z/OSMF 2.1 Advanced Programming

Joey Zhu (zhuxiaoz@cn.ibm.com)
IBM Corporation

March 4, 2015
Session Number 16935

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*	ServerPac*	* Registered trademarks of IBM Corporation
IBM (logo)	WebSphere*	
RACF*	z/OS*	

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Firefox is a trademark of Mozilla Foundation
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Internet Explorer is a trademark of Microsoft Corp
InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

See url http://www.ibm.com/legal/copytrade.shtml for a list of IBM trademarks.

Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Topology service**
  - Multisystem routing service**
  - z/OSMF workflow service**
  - Software Management service**
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler
- Summary

** available for z/OSMF V2R1 with APAR PI32148

What is z/OSMF

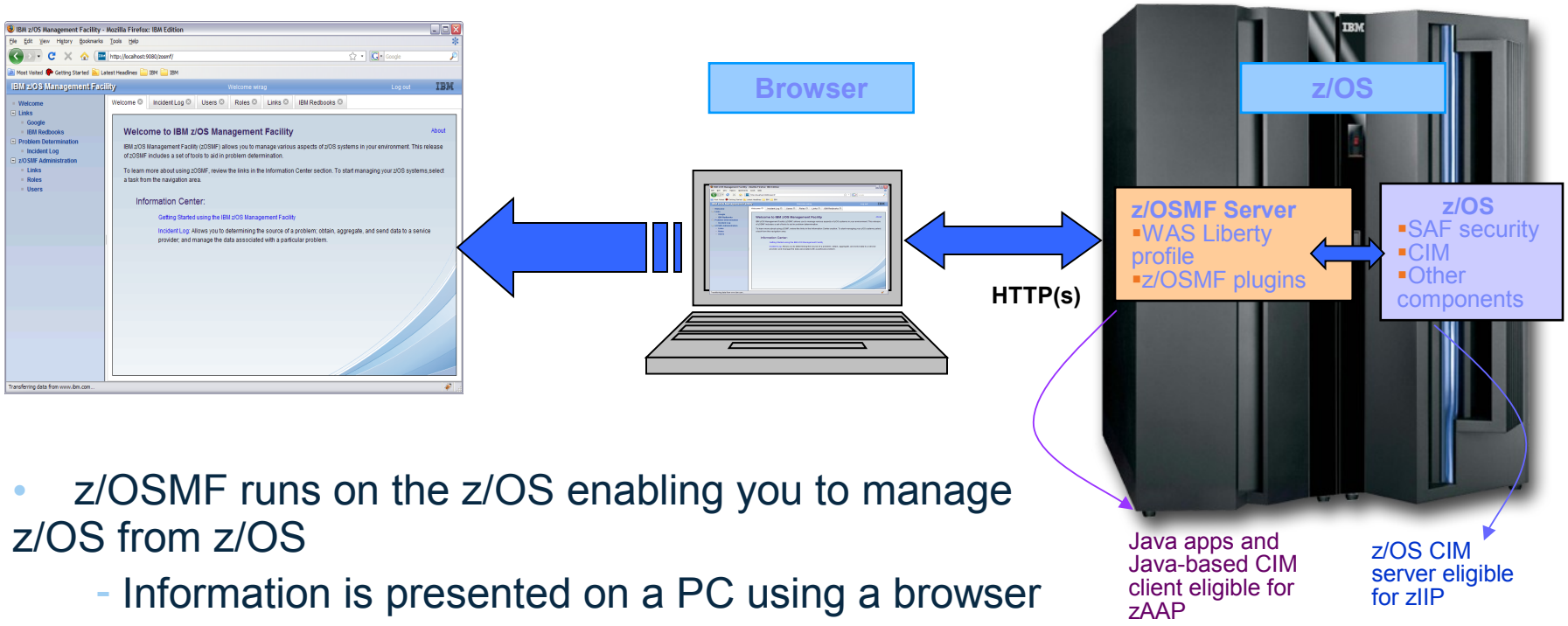
- IBM z/OS Management Facility (z/OSMF) delivers on IBM's strategy for mainframe simplification and modernization
  - The IBM z/OS Management Facility is a separate product for z/OS that provides support for a modern, Web-browser based management console for z/OS.
  - It helps system programmers more easily manage and administer a mainframe system by simplifying day to day operations and administration of a z/OS system.
  - More than just a graphical user interface, the z/OS Management Facility is intelligent, addressing the needs of a diversified skilled workforce and maximizing their productivity.
    - Automated tasks can help reduce the learning curve and improve productivity.
    - Embedded active user assistance (such as wizards) guide you through tasks and helps provide simplified operations.

- z/OSMF has a zero price for z/OS customers

- z/OSMF has its own product number and must be ordered separately

What is z/OSMF

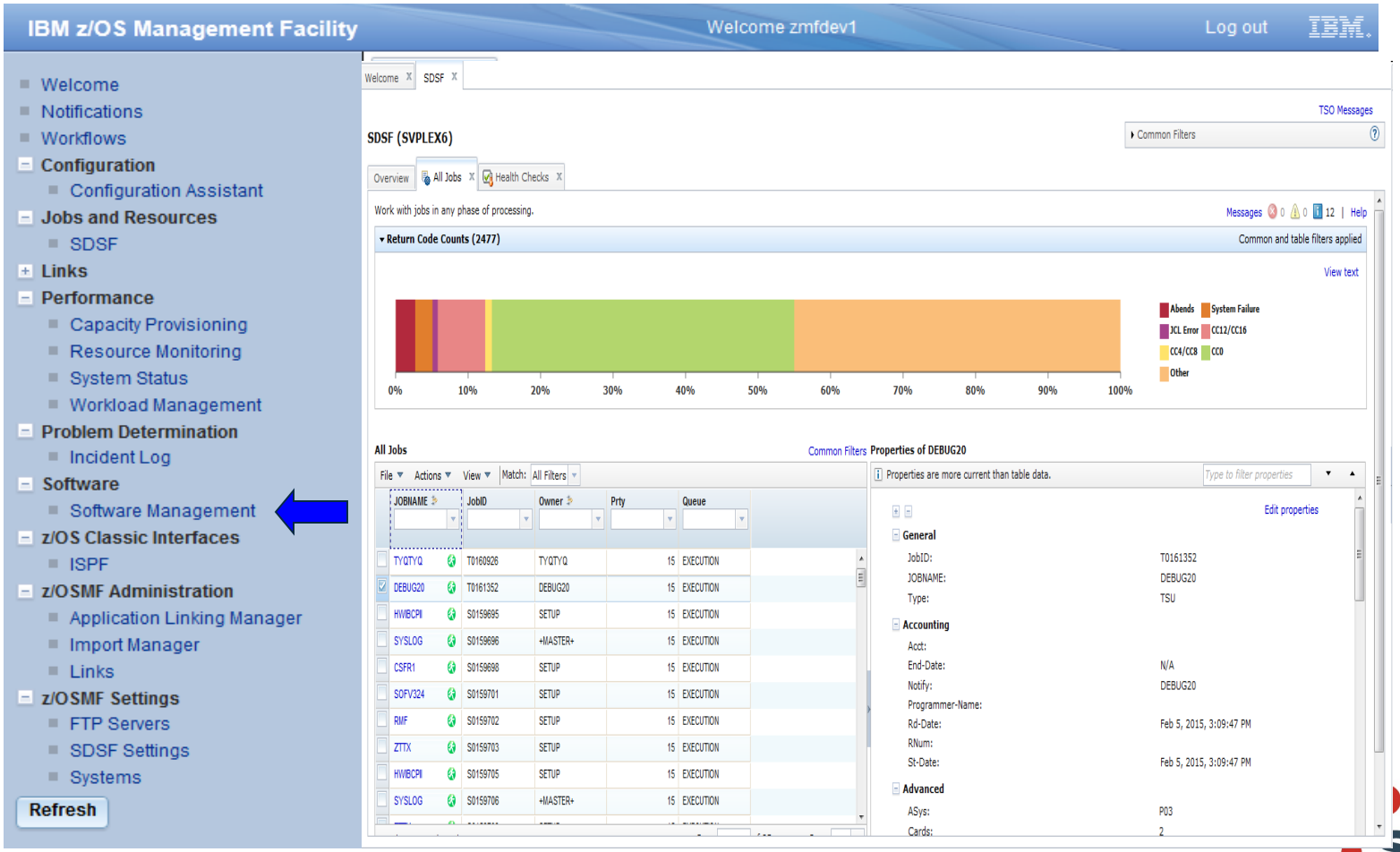
- How does z/OSMF function in the z/OS environment?



- z/OSMF runs on the z/OS enabling you to manage z/OS from z/OS
  - Information is presented on a PC using a browser
  - No client install required

What is z/OSMF

- Gain simplification and modernization through z/OSMF plugins



What is z/OSMF

What's more?

Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming 
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Topology service**
  - Multisystem routing service**
  - z/OSMF workflow service**
  - Software Management service**
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler
- Summary

** available for z/OSMF V2R1 with APAR PI32148

What is z/OSMF Advanced Programming

- In addition to z/OSMF plugins with modern UI and simplified task, z/OSMF also provides services and facilities to help you write programs.

- z/OSMF Advanced Programming includes:
  - Using z/OSMF RESTful services
  - Develop workflow
  - Create your own z/OSMF plugins

- This session focus on “Using z/OSMF RESTful services”. Please refer to <IBM z/OS Management Facility Programming Guide>* for more details about “Develop workflow” and “Create your own z/OSMF plugins”.

* The latest order number of V2R1 is “SA32-1066-02”

What is z/OSMF Advanced Programming

- Why would I need “z/OSMF RESTful services”
  - Representational State Transfer (REST) is a software architecture style. RESTful web service:
    - Has gained widespread acceptance across the Web. (Amazon, Twitter)
    - Is easy to call
    - Could be driven remotely (via HTTPS) and securely.
    - Language and platform independent.
      - Mobile (iOS and Android) platform also has strong support for calling RESTful web service.
  - z/OSMF RESTful services makes z/OS and z/OSMF more approachable.

What is z/OSMF Advanced Programming

- Why would I need to “develop workflow”
  - z/OSMF Workflows application provides a framework supports user (workflow provider) to define a guided flow (a.k.a. workflow) through steps to accomplish a task in z/OS.
  - z/OSMF Workflows application is useful to:
    - Assist people unfamiliar with how to perform a given task, or a task that they perform rarely
    - Ensure that all tasks are performed in the right order and only when their dependencies have been met
    - Monitor and track progress toward the completion of the task in the centralized place
    - Provide a history (audit trail) of the steps performed for a task
    - Automate a workflow without the need of interaction
    - Adjust step status dynamically according to various pre-defined condtion
    - Work with workflow through REST service instead of having to logon to z/OSMF UI**.
  - Exploiters of workflow
    - zEDC workflow to enable zEDC on multiple systems
    - z/OS Migration workflow
    - z/OSMF Configure workflow
    - z/OSMF CA workflows

** available for z/OSMF V2R1 with APAR PI32148

What is z/OSMF Advanced Programming

- Why would I need to “create my own z/OSMF plugins”
  - z/OSMF V2R1 provides “External application support” solution enables non-z/OSMF web application (plugin) – html and/or JavaScript applications to render their UI and run in the z/OSMF browser.
  - “External application support” solution provides:
    - z/OS TSO/E address space RESTful service allows “external application” to communicate with their back-end TSO/E application which facilitates reusing existing TSO application to serve web application.
    - Data persistence RESTful service helps “external application” to manage (read/write) their persistent data in z/OS side.
    - Import Manager plugin provides the UI interface to import “external application” into z/OSMF.
  - “Create your own z/OSMF plugins” builds centralized z/OS management portal and makes developing and deploying web application more easy.

Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services 
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Topology service**
  - Multisystem routing service**
  - z/OSMF workflow service**
  - Software Management service**
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler
- Summary

** available for z/OSMF V2R1 with APAR PI32148

Using z/OSMF RESTful services

- z/OSMF provides a set of RESTful services to make z/OS and z/OSMF more approachable.
- Process overview
  - z/OSMF RESTful services can be invoked by any HTTP client application running on the z/OS local system or remote system.
  - Your application (client) issues HTTP request to the target system (z/OS).
  - z/OSMF determines if the request is valid or not. If it's valid, z/OSMF performs the requested service and return the HTTP response contains the result.
- Security consideration
  - Your client program must authenticate to z/OSMF in order to access z/OSMF RESTful services. (Certificate authentication is supported on "zOS Jobs RESTful service"**)
  - Other authorization may be required for different RESTful services. Generally, your user ID requires the same authorization when performing similar operation from traditional channel such as TSO/E session. E.g. Submitting a job through "z/OS jobs RESTful service" requires that your user ID be authorized to run jobs on the system.
  - Please refer to <z/OS Management Facility Programming Guide> for details.

** available for z/OSMF V2R1 with APAR PI32148

Using z/OSMF RESTful services

- z/OSMF provides below RESTful services:
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Topology service**
  - Multisystem routing service**
  - z/OSMF workflow service**
  - Software Management service**

** available for z/OSMF V2R1 with APAR PI32148

z/OS jobs service – API list

- z/OS jobs RESTful service is provided for working with batch jobs on a z/OS system.
- z/OS jobs RESTful service provides below operations (APIs):

Operation	Method
Obtain the status of a job.	GET
List the jobs for an owner, prefix, or job ID.	GET
List the spool files for a job.	GET
Retrieve the contents of a job spool file.	GET
Submit a job to run on z/OS.	PUT
Hold a job.	PUT
Release a job.	PUT
Change the job class.	PUT
Cancel a job.	PUT
Delete a job (cancel a job and purge its output).	DELETE

z/OS jobs service – API example

- List the jobs for an owner, prefix or job ID
  - URL format

  There are various formats of the URL for this operation:

  https://host:port/zosmf/restjobs/jobs
  https://host:port/zosmf/restjobs/jobs?owner=owner
  https://host:port/zosmf/restjobs/jobs?prefix=prefix*
  https://host:port/zosmf/restjobs/jobs?owner=owner&prefix=prefix*
  https://host:port/zosmf/restjobs/jobs?jobid=jobid
  https://host:port/zosmf/restjobs/jobs?max-jobs=nnn
  https://host:port/zosmf/restjobs/jobs?user-correlator=correlator

  - Expected response

  On completion, the z/OS jobs REST service returns an HTTP response with an Array of matching jobs, each as a JSON job document.

z/OS jobs service – API example

- List the jobs for an owner, prefix or job ID
  - Example request

  In the following example, the GET method is used to list the jobs that are owned by "IBMUSER" and have a job name prefix beginning with "TESTJOB":

```
GET /zosmf/restjobs/jobs?owner=IBMUSER&prefix=TESTJOB* HTTP/1.1
```

  - Example response

```
HTTP/1.1 200 OK
Date: Fri, 17 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

[
{"jobid":"JOB00023","jobname":"TESTJOB2","subsystem":null,"owner":"IBMUSER",
"status":"OUTPUT","type":"JOB","class":"A","retcode":"CC 0000",
"url":"https:\/\/host:port\/zosmf\/restjobs\/jobs\/TESTJOB2\/JOB00023",
"files-url":"https:\/\/host:port\/zosmf\/restjobs\/jobs\/TESTJOB2\/JOB00023\/files"},
{"jobid":"JOB00024","jobname":"TESTJOB3","subsystem":null,"owner":"IBMUSER",
"status":"OUTPUT","type":"JOB","class":"A","retcode":"ABEND S000",
"url":"https:\/\/host:port\/zosmf\/restjobs\/jobs\/TESTJOB3\/JOB00024",
"files-url":"https:\/\/host:port\/zosmf\/restjobs\/jobs\/TESTJOB3\/JOB00024\/files"}
]
```

z/OS jobs service – API example

- Submit a job
  - URL format

https://host:port/zosmf/restjobs/jobs{/-JESB}

    - To submit a job to secondary JES, use “-JESB” in the URL
    - The job to be submitted could be included in the request body OR resided in a data set or unix file in the host z/OS system.

  - Expected response

On completion, the z/OS jobs REST service returns an HTTP response with a JSON job document. The document contains information about the submitted job such as jobid, status, type, etc.

z/OS jobs service – API example

- Submit a job
  - Example request

The following request submits a job “TESTJOBX” to run on z/OS. The content of the job to be submitted is included in the request.

```
PUT /zosmf/restjobs/jobs HTTP/1.1
Content-Type: text/plain
X-IBM-Intrdr-Class: A
X-IBM-Intrdr-Recfm: F
X-IBM-Intrdr-Lrecl: 80
X-IBM-Intrdr-Mode: TEXT

//TESTJOBX JOB (),MSGCLASS=H
// EXEC PGM=IEFBR14
```

  - Example response

```
HTTP/1.1 201 Created
Date: Fri, 17 Jan 2014 05:39:28 +0000GMT
Content-Type: application/json
Connection: close

{
"jobid":"JOB00025","jobname":"TESTJOBX","subsystem":null,"owner":"IBMUSER",
"status":"INPUT","type":"JOB","class":"A","retcode":null,
"url":"https:\/\/host:port\/zosmf\/restjobs\/jobs\/TESTJOBX\/JOB00025",
"files-url":"https:\/\/host:port\/zosmf\/restjobs\/jobs\/TESTJOBX\/JOB00025\/files"
}
```

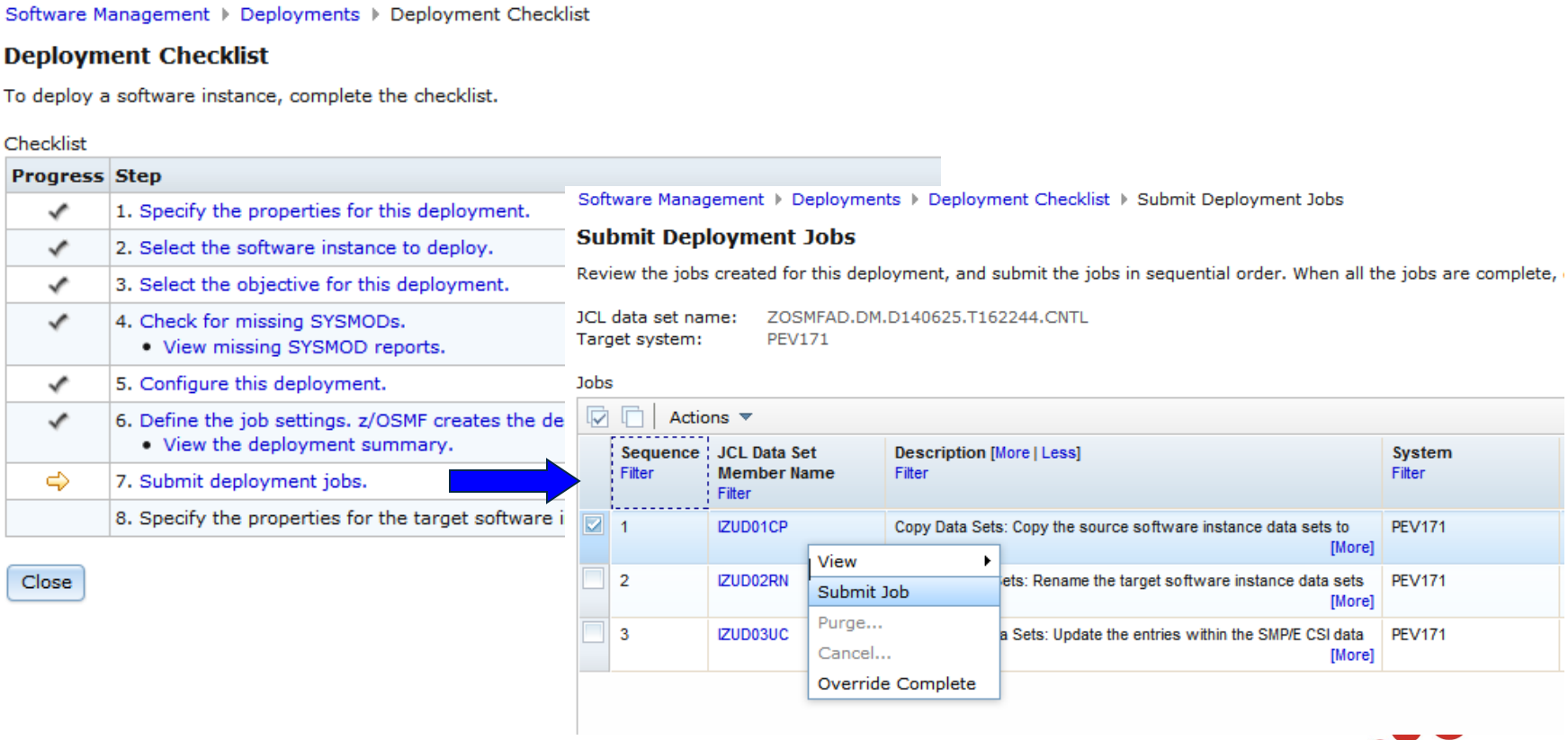
z/OS jobs service – API exploiters

- By exploiting z/OS jobs RESTful service:
  - application could focus on implementing their own logic and be released from taking care about how to work with z/OS jobs remotely or locally
  - application can easily call these jobs API.

- Current exploiters:
  - Explorer family such as CICS Explorer, z/OS Explorer.
  - z/OSMF Software Management
  - z/OSMF SDSF UI
  - z/OSMF Workflows
  - Customer's self-developed web application

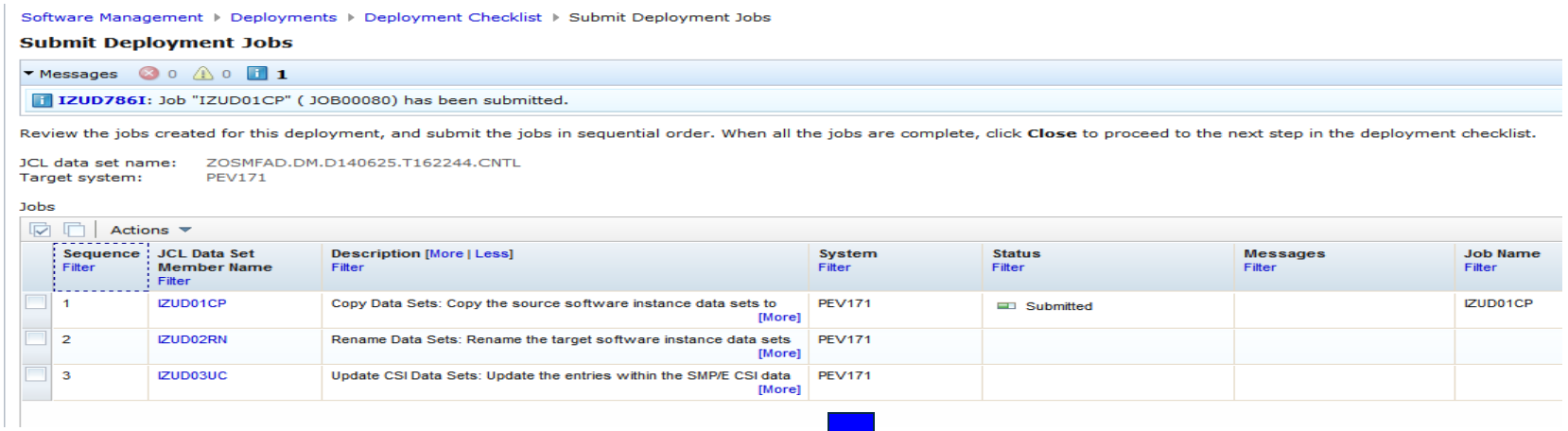
z/OS jobs service – API exploiters

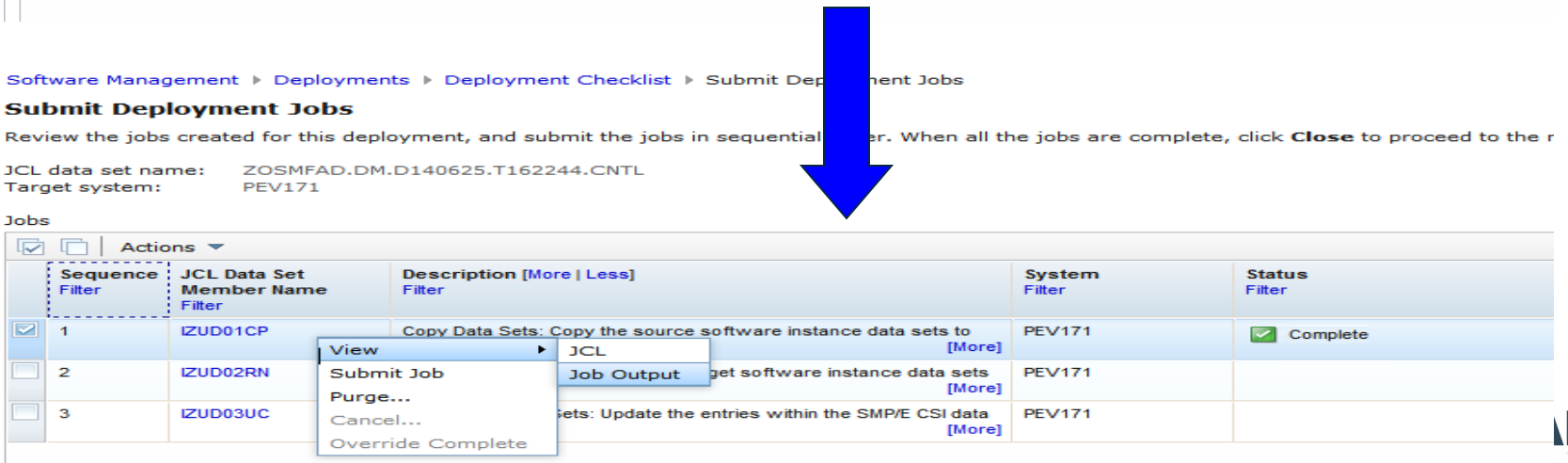
- z/OSMF Software Management plugin easily gets job management capability to manage deploy jobs (1/3)



z/OS jobs service – API exploiters

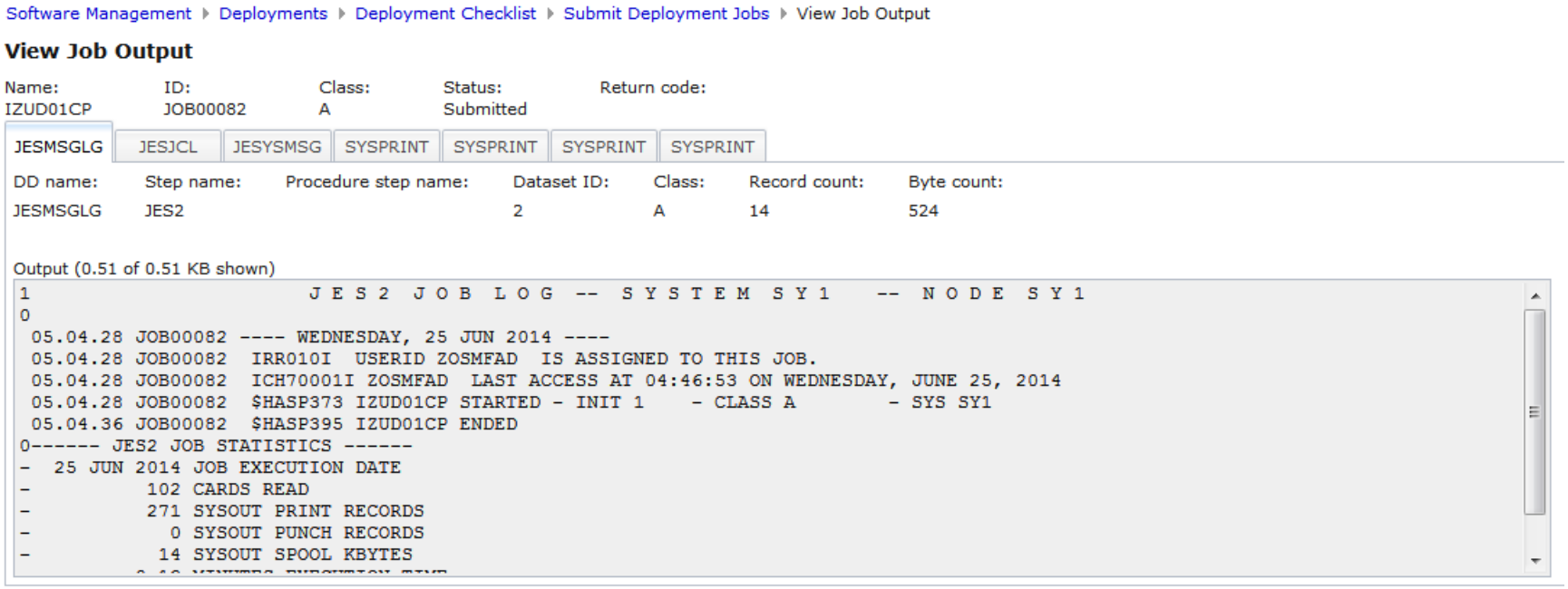
- z/OSMF Software Management plugin easily gets job management capability to manage deploy jobs (2/3)





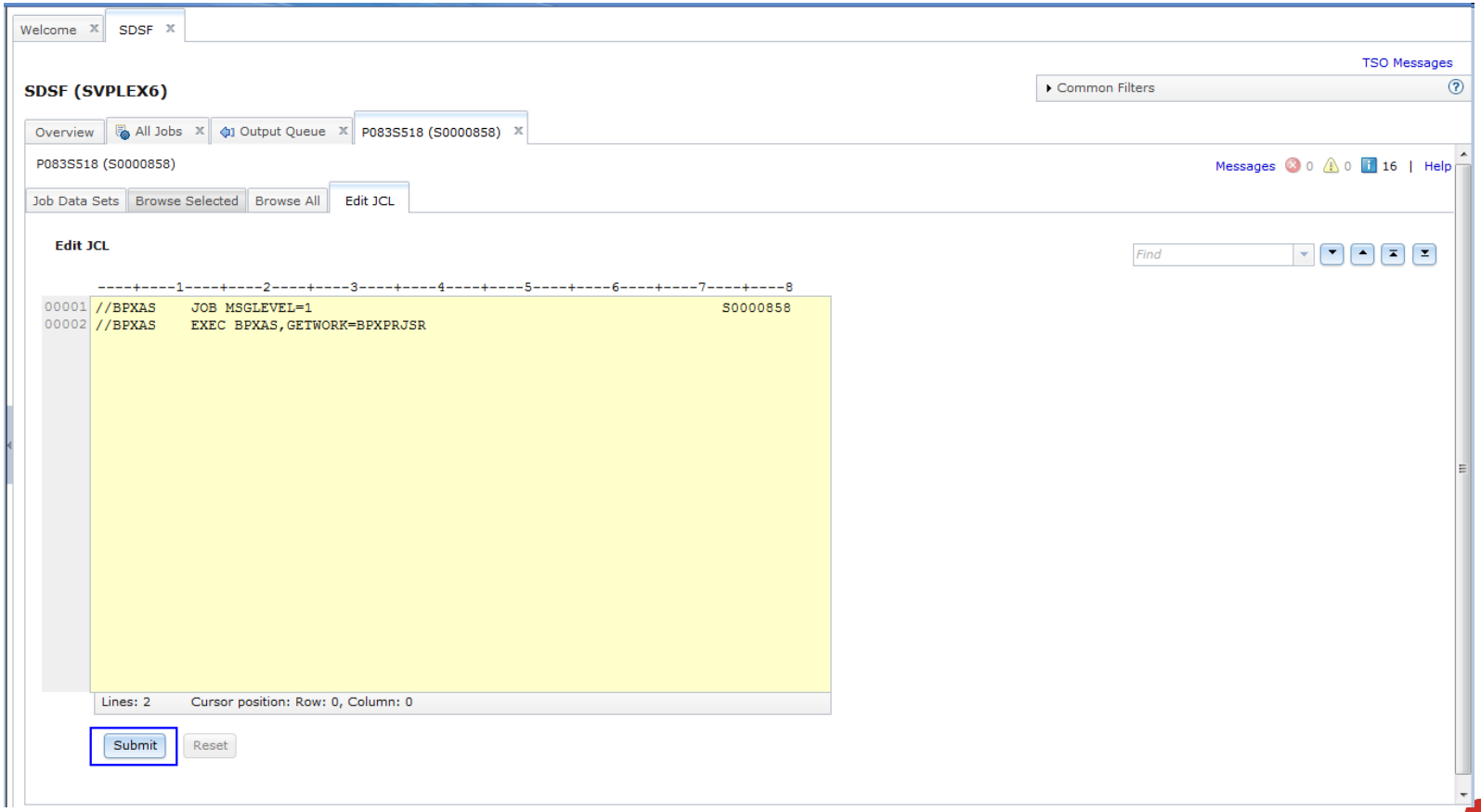
z/OS jobs service – API exploiters

- z/OSMF Software Management plugin easily gets job management capability to manage deploy jobs (3/3)



z/OS jobs service – API exploiters

- z/OSMF SDSF UI exploits “z/OS jobs RESTful service” to submit job.



z/OS data set and file service – API list

- z/OS data set and file service is provided for working with data sets and unix files on a z/OS system.
- z/OS data set and file RESTful service provides below operations (APIs):

Operation	Method
List the data sets on a z/OS system. You can filter the returned list of data set names through the specification of high-level qualifiers and wildcards.	GET
List the members of a z/OS partitioned data set (PDS or PDSE). You can filter the returned list of member names through the specification of wildcards.	GET
List the files and directories in a UNIX file path on a z/OS system.	GET
Retrieve the contents of a sequential data set, or a member of a PDS or PDSE.	GET
Retrieve the contents of a z/OS UNIX file.	GET
Write data to a sequential data set or a member of a PDS or PDSE.	PUT
Write data to a z/OS UNIX file.	PUT

z/OS data set and file service – API example

- List the z/OS data sets on a system
  - URL format

    https://host:port/zosmf/restfiles/ds/?dslevel=filter-criteria

    https://host:port/zosmf/restfiles/ds/?dslevel=filter-criteria&volser=volume-serial

  - Expected response

    On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred.

    For a successful request, the HTTP response includes an array of matching data sets, each as a JSON list document.

z/OS data set and file service – API example

- List the z/OS data sets on a system
  - Example request

  The following request is used to list all of the cataloged data sets that match the partial name "sys1.*lib". That is, the data sets that have a name beginning with "sys1". and a last qualifier that ends in "lib".

```
GET https://zosmf1.yourco.com/zosmf/restfiles/ds/?dslevel=sys1.*lib HTTP/1.1
```

  - Example response

```
HTTP/1.1 200 OK
Date: Wed, 23 Oct 2013 00:43:49 GMT
Content-Type: application/json
Connection: close
{ "items":[ { "dsname":"SYS1.AUXLIB" }, { "dsname":"SYS1.BDTLIB" }, { "dsname":"SYS1.CHSLIB" },
{ "dsname":"SYS1.CMDLIB" }, { "dsname":"SYS1.COBLIB" }, { "dsname":"SYS1.CSSLIB" },
{ "dsname":"SYS1.FDEFLIB" }, { "dsname":"SYS1.FONTLIB" }, { "dsname":"SYS1.FORTLIB" },
{ "dsname":"SYS1.GRSLIB" }, { "dsname":"SYS1.IMAGELIB" }, { "dsname":"SYS1.JES3LIB" },
{ "dsname":"SYS1.KANLIB" }, { "dsname":"SYS1.LINKLIB" }, { "dsname":"SYS1.LPALIB" },
{ "dsname":"SYS1.MACLIB" }, { "dsname":"SYS1.MIGLIB" }, { "dsname":"SYS1.NCPLIB" },
{ "dsname":"SYS1.NFSLIB" }, { "dsname":"SYS1.OVERLIB" }, { "dsname":"SYS1.PARMLIB" },
{ "dsname":"SYS1.PARMLIB.CB" }, { "dsname":"SYS1.PARMLIB.INSTALL" },
{ "dsname":"SYS1.PARMLIB.MSYS" }, { "dsname":"SYS1.PARMLIB.PD" }, { "dsname":"SYS1.PROCLIB" },
{ "dsname":"SYS1.PROCLIB.CB" }, { "dsname":"SYS1.PROCLIB.INSTALL" },
{ "dsname":"SYS1.PROCLIB.MSYS" }, { "dsname":"SYS1.PROCLIB.PD" },
{ "dsname":"SYS1.PROCLIB.TEST" }, { "dsname":"SYS1.PSEGLIB" }, { "dsname":"SYS1.PSPMLIB" },
{ "dsname":"SYS1.SADRYLIB" }, { "dsname":"SYS1.SAMPLIB" }, { "dsname":"SYS1.SIATLIB" },
{ "dsname":"SYS1.SICETLIB" }, { "dsname":"SYS1.SIFALIB" }, { "dsname":"SYS1.SISTCLIB" },
{ "dsname":"SYS1.SORTLIB" }, { "dsname":"SYS1.SVCLIB" }, { "dsname":"SYS1.VTAMLIB" },
{ "dsname":"SYS1.XCFLIB" } ], "returnedRows":43 }
```

z/OS data set and file service – API example

- Retrieve the contents of a z/OS data set or member
  - URL format

    For a request to retrieve data from a sequential data set:

    https://host:port/zosmf/restfiles/ds/<data-set-name>

    For a request to retrieve data from a member of a PDS or PDSE:

    https://host:port/zosmf/restfiles/ds/<data-set-name>(<member-name>)

    For a request to retrieve data from an uncataloged sequential data set:

    https://host:port/zosmf/restfiles/ds/-(volser)/<data-set-name>

    For a request to retrieve data from a member of an uncataloged PDS or PDSE:

    https://host:port/zosmf/restfiles/ds/-(volser)/<data-set-name>(<member-name>)

  - Expected response

    On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success.

z/OS data set and file service – API example

- Retrieve the contents of a z/OS data set or member
  - Example request

  The following request is used to is used to retrieve the contents of the member SMFPRM00 in data set SYS1.PARMLIB:

```
GET https://zosmf1.yourco.com/zosmf/restfiles/ds/SYS1.PARMLIB(SMFPRM00) HTTP/1.1
```

  - Example response

```
200 OK
Etag: B5C6454F783590AA8EC15BD88E29EA63
Content-Type: text/plain; charset=UTF-8
Content-Language: en-US
Content-Length: 1944
Date: Fri, 07 Nov 2014 02:13:07 GMT
Connection: close
```

```
ACTIVE                        /*ACTIVE SMF RECORDING*/              00010000
DSNAME(SYS1.&SMFDSN1,SYS1.&SMFDSN2,    /*SMF ON 3390        */     00020000
 SYS1.&SMFDSN3)                      /*FT: SYSAQ3, TS: SYSAQ4 */     00030000
NOPROMPT                     /*PROMPT THE OPERATOR FOR OPTIONS*/    00040000
REC(PERM)                    /*TYPE 17 PERM RECORDS ONLY*/          00050000
MAXDORM(3000)                /* WRITE AN IDLE BUFFER AFTER 30 MIN*/ 00060000
MEMLIMIT(256M)               /* 256M FOR 64 BIT APPS             */ 00061005
STATUS(003000)               /* WRITE SMF STATS AFTER HALF HOUR*/   00070000
JWT(0700)                    /* INVOKE EXIT IEFUTL AFTER  7HR 00M*/ 00080002
SID(&SYSNAME),               /* SYSTEM ID FOR 3084 - SINGLE IMAGE*/ 00090000
LISTDSN                      /* LIST DATA SET STATUS AT IPL*/       00100000
INTVAL(30)                   /* INTVAL OPTION SP430 */              00110000
SYNCVAL(00)                  /* SYNCVAL OPTION SP430  */            00120000
SYS(NOTYPE(19,40,92),                                               00130001
 EXITS(IEFU83,IEFU84,IEFACTRT,IEFUJV,IEFUJI,                        00140000
        IEFUSI,IEFUTL,IEFU29),INTERVAL(010000),DETAIL)              00150000
                                                                    00160000
/* WRITE ALL RECORDS AS THE SYSTEM DEFAULT, TAKE ALL KNOWN          00170000
   EXITS, NOTE: JES EXITS CONTROLED BY JES , THERE IS NO            00180000
   DEFAULT INTERVAL RECORDS WRITTEN AND ONLY SUMMARY T32            00190000
   RECORDS AS A DEFAULT FOR TSO */                                  00200000
                                                                    00210000
SUBSYS(STC,NOTYPE(19,40,92),                                        00220001
 EXITS(IEFU29,IEFU83,IEFU84,IEFUTL),                                00230000
 INTERVAL(SMF,SYNC),DETAIL)          /*SP430*/                    00240000
                                                                    00250000
/* WRITE ALL RECORDS AS BY  SYSTEM DEFAULT, TAKE ONLY THREE         00260000
   EXITS, NOTE: IEFU29 EXECUTES IN THE MASTER ASID WHICH IS A       00270000
   STC ADDRESS SPACE SO IEFU29 MUST BE ON FOR STC. USE ALL OTHER    00280000
   SYS PARAMETERS AS A DEFAULT  */                                  00290000
```

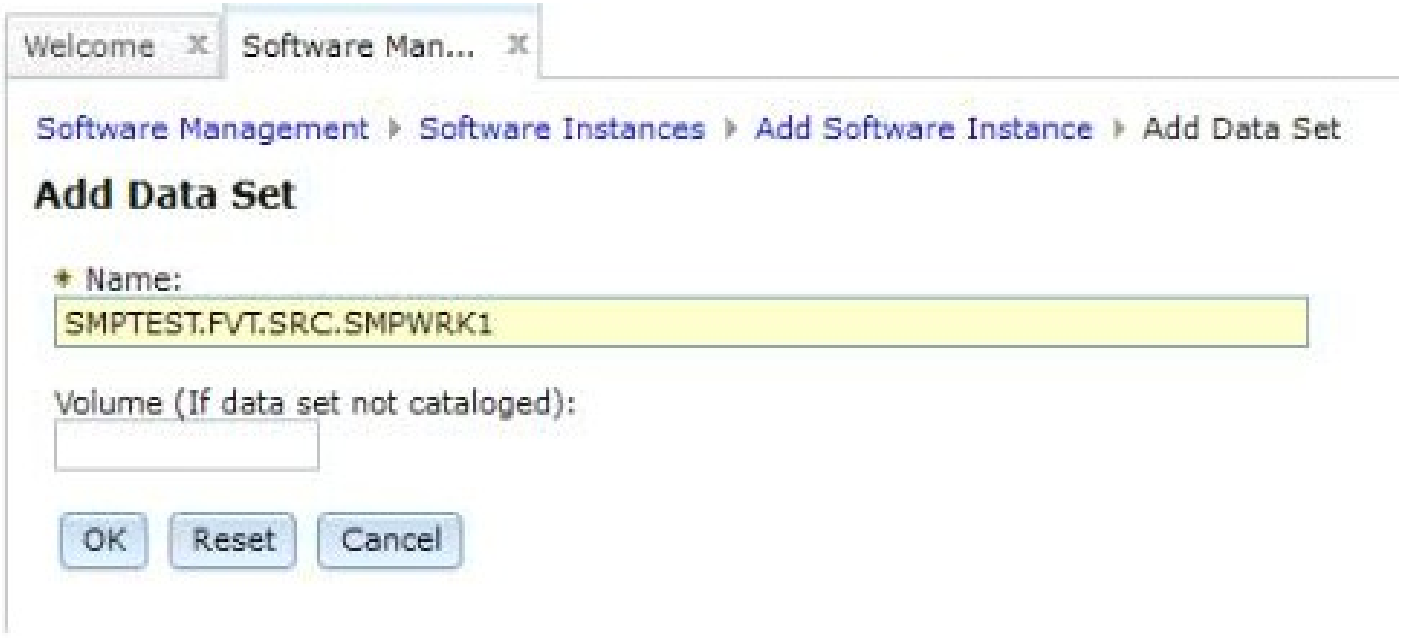
z/OS data set and file service – API exploiters

- By exploiting z/OS data set and file RESTful service, application could easily get the capability of managing data set and unix files remotely or locally.

- Current exploiters:
  - z/OSMF Software Management
  - Customer's self-developed web application

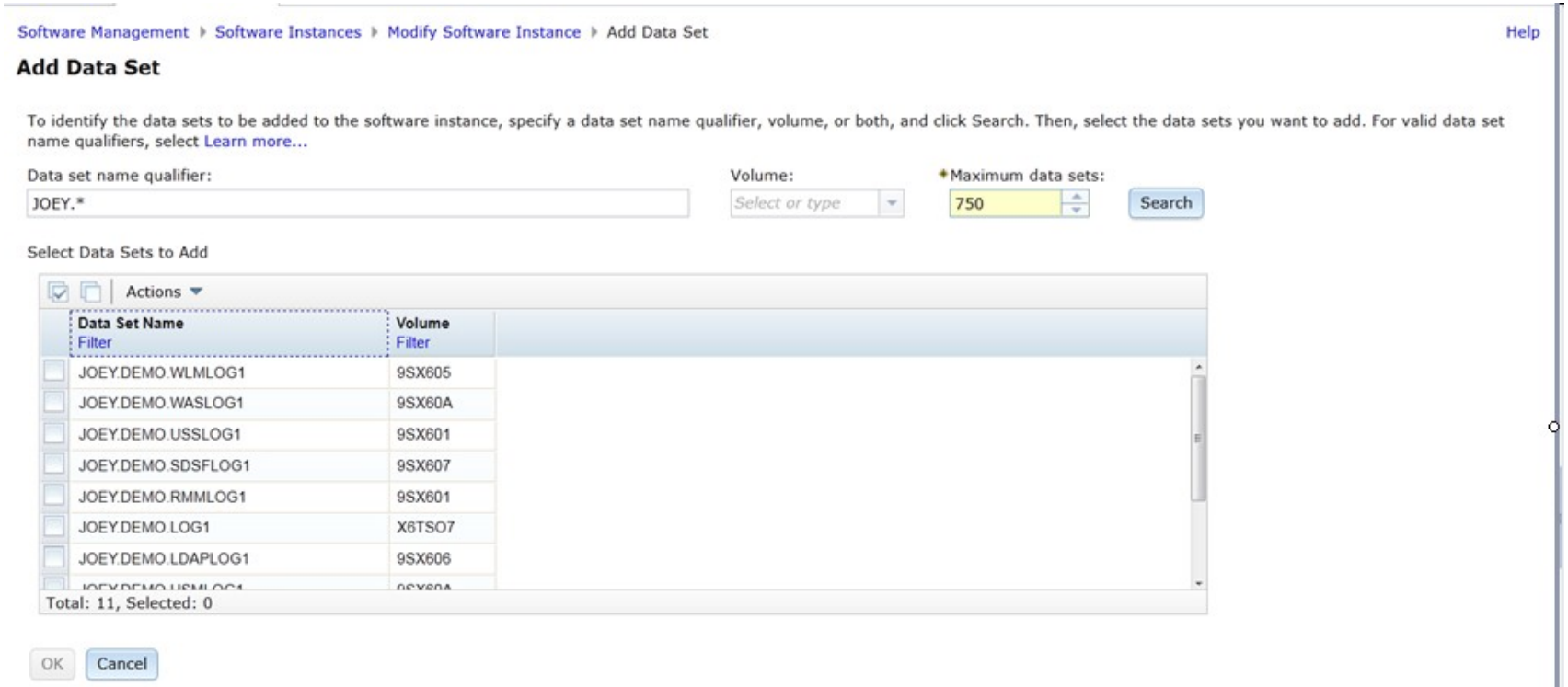
z/OS data set and file service – API exploiters

- z/OSMF Software Management plugin simplifies the process of adding data set into software instance.
- Previously, user has to enter data set name manually, and, therefore, one data set at a time.



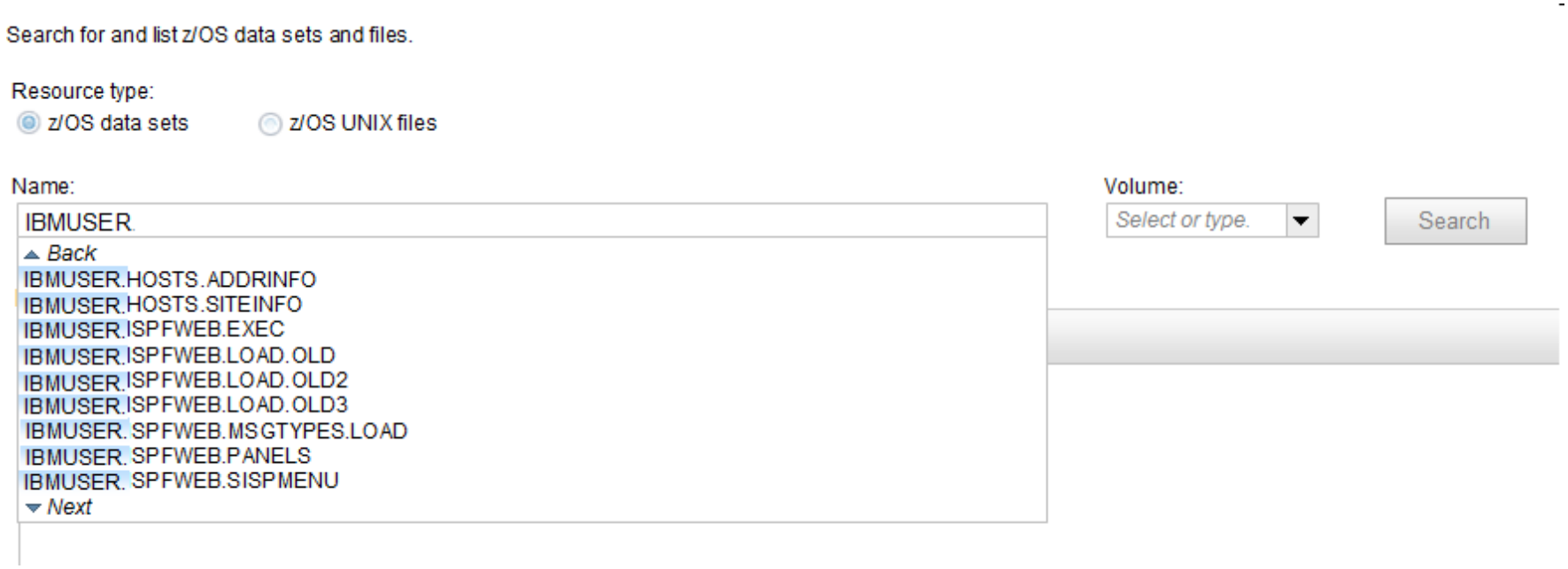
z/OS data set and file service – API exploiters

- With exploiting “z/OS data set and file service”, Software Management plugin provides searching data set function. User can select multiple data sets returned by the search operation and add them once.



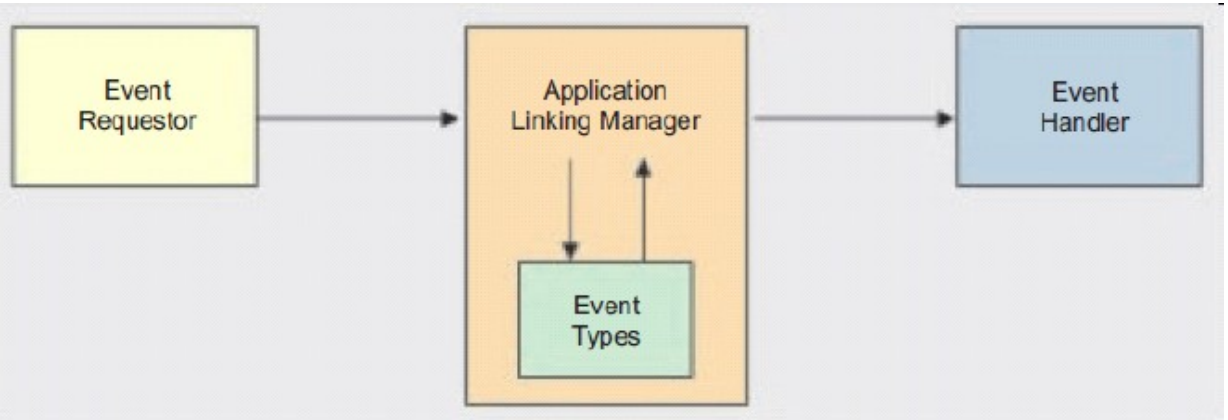
z/OS data set and file service – API exploiters

- Build something cool by exploiting “z/OS data set and file service”:



Application Linking Manager service

- To perform traditional system management tasks in z/OS, you might interact with several different interfaces.
- In z/OSMF, it is possible to link or connect some of these tasks and external applications together for a smoother user experience via the Application linking Manager service.
- Key components:
  - Event Requestor: z/OSMF task or external application
  - Event: Action requested by the “Event Requestor”
  - Event type: Object that contains an “Event Requestor” to an “Event Handler”
  - Event handler: z/OSMF task or external application



Application Linking Manager service – API list

- Application Linking Manager service provides below operations (APIs):

Operation	Method
Register an event type.	POST
Register a handler for an event type	POST
List all tasks that are eligible to be handlers	GET
List the registered handlers for an event type	GET
Unregister a handler	DELETE
Unregister an event type	DELETE

Application Linking Manager service – API example

- Register a handler for an event type
  - URL format

https://*{host}*:*{port}*/zosmf/izual/rest/handler?eventTypeId=*{eventTypeId}*

Request content:

type: handler type. “INTERNAL” for z/OSMF plug-in, “EXTERAL” for external application.

Id: Unique identifier for a launch point within the handler task or application.

applID: Identifier assigned to z/OSMF plugin

displayName: handler name

URL: URL to be used for accessing the handler.

options: indicates how the handler will be displayed when it process events.

  - Expected response

On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred.

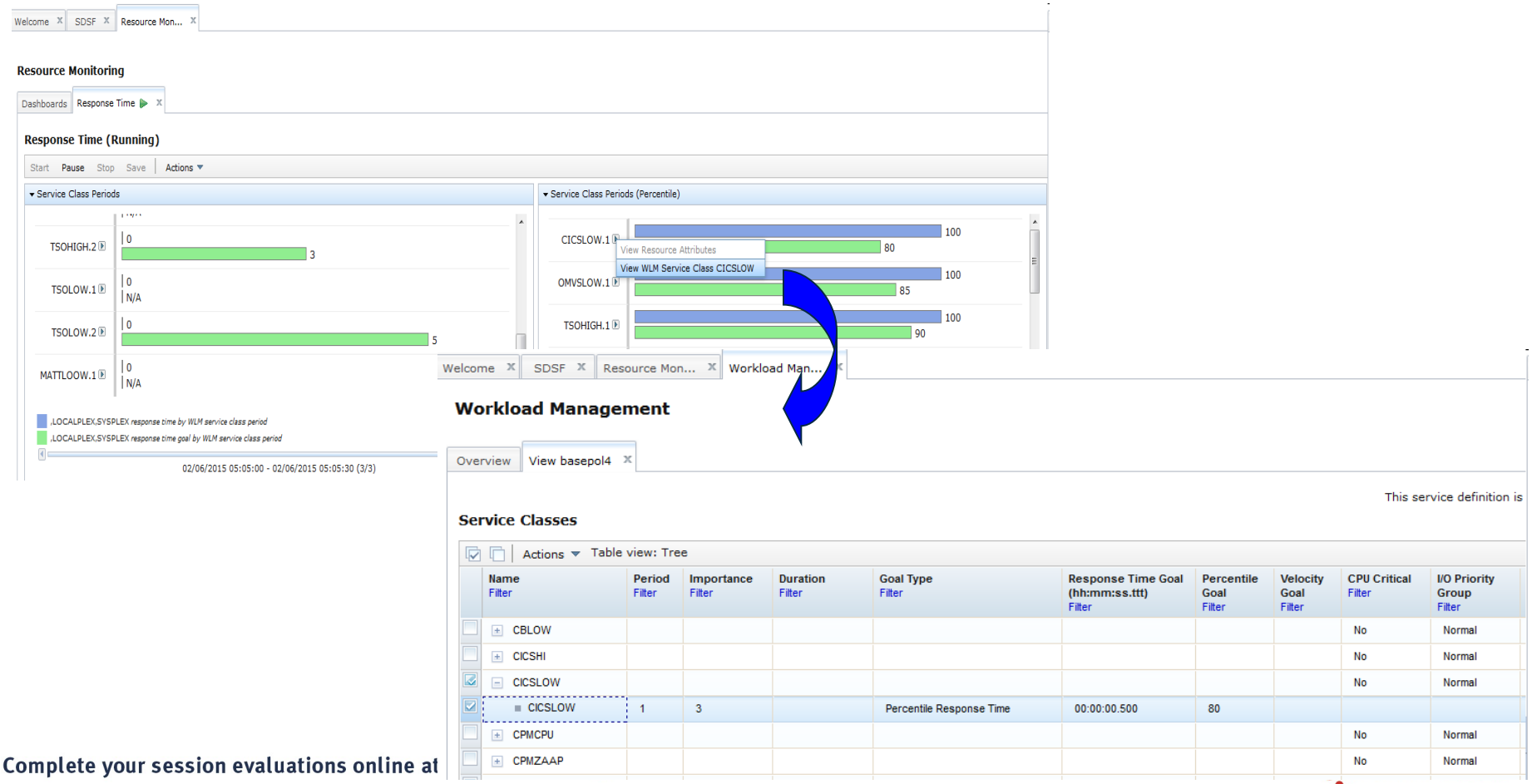
Application Linking Manager service – API exploiters

- z/OSMF WLM plugin is used to define performance policy. z/OSMF RMF plugin is used to monitor performance. By exploiting “Application Linking Manager service”, WLM and RMF provide a smooth user experience when there is need to switch between these two plugins.

- Example of linkage between z/OSMF RMF plugin and z/OSMF WLM plugin (1/2)
  - Event Requestor: z/OSMF RMF plugin
  - Event type: IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.SERVICE_CLASS
  - Event Handler: z/OSMF WLM plugin

Application Linking Manager service – API exploiters

- Example of linkage between z/OSMF RMF plugin and z/OSMF WLM plugin (2/2)



TSO/E address space service – API list

- TSO/E address space service is provided for web application (especially for external web application) to communicate with their back-end TSO/E application running on the z/OS system.
- TSO/E address space RESTful service provides below operations (APIs):

Operation	Method
Start or reconnect to a TSO/E address space	POST
Start an application on a TSO/E address space	POST
Receive messages from a TSO/E address space	GET
Receive messages from an application running in a TSO/E address space	GET
Send messages to a TSO/E address space	PUT
Send messages to an application running in a TSO/E address space	PUT
Ping a TSO/E address space	PUT
End a TSO/E address space	DELETE

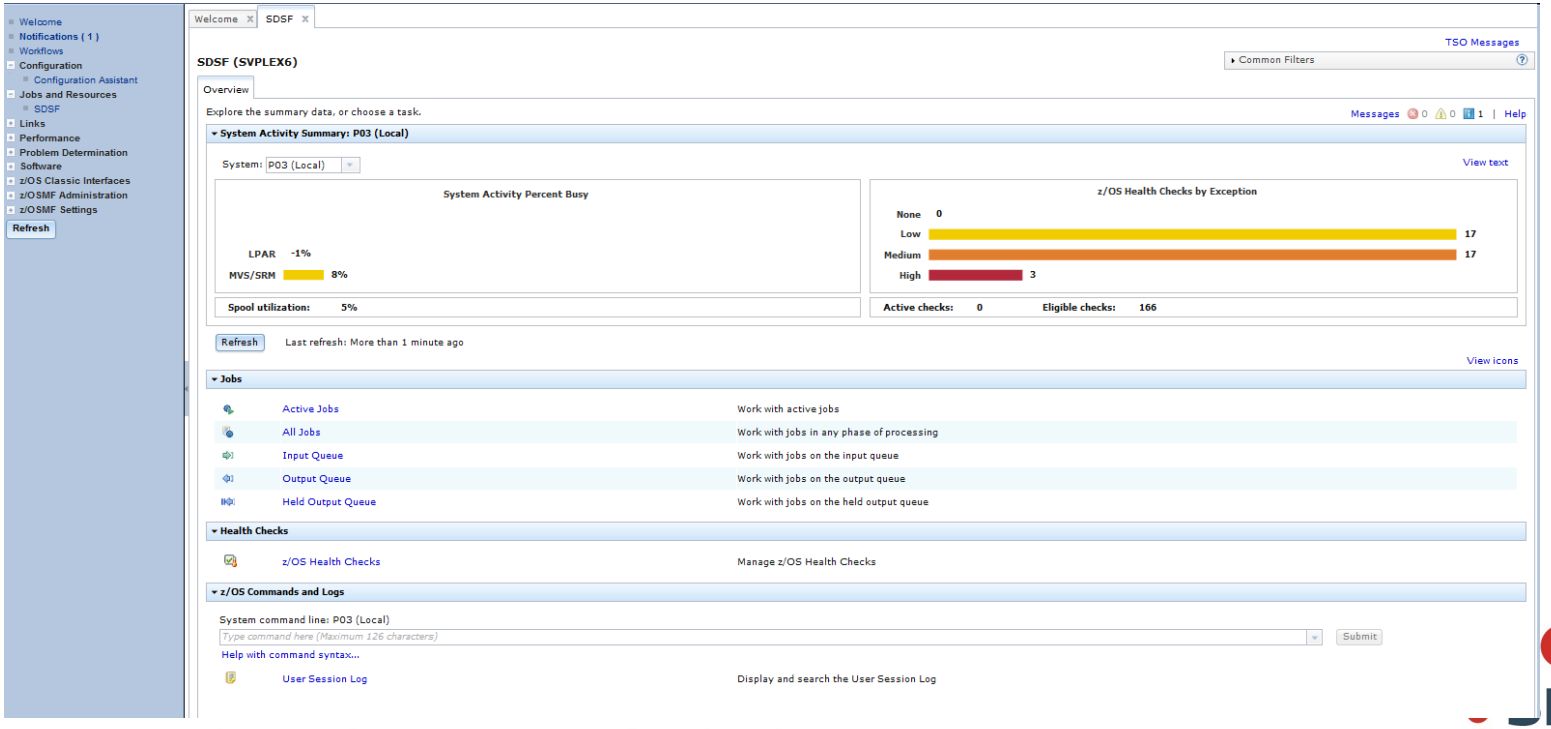
Data persistence service – API list

- Data persistence service is provided for web application (especially for external web application) to manage their persistent data in z/OS system.
- Data persistence service provides below operations (APIs):

Operation	Method
Persist user-specific and global application data	PUT
Retrieve user-specific and global application data	GET
Delete user-specific and global application data	DELETE

TSO/E address space service & Data persistence service – API Exploiter

- Exploiters (not only web application) could easily get the capability of communicating with TSO/E application and managing persistence data in the z/OS system such that they can focus on implementation of business logic.
- z/OS SDSF created their own z/OSMF plugin (a.k.a. external plugin) “SDSF UI” by exploiting these RESTful services.



Topology service** – API list

- Topology RESTful service is provided for working with the groups, sysplexes and the systems that are defined to z/OSMF.

- Topology RESTful service provides below operations (APIs):

Operation	Method
List the systems defined to z/OSMF	GET
List the groups defined to z/OSMF	GET
List the systems included in a group	GET
List the sysplexes defined to z/OSMF	GET
List the systems included in a sysplex	GET

** available for z/OSMF V2R1 with APAR PI32148

Multisystem routing service** – API list

- To communicate with and transfer data between systems within your enterprise, z/OSMF uses z/OSMF-to-z/OSMF communication. Multisystem routing service plays key role in the z/OSMF-to-z/OSMF model. It has the capability of:
  - Forwarding request to single remote z/OSMF instance and return the response (Basic-Proxy mode)
  - Forwarding request to a group of remote z/OSMF instances, packaging the responses and return it. (Aggregation mode)
- Multisystem routing service provides below operations (APIs):

Operation	Method
Retrieve data from one system, a list of systems, or all the systems in a group.	GET
Update data for one system, a list of systems, or all the systems in a group.	POST or PUT
Delete data from one system, a list of systems, or all the systems in a group.	DELETE
Authenticate with a secondary z/OSMF instance.	POST

** available for z/OSMF V2R1 with APAR PI32148

Topology service & Multisystem routing service – API Exploiter

- Exploiters could:
  - Retrieve topology information through “Topology service”
  - Manage multiple systems with only connected to primary z/OSMF which runs “Multisystem routing service”.
  - Get the aggregation capability by only specifying target systems/groups to “Multisystem routing service”.

** available for z/OSMF V2R1 with APAR PI32148

Workflow service** – API list

- Workflow RESTful service allows user to create, start and manage work flow in z/OS through programmatic way instead of having to operate in z/OSMF UI.

- Workflow RESTful service provides below operations (APIs):

Operation	Method
Create a workflow	POST
Start a workflow	PUT
Lists workflows by search criteria	GET
Delete a workflow	DELETE
Cancel workflow	PUT
Retrieve workflow definition	GET

** available for z/OSMF V2R1 with APAR PI32148

Workflow service – API exploiter

- User scenario
  - Product (workflow provider) ships workflow which intends to accomplish a task in z/OS system.
  - End user of the product (or even the product itself) could build application which (remotely or locally) calls workflow RESTful service to:
    - Create workflow on demand
    - Perform the workflow to accomplish the task in an automatic way
    - Activities of performing workflow is recorded by z/OSMF workflows application for future review

Software management service** – API list

- Software management RESTful service allows a client application to interact with the z/OSMF Software Management task.
- Software managment RESTful service provides below operations (APIs):

Operation	Method
List the software instances defined to z/OSMF.	GET
Retrieve the properties of a software instance.	GET
Add a new software instance.	POST

** available for z/OSMF V2R1 with APAR PI32148

Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Topology service**
  - Multisystem routing service**
  - z/OSMF workflow service**
  - Software Management service**
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler
- Summary

** available for z/OSMF V2R1 with APAR PI32148

Using z/OSMF RESTful services – simple demo

- Experience z/OSMF RESTful services with a simple html page (1/3)

```
<body>
    <div>
    <button id="getMyJobs" onclick="getMyJobs()">List my jobs</button>
    </div><br/>
    <hr />
    <div id="submitArea">
        <textarea id="jclArea" style="width: 400px">//TESTJOBW JOB (),MSGCLASS=H
// EXEC PGM=IEFBR14</textarea>
    </div><br/>
    <button id="subbtn" onclick="submit()">Submit a job</button><br/><br/>
    <hr/>
    <div id="getJobArea">
            <label>Job Name:</label><input id="jobname" />
            <label>Job ID:</label><input id="jobid" />
    </div><br />
    <button id="getbtn" onclick="getJobStatus()">Get job status</button><br/><br/>
    <div id="resultArea">
        <div><h4>Result:</h4></div>
        <div id="result"></div>
    </div>
</body>
</html>
```

Using z/OSMF RESTful services – simple demo

- Experience z/OSMF RESTful services with a simple html page (2/3)

```
<head>
    <title>Demo of REST zOS Jobs service</title>
    <script type="text/javascript">
        function submit(){
            var jcl=document.getElementById("jclArea").value;
            var xhrs = new XMLHttpRequest();
            xhrs.open("PUT", "https://mvs1.centers.ihost.com/zosmf/restjobs/jobs", true);
            xhrs.setRequestHeader("Content-Type", "text/plain");
            xhrs.onreadystatechange = function (){
                if(xhrs.readyState == 4){
                    document.getElementById("result").innerHTML = xhrs.responseText;
                }
            }
            xhrs.send(jcl);
        }
        function sendGetRequest(url) {
            var xhrget = new XMLHttpRequest();
            xhrget.open("GET", url, true);
            xhrget.onreadystatechange = function (){
                if(xhrget.readyState == 4){
                    document.getElementById("result").innerHTML = xhrget.responseText;
                }
            }
            xhrget.send();
        }
        function getJobStatus(){
            var xhrg = new XMLHttpRequest();
            var jobname = document.getElementById("jobname").value;
            var jobid = document.getElementById("jobid").value;
            var url = "https://mvs1.centers.ihost.com/zosmf/restjobs/jobs/"+jobname+"/"+jobid;
            sendGetRequest(url);
        }
        function getMyJobs() {
            var url = "https://mvs1.centers.ihost.com/zosmf/restjobs/jobs";
            sendGetRequest(url);;
        }
    </script>
</head>
```

Construct the URL

Access the RESTful service with JCL to be submitted

Construct the URL

Access the RESTful service

Construct the URL

Access the RESTful service

Using z/OSMF RESTful services – simple demo

- Experience z/OSMF RESTful services with a simple html page (3/3)

List my jobs

```
//TESTJOBW JOB (),MSGCLASS=H
// EXEC PGM=IEFBR14
```

Submit a job

Job Name: TESTJOBW Job ID: JOB27785

Get job status

Result:

{"jobname":"TESTJOBW","retcode":"CC 0000","subsystem":"JES2","url":"https:\/\/mvs1.centers.ihost.com:443\/zosmf\/restjobs\/jobs\/J0027785N1.....CE96D4ED.....%3A","status":"OUTPUT","owner":"SHARA01","jobid":"JOB27785","class":"A","job-correlator":"J0027785N1.....CE96D4ED.....:","files-url":"https:\/\/mvs1.centers.ihost.com:443\/zosmf\/restjobs\/jobs\/J0027785N1.....CE96D4ED.....%3A\/files","type":"JOB","phase":20,"phase-name":"Job is on the hard copy queue"}

Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Topology service**
  - Multisystem routing service**
  - z/OSMF workflow service**
  - Software Management service**
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler 
- Summary

** available for z/OSMF V2R1 with APAR PI32148

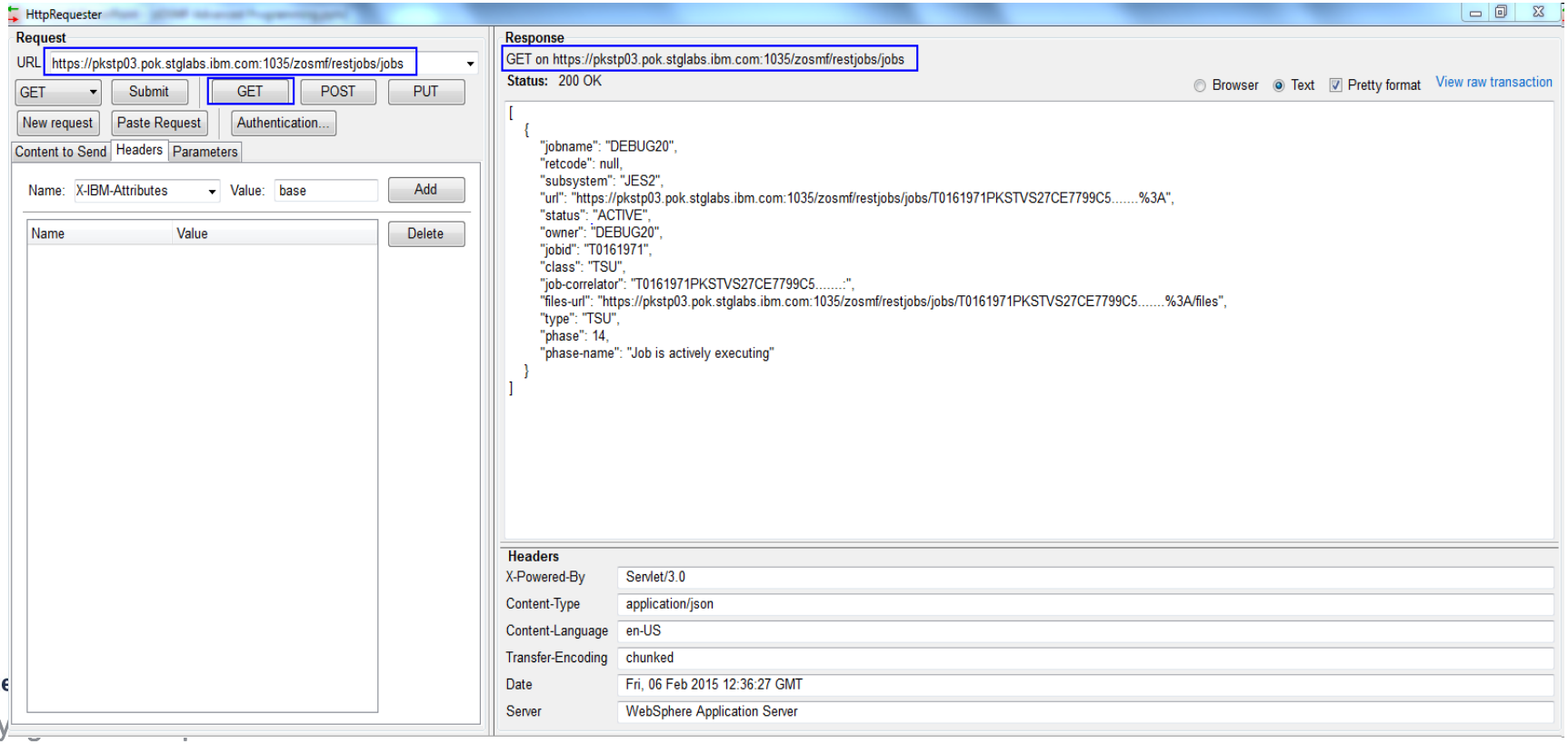
Using z/OSMF RESTful services - Simpler than Simpler

- Experience z/OSMF RESTful services without the need to write any code:

  What you need are just:
  - Valid z/OSMF user id and password
    - What can I do if I don't have a z/OSMF installation?
  - Firefox browser with "HttpRequester" plugin installed

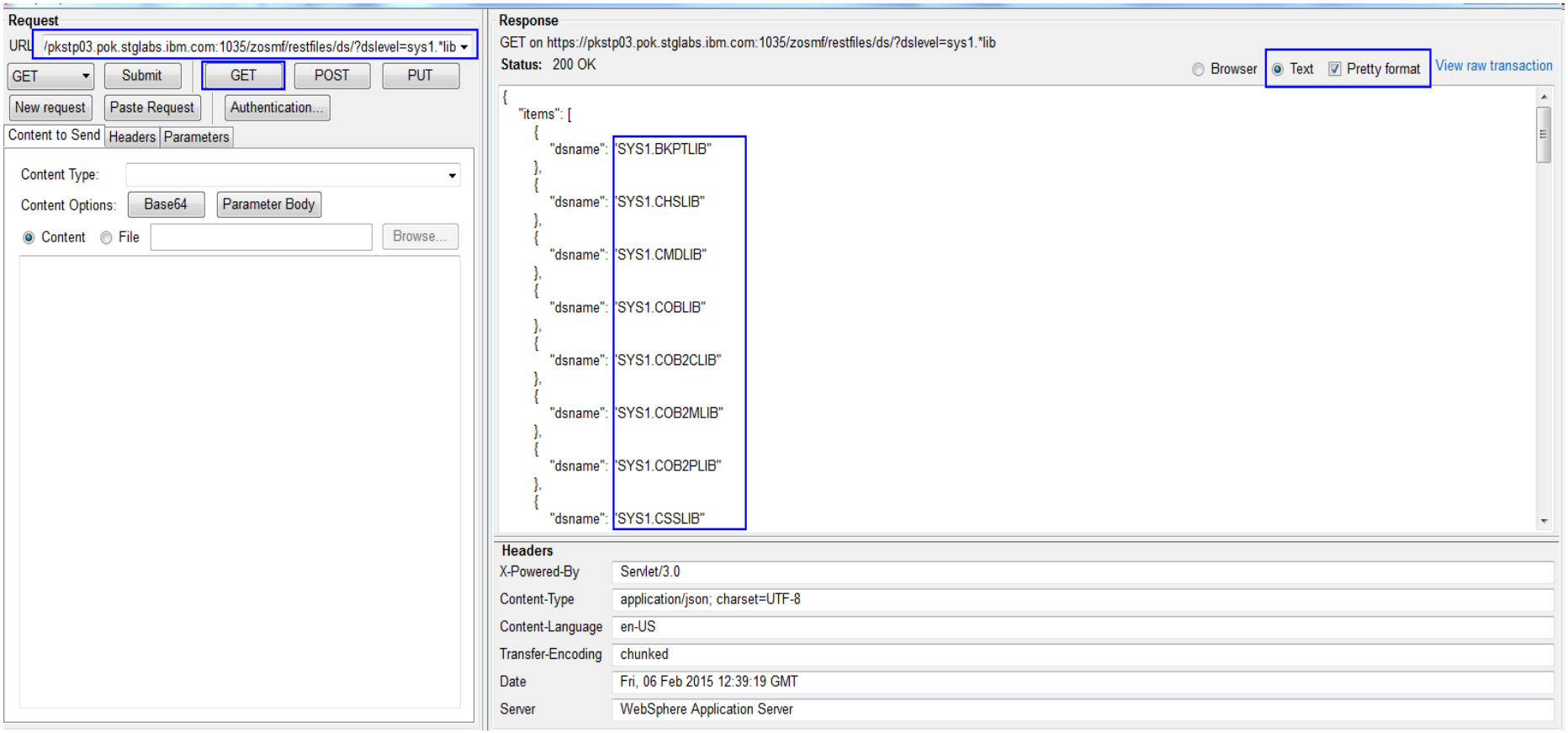
Session 16656 z/OSMF Hands-On labs

- Example 1 – list jobs owned by your logon user



Using z/OSMF RESTful services - Simpler than Simpler

- Example 2 – list data set names matches the criteria of “SYS1.*lib”



Advertisements

- Session 16656 Lab: z/OSMF Hands-On Labs - Choose Your Own – III

  Friday, March 6, 2015: 8:30 AM-9:30 AM Redwood (Sheraton Seattle)

  (If you missed the labs you can review the handouts and see how z/OSMF works)

- Possible missed opportunities (handouts available for download):
  - Session 16643 The New and Improved z/OSMF 2.1
  - Session 16801 Getting Started with z/OSMF Resource Monitoring - Hands-on Lab
  - Session 16646 Using z/OSMF Workflows for Configuration
  - Session 16659 z/OSMF User Experiences
  - Session 16654 z/OSMF 2.1 Implementation and Configuration

** available for z/OSMF V2R1 with APAR PI32148

Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Topology service**
  - Multisystem routing service**
  - z/OSMF workflow service**
  - Software Management service**
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler
- Summary 

** available for z/OSMF V2R1 with APAR PI32148

Summary

- In addition to z/OSMF plugins with modern UI and simplified task, z/OSMF also provides services and facilities to help you write programs.
- z/OSMF Advanced Programming consists of:
  - Using z/OSMF RESTful services
  - Develop workflow
  - Create your own z/OSMF plugins
- z/OSMF RESTful services make z/OS and z/OSMF more approachable:
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Topology service**
  - Multisystem routing service**
  - z/OSMF workflow service**
  - Software Management service**

** available for z/OSMF V2R1 with APAR PI32148

Thank You

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

Appendix

- z/OSMF Home Page

http://www-03.ibm.com/systems/z/os/zos/features/zosmf/index.html

- z/OSMF V2R1 Programming Guide

http://www-01.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.izua700/toc.htm