includes
```



Tech-Tip: Copying WOLA executables from OMVS to a PDSE

```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=JOHNSON,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)  
//*****  
///* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET DSNAME='USER1.WOLA2106.LOADLIB'  
// SET ZCEEPATH='/usr/lpp/IBM/zosconnect/v3r0'  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
//*****  
///* Step ALLOC - Allocate a PDSE load library  
//*****  
//ALLOC EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *,SYMBOLS=EXECSYS  
DELETE '&DSNAME'  
SET MAXCC=0  
ALLOC DSNAME('&DSNAME') -  
    NEW CATALOG SPACE(2,1) DSORG(PO) CYLINDERS -  
    RECFM(U) DSNTYPE(LIBRARY)  
//*****  
///* Step WOLACOPY - copy the WOLA executables to the PDSE  
//*****  
//WOLACOPY EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
    export JAVA_HOME=&JAVAHOME; +  
    export DSNAME=&DSNAME; +  
    cp -Xv &ZCEEPATH/wlp/clients/zos/* "/* '$DSNAME'"
```



Tech-Tip: Sample JCL - Executing the Liberty *securityUtility* command

```
//*****  
/* Use securityUtility to encrypt a password using an  
/* encryption key of a certificate  
//*****  
//IKJEFT01 EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *  
BPXBATCH SH +  
/usr/lpp/IBM/zosconnect/v3r0/wlp/bin/securityUtility encode +  
--encoding=aes +  
--keyring=safkeyring://JOHNSON/Liberty.KeyRing +  
--keyringType=JCERACFKS --keyLabel="Johnson Client Cert" +  
passwordToEncrypt
```

```
<featureManager>  
  <feature>zosPasswordEncryptionKey-1.0</feature>  
</featureManager>  
  
<zosPasswordEncryptionKey  
keyring="safkeyring://JOHNSON/Liberty.KeyRing"  
label="Johnson Client Cert" type="JCERACFKS"/>
```

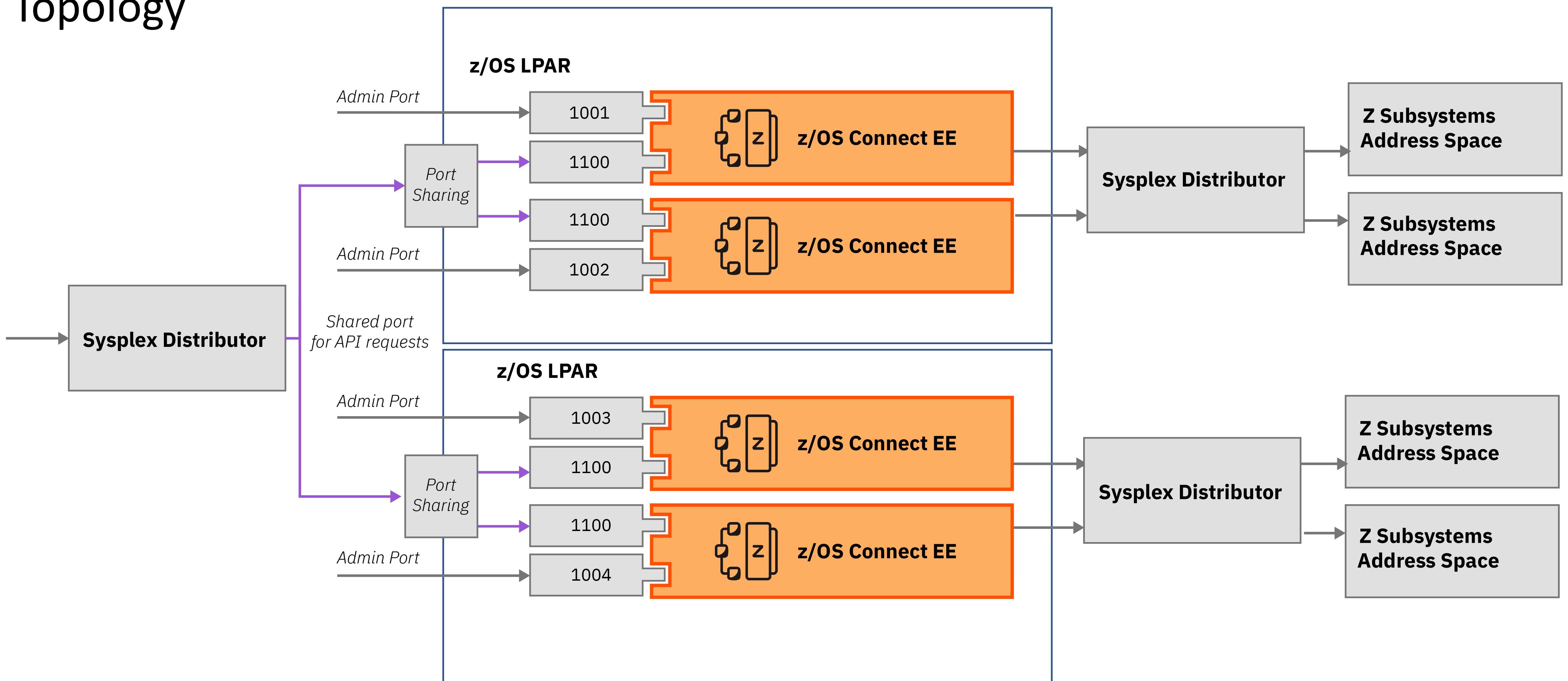
```
//*****  
/* Use securityUtility to encrypt a password using an  
/* encryption key string  
//*****  
//IKJEFT01 EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *  
BPXBATCH SH +  
/usr/lpp/IBM/zosconnect/v3r0/wlp/bin/securityUtility encode +  
--encoding=aes -key myEncryptionKey +  
passwordToEncrypt
```

```
wlp.password.encryption.key=myEncryptionKey
```



High Availability

- Topology



ibm.biz/zosconnect-ha-concepts

ibm.biz/zosconnect-scenarios

Tech-Tip: Sample Sysplex DVIPAs Configuration

SYS1.TCPIP.TCPPARMS (IPNODES)

```
192.168.17.241 MPZ1.DMZ MPZ1 mpz1.washington.ibm.com
192.168.17.242 MPZ2.DMZ MPZ2 mpz2.washington.ibm.com
192.168.17.243 MPZ3.DMZ MPZ3 mpz3.washington.ibm.com
192.168.17.240 dvipa dvipa.washington.ibm.com
```

SYS1.TCPIP.TCPPARMS (PROFMPZ3)

```
IPCONFIG SYSPLEXROUTING
DYNAMICXCF 172.1.1.243 255.255.255.0 3
VIPADYNAMIC
VIPADEFINE 255.255.255.0 192.168.17.240
VIPADISTRIBUTE DEFINE DISTM ROUNDROBIN|BASEWLM 192.168.17.240
PORT 23 1416 1491 2446 9443 9453 9463
DESTIP
172.1.1.241
172.1.1.242
172.1.1.243
ENDVIPADYNAMIC
```

SERVERWLM is not an option

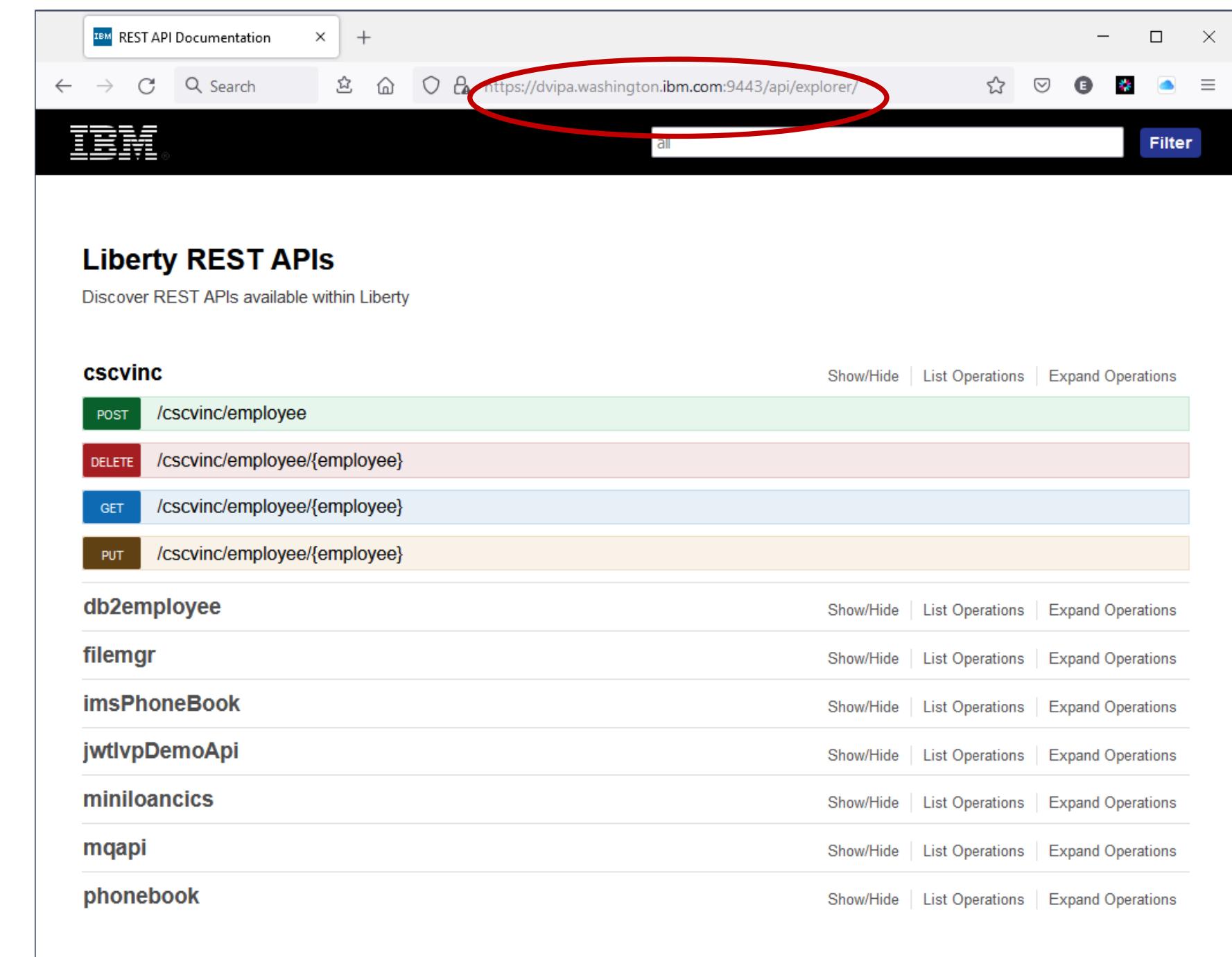
HOMETEST

```
EZA0619I Running IBM MVS TCP/IP CS V2R4 TCP/IP Configuration Tester
EZA0602I TCP Host Name is: MPZ3

EZA0605I Using Name Server to Resolve MPZ3
EZA0611I The following IP addresses correspond to TCP Host Name: MPZ3
EZA0612I 192.168.17.243
EZA0614I The following IP addresses are the HOME IP addresses defined in PROFILE.TCPIP:
EZA0615I 192.168.17.243
EZA0615I 172.1.1.243
EZA0615I 192.168.17.240
EZA0615I 127.0.0.1

EZA0618I All IP addresses for MPZ3 are in the HOME list!
EZA0622I Hometest was successful - all Tests Passed!
```

```
<zosconnect_cicsIpicConnection id="cscvinc"
    host="dvipa.washington.ibm.com"
    port="1491"/>
<zosconnect_endpointConnection id="mqapi"
    host="http://dvipa.washington.ibm.com"
    port="9453"
    basicAuthRef="myBasicAuth"
    connectionTimeout="10s"
    receiveTimeout="20s" />
```



cscvinc	
POST	/cscvinc/employee
DELETE	/cscvinc/employee/{employee}
GET	/cscvinc/employee/{employee}
PUT	/cscvinc/employee/{employee}
db2employee	
filemgr	
imsPhoneBook	
jwtlvpDemoApi	
miniloancics	
mqapi	
phonebook	

Tech/Tip: Use the TCPIP resolver trace to display name resolution information

```

ALLOC FILE(SYSTCPT) DA(*)
ping wg31.washington.ibm.com
Resolver Trace Initialization Complete -> 2021/09/12 12:54:37.36

res_init Resolver values:
Setup file warning messages = No
CTRACE TRACERES option = No
Global Tcp/Ip Dataset = SYS1.TCPIP.TCPPARMS(TCPDAT3)
Default Tcp/Ip Dataset = SYS1.TCPIP.TCPPARMS(TCPDAT3)
Local Tcp/Ip Dataset = //DD:SYSTCPD
                         ==> SYS1.TCPIP.TCPPARMS(TCPDAT3)
Translation Table = SYS1.TCPIP.STANDARD.TCPXLBIN
UserId/JobName = JOHNSON
Caller API = TCP/IP Sockets Extended
Caller Mode = EBCDIC
System Name = WSC13 (from VMCF)
UnresponsiveThreshold = 25
(G) DataSetPrefix = SYS1.TCPIP
(G) HostName = wg31
. . .
res_query Failed: RetVal = -1, RC = 1, Reason = 0x78981005
res_querydomain Failed: RetVal = -1, RC = 1, Reason = 0x78981005
res_search Failed: RetVal = -1, RC = 1, Reason = 0x78981005
GetAddrInfo Closing IOCTL Socket 0x00000000
BPX1CLO: RetVal = 0, RC = 0, Reason = 0x00000000
GetAddrInfo Failed: RetVal = -1, RC = 1, Reason = 0x78AE1004
GetAddrInfo Ended: 2021/09/12 12:55:32.364732
*****
EZ2311I Unknown host 'WG31.WASHINGTON.IBM.COM'

```

Root cause – Host wg31.washington.ibm.com was missing from SYS1.TCPIP.TCPPARMS(IPNODES)



Tech/Tip: Using a cURL trace to show the flow with mutual authentication

```
* successfully set certificate verify locations:  
* TLSv1.3 (OUT), TLS handshake, Client hello (01):  
* TLSv1.3 (IN), TLS handshake, Server hello (02):  
* TLSv1.2 (IN), TLS handshake, Certificate (11):  
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):  
* TLSv1.2 (IN), TLS handshake, Request CERT (13):  
* TLSv1.2 (IN), TLS handshake, Server finished (14):  
* TLSv1.2 (OUT), TLS handshake, Certificate (11):  
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):  
* TLSv1.2 (OUT), TLS handshake, CERT verify (15):  
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (01):  
* TLSv1.2 (OUT), TLS handshake, Finished (20):  
* TLSv1.2 (IN), TLS handshake, Finished (20):  
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384  
* Server certificate:  
* subject: O=IBM; OU=LIBERTY; CN=wg31.washington.ibm.com  
* start date: Jan 4 04:00:00 2021 GMT  
* expire date: Jan 1 03:59:59 2023 GMT  
* common name: wg31.washington.ibm.com (matched)  
* issuer: OU=LIBERTY; CN=CA for Liberty  
* SSL certificate verify ok.
```

```
enum {  
    hello_request(0),  
    client_hello(1),  
    server_hello(2),  
    certificate(11),  
    server_key_exchange (12),  
    certificate_request(13),  
    server_hello_done(14),  
    certificate_verify(15),  
    client_key_exchange(16),  
    finished(20),  
    (255) }  
HandshakeType;
```

```
* TLS 1.2 https://tools.ietf.org/html/rfc5246  
TLS 1.3 https://tools.ietf.org/html/rfc8446
```



Tech-Tip: CICS IPCONN and TCPIPSERVICE resources for HA

CICS Specific TCPIPSERVICE - IPIC

```
TCpipservice : IPIC1
GROup       : SYSPGRP
Urm          ==> DFHISAIP
POrtnumber   ==> 01492
SStatus      ==> Open
PROtocol     ==> IPic
TRansaction  ==> CISS
Host         ==> ANY
Ipaddress    ==> ANY
SPeciftcps  ==>
```

CICS Generic TCPIPSERVICE - IPICG

```
TCpipservice : IPICG1
GROup       : SYSPGRP
Urm          ==> DFHISAIP
POrtnumber   ==> 01491
SStatus      ==> Open
PROtocol     ==> IPic
TRansaction  ==> CISS
Host         ==> ANY
Ipaddress    ==> ANY
SPeciftcps  ==> IPIC
```

A client connects first to the CICS region's generic port (1491) and then the CICS region redirects the client to the region's specific port (1492).

I IPCONN ACQ

```
STATUS: RESULTS - OVERTYPE TO MODIFY
Ipc(BAQSVR1 ) App(BAQSVR1) Net(BAQSVR1) Ins Acq Nos
        Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR1C) App(BAQSVR1C) Net(BAQSVR1C) Ins Acq Nos
        Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR1M) App(BAQSVR1M) Net(BAQSVR1M) Ins Acq Nos
        Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR2 ) App(BAQSVR2) Net(BAQSVR2) Ins Acq Nos
        Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR2C) App(BAQSVR2C) Net(BAQSVR2C) Ins Acq Nos
        Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR2M) App(BAQSVR2M) Net(BAQSVR2M) Ins Acq Nos
        Rece(001) Sen(000) Tcp(IPIC)
```

Number of
IPCONN resources
equals the number
of zCEE server
times the number of
unique connection
references

¹CICS requires the specific TCPIPSERVICE be installed before the corresponding generic TCPIPSERVICE resource. TCPIPServices are installed in alphabetically order, so the name of specific service must be alphabetically prior to the name of the generic TCPIPSERVICE.



Tech-Tip: CICS IPIC connection processing for high availability load balancing*

If the *reconnectInterval* attribute is set, at the specified time interval, a check is made to see if a new connection attempt should be attempted

A new connection is established if the current connection properties are not the preferred connection properties:

- If *reconnectInterval*, *preferredSpecificHost* and *preferredSpecificPort* are not set,
 - New connection attempts are disabled (this is the default behavior).
- If *reconnectInterval* is set and *preferredSpecificHost* and *preferredSpecificPort* are not set,
 - A new connection is attempted at the interval specified by the *reconnectInterval* time. Use this to enable regular connection rebalancing.
- If *reconnectInterval* and *preferredSpecificPort* are set and *preferredSpecificHost* is not set,
 - A new connection is attempted at the expiration time interval and if the current connected port in use does not match the preferred port
 - Relevant when shared port is for a single LPAR
 - Specific CICS region is preferred
- If *reconnectInterval* and *preferredSpecificHost* are set and *preferredSpecificPort* is not set
 - A new connection is attempted at the expiration time interval and if the current host in use does not match the preferred port
 - Relevant when shared port is across Sysplex
 - Any CICS region on a specific LPAR is preferred
- If *reconnectInterval*, *preferredSpecificHost* and *preferredSpecificPort* are all set
 - A new connection is attempted at the expiration time interval time and if both the current host and port in use do not match the preferred host and port
 - Relevant when shared port is on a single LPAR or across a Sysplex
 - Specific CICS region is preferred.

When the reconnection attempt results in a new connection to a CICS region, new requests are sent over the new connection. Previous connections will continue and when all requests have completed processing, the previous or old connection will be closed.



Tech-Tip: Executing the z/OS Connect *zconsetup* script using JCL

```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=JOHNSON,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)  
//*****  
//* Set symbols  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
// SET ZCEEPATH='/usr/lpp/IBM/zosconnect/v3r0'  
//*****  
//* Step ZCSETUP - Invoke the zconsetup script  
//*****  
//ZCSETUP EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
  export JAVA_HOME=&JAVAHOME; +  
  &ZCEEPATH/bin/zconsetup install
```

Tech-Tip: Differences between z/OS Connect OpenAPI2 and OpenAPI3 server.xml files



```
default template - OpenAPI 2 server.xml configuration file
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:zosConnect-2.0</feature>
        <feature>zosconnect:zosConnectCommands-1.0</feature>
        <feature>apiDiscovery-1.0</feature> *
    </featureManager>

    <!-- To access this server from a remote client add a host attribute
    <httpEndpoint id="defaultHttpEndpoint"
        host="*"
        httpPort="9080"
        httpsPort="9443" />
    <!-- add cors to allow cross origin access, e.g. when using swagger UI
    to fetch swagger doc from zOS Connect Enterprise Edition -->
    <cors id="defaultCORSConfig"
    - - - - - 24 Line(s) not Displayed

    <!-- config requires updateTrigger="mbean" for REFRESH command support
-->
<config updateTrigger="mbean" monitorInterval="500"/>

    <zosconnect_zosConnectManager setUTF8ResponseEncoding="true"/>

    <!-- zosConnect APIs -->
    <zosconnect_zosConnectAPIs updateTrigger="disabled" pollingRate="5s"
        <!-- zosConnect Services -->
    <zosconnect_services updateTrigger="disabled" pollingRate="5s"/>

    <!-- applicationMonitor is not applicable for z/OS Connect EE servers --
->
    <applicationMonitor updateTrigger="disabled" dropinsEnabled="false"/>

</server>
```

```
openApi3 template - OpenAPI 3 server.xml configuration file
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:zosConnect-3.0</feature>
        <feature>openapi-3.0</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute
    <httpEndpoint id="defaultHttpEndpoint"
        host="*"
        httpPort="9080"
        httpsPort="9443" />
    - - - - - 12 Line(s) not Displayed
        <!-- config requires updateTrigger="mbean" for REFRESH command support
        config updateTrigger="mbean"/>

        <!-- applicationMonitor requires updateTrigger="mbean" for REFRESH command
        support -->
        <applicationMonitor updateTrigger="mbean" dropinsEnabled="false"/>

        <!-- Automatic expansion of WAR files is required for z/OS Connect native
        servers running the zosConnect-3.0 feature -->
        <applicationManager autoExpand="true" />

        <!-- APIs are deployed as WAR files and a webApplication element must be
        used to specify the location of the API WAR and optionally the name of the API
        -->
        <webApplication id="My API" location="${server.config.dir}/apps/api.war"
            name="MyAPI"/>

    </server>
```

Note there are no *zosconnect* or *cors* configuration elements present with Open API 3.

Tech-TIP: Contrast a Liberty JCL procedure versus a z/OS Connect JCL procedure

```
//ZCEESRVR PROC PARMs='serverName'
//*
// SET ZCONHOME='/usr/lpp/IBM/zosconnect/v3r0'
// SET INSTDIR='/usr/lpp/liberty_zos/21.0.0.9'
//*
//ZCON      EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=8G,
//  PARM='PGM &ZCONHOME./bin/zosconnect run &PARMS. --clean'
//  PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv &PARMS'
//STDOUT    DD   SYSOUT=*
//STDERR    DD   SYSOUT=*
//STDIN     DD   DUMMY
//MSGLOG    DD   SYSOUT=*
//STDENV    DD   *
_BPX_SHAREAS=YES
_CEE_RUNOPTS=HEAPPOOLS (ON) ,HEAPPOOLS64 (ON)
JAVA_HOME=/usr/lpp/java/J8.0_64
WLP_USER_DIR=/var/zosconnect
JVM_OPTIONS=-Dcom.ibm.ws.zos.core.angelName=zCEEAngel -Xmx512m
OPENJ9_JAVA_OPTIONS=-Xoptionsfile=/var/zcee/properties/myServer.property
```

OMVS
LE
JAVA
LIBERTY
z/OS Connect

Tech-Tip: Executing Gradle build commands in JCL



```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=&SYSUID,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)  
//*****  
//* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
// SET GRADLSRC='/u/johnson/gradle'  
// SET GRADLE='/usr/lpp/gradle/gradle-7.6.1'  
//*****  
//* Step GRADLE - Invoke the gradle build command  
//*****  
//CSCVINC EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
export JAVA_HOME=&JAVAHOME; +  
cd &GRADLSRC./cscvinc; +  
&GRADLE./bin/gradle build -i
```

settings.gradle

```
pluginManagement {  
    repositories {  
        maven {  
            url '/u/johnson/gradle/gradleLibs'  
        }  
    }  
}
```

build.gradle

```
plugins {  
    id 'com.ibm.zosconnect.requester' version '1.1.7'  
}
```

This assumes the z/OS Connect provided *dependencies.zip* files was expanded into directory */u/johnson/gradle/gradleLibs* using command *jar -tf dependencies.zip* and that the gradle files *settings.gradle* and *build.gradle* are encoded in ASCII in directory */u/johnson/gradle/cscvinc*

Tech-Tip: - Executing the z/OS Connect Build Toolkit in JCL



```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=&SYSUID,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)  
//*****  
//* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET WORKDIR='/u/johnson/zconbt'  
// SET ZCONDIR='/usr/lpp/IBM/zosconnect/v3r0/zconbt/bin'  
//ZCONBT EXEC PGM=IKJEFT01,REGION=0M,MEMLIMIT=4G  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
  export WORKDIR=&WORKDIR; +  
  export ZCONDIR=&ZCONDIR; +  
  cd $WORKDIR; +  
  $ZCONDIR/zconbt.zos -p cscvinc.properties -f=cscvinc.ara; +  
  cp -v $WORKDIR/syslib/* //'JOHNSON.ZCONBT.COPYLIB'"
```

cscvinc.properties

```
apiDescriptionFile=./cscvinc.json  
dataStructuresLocation=./syslib  
apiInfoFileLocation=./syslib  
logFileDirectory=./logs  
language=COBOL  
connectionRef=cscvincAPI  
requesterPrefix=csc
```

This assumes the zconbt.zip files was expanded into directory /usr/lpp/IBM/zosconnect/v3r0/zconbt using command *jar -tf zconbt.zip* and that the property file and Swagger JSON document are encoded in ASCII in directory /u/johnson/zconbt.

Tech-Tip: Executing multiple OMVS commands in one step



```
*****  
/* SET SYMBOLS  
*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET CURL= '/usr/lpp/rocket/curl'  
*****  
/* CURL Procedure  
*****  
//CURL PROC  
//CURL EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
// PEND  
*****  
/* STEP CURL - use curl to deploy API cscvinc  
*****  
//DEPLOY EXEC CURL  
BPXBATCH SH export CURL=&CURL; +  
$CURL/bin/curl -X PUT -s +  
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +  
https://wg31.washington.ibm.com:9445/zosConnect/apis/cscvinc?status=stop+  
pped > null; +  
$CURL/bin/curl -X DELETE -s +  
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +  
https://wg31.washington.ibm.com:9445/zosConnect/apis/cscvinc > null; +  
$CURL/bin/curl -X POST -s +  
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +  
--data-binary @/u/johnson/cscvinc.aar +  
--header "Content-Type: application/zip" +  
https://wg31.washington.ibm.com:9445/zosConnect/apis  
*****  
/* STEP CURL - use curl to invoke the API cscvinc  
*****  
//INVOKE EXEC CURL  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH export CURL=&CURL; $CURL/bin/curl -X GET -s +  
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +  
https://wg31.washington.ibm.com:9445/cscvinc/employee/000100
```

Always be aware of the beginning and trailing spaces.

<https://www.rocketsoftware.com/platforms/ibm-z/curl-for-zos>

Tech-Tip: Using ADRDSSU to dump/restore MVS data sets

ZCEEDUMP

```
//EXPORT EXEC SYMLIST=(*)
// SET ZCEELVL=349
//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *,SYMBOLS=EXECSYS
      DELETE IBM.ZCEE30.BKUP&ZCEELVL.
      SET MAXCC=0
//DUMP      EXEC PGM=ADRDSSU,REGION=2048K
//SYSPRINT DD SYSOUT=*
//DUMPDD DD DSN=IBM.ZCEE30.BKUP&ZCEELVL.,
//          DISP=(NEW,CATLG),
//          UNIT=SYSDA,SPACE=(CYL,(3000,2000,0),RLSE)
//SYSIN    DD *,SYMBOLS=EXECSYS
      DUMP DATASET(INCLUDE(
      ZCEE30.SBAQ* -
      ZCEE30.WOLA*.** -
      OMVS.ZCEE*.** -
      )) OPTIMIZE(4) OUTDDNAME(DUMPDD) TOLERATE(ENQF)
```

ZCEERSTR

```
//RESTORE EXEC PGM=ADRDSSU,REGION=2048K
//SYSPRINT DD SYSOUT=*
//DUMPDD DD DISP=SHR,DSN=JOHNSON.ZCEE30.BKUP349
//SYSIN    DD *
      RESTORE DATASET(INCLUDE(**)) -
      INDDNAME(DUMPDD) OUTDYNAM(WAS004) -
      NULLSTORCLAS -
      REPLACE CATALOG TOLERATE(ENQF)
```

Tech-Tip: Define and format a ZFS data set using IOEAGFMT

ZFS

```
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYSIN    DD *
      SET MAXCC=0
      DEFINE CLUSTER (NAME(OMVS.ZCEE.GROUP1.ZFS) -
                      LINEAR CYLINDERS(100 100) SHAREOPTIONS(3))
//CREATE EXEC PGM=IOEAGFMT,REGION=0M,
//  PARM=( '-aggregate OMVS.ZCEE.GROUP1.ZFS -compat' )
//SYSPRINT DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
```