Zowe redefines the way we interact with z/OS

Joe Winchester – IBM
The traditional mainframe

55 years ago the mainframe was created
– Code written then still runs now unmodified

s360 was named after 360 degrees – full compass orientation coverage

Mainframes helped the Apollo program put humans onto the moon in 1969
The modern mainframe

70% of the world data is stored in mainframes

Used by 25 of the world's top banks and 70% of Fortune 500 companies

Mainframes process 30 billion business transactions per day around the world

- 68,542 google searches every second globally vs 1,200,000,000 on mainframe

- 68% of world's production workload capacity, 6.2% of total server spend

Mainframes have pervasive encryption at rest and in flight.
Open source and mainframe

- **Multi language server** with older languages (COBOL, PL/1) side by side with newer languages (Java, node, Swift, Python, ...)

- **z/LINUX** and **z/OX**
  Container Extensions
Why? What is the problem Zowe is trying to solve?

Zowe background
Zowe Command Line Interface
Zowe Desktop
Zowe APIs
Zowe futures

How to get involved and become a Zoweian

Zowe.org
We are building more than just technology, we are building a community
Different interfaces and tools for DevOps and infrastructure services

- TSO
- ISPF
- Eclipse
- Rexx
- JCL
- USS

Development and Operations Teams (Internal/IT Facing)
Enterprise

z/OS

IMS

Create, Build,
Manage,
Operate

Development and
Operations Teams
(Internal/IT Facing)

TSO
ISPF
Eclipse
Rexx
JCL
USS
Batch
WAS
MQ
CICS
DB2

PL/1
Node
Java
HLASM
Cobol

Debuggers

Problem

Determination

SCM
Monitors
Compilers

Attachments

APIs, Web UI, CLI for
Development and
operations

Zowe core base function

Vendors can build Zowe
extensions

Web UI
APIs
CLI

Development and
Operations Teams
(Internal/IT Facing)
Command Line Interface

Zowe.org

We are building more than just technology, we are building a community
Challenge

Michelle (App Dev)

AWS CLI, Azure CLI, Cloud Foundry CLI, ...

Add your favorite tool
Add your favorite tool
Installing and set-up

```sh
npm config set @brightside:registry https://api.bintray.com/npm/ca/brightside

npm install -g @zowe/core@lts-incremental

zowe plugins install @zowe/cics@lts-incremental

zowe profiles create zosmf 3b --user stevenh --password ****** --host winmvs3b.hursley.ibm.com --port 32070 --reject-unauthorized false --overwrite --base-path <my/api/layer>
```
Command Line Interface Basics
### Issue MVS Command

```bash
stevens-mbp:~ steven$ zowe console issue cmd "d a,l" --zosmf-p 27sh
```

<table>
<thead>
<tr>
<th>JOBS</th>
<th>M/S</th>
<th>TS</th>
<th>USERS</th>
<th>SYSAS</th>
<th>INITs</th>
<th>ACTIVE/MAX V TAM</th>
<th>OAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>00022</td>
<td>00071</td>
<td>00002</td>
<td>00041</td>
<td>00202</td>
<td>00001/00090</td>
<td>00063</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LLA</th>
<th>LLA</th>
<th>LLA</th>
<th>NSW</th>
<th>SDSF</th>
<th>SDSF</th>
<th>SDSF</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ISGECMON</th>
<th>ISGECMON</th>
<th>ISGECMON</th>
<th>OWT</th>
<th>VLF</th>
<th>VLF</th>
<th>VLF</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ASCH</th>
<th>ASCH</th>
<th>ASCH</th>
<th>NSW</th>
<th>JES2</th>
<th>JES2</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NETVSSI</th>
<th>NETVSSI</th>
<th>NETVSSI</th>
<th>NSW</th>
<th>NETVAUTO</th>
<th>NETVAUTO</th>
<th>AUTOMGR</th>
<th>NSW</th>
<th>SO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NETVIEW</th>
<th>NETVIEW</th>
<th>NETVIEW</th>
<th>NSW</th>
<th>SO</th>
<th>APPC</th>
<th>APPC</th>
<th>APPC</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ICSF</th>
<th>ICSF</th>
<th>CSF</th>
<th>NSW</th>
<th>TN3270</th>
<th>TN3270</th>
<th>TN3270</th>
<th>NSW</th>
<th>SO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>RACF</th>
<th>RACF</th>
<th>RACF</th>
<th>NSW</th>
<th>SO</th>
<th>SDSFAUX</th>
<th>SDSFAUX</th>
<th>SDSFAUX</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CRON1</th>
<th>STEP1</th>
<th>SYSTASK</th>
<th>OWT</th>
<th>AO</th>
<th>IM2HCTRL</th>
<th>IM2HCTRL</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>IM3HCTRL</th>
<th>IM3HCTRL</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>S</th>
<th>INETD1</th>
<th>STEP1</th>
<th>SYSTASK</th>
<th>OWT</th>
<th>AO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>IM3HDLI</th>
<th>IM3HDLI</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>S</th>
<th>IM3HDBC</th>
<th>IM3HDBC</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>IM2HDLI</th>
<th>IM2HDLI</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>S</th>
<th>IM2HDBC</th>
<th>IM2HDBC</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>RMF</th>
<th>RMF</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>S</th>
<th>SYSLOGD</th>
<th>STEP1</th>
<th>SYSTASK</th>
<th>NSW</th>
<th>AO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>NET</th>
<th>NET</th>
<th>NETVIEW</th>
<th>NSW</th>
<th>SO</th>
<th>ZOWESVR</th>
<th>ZOWESVR</th>
<th>ZOWESTEP</th>
<th>NSW</th>
<th>SO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ZWESIS01</th>
<th>ZWESIS01</th>
<th>ZWESIS01</th>
<th>NSW</th>
<th>SO</th>
<th>ZOWESVR6</th>
<th>*OMVSEX</th>
<th>IZUSVR</th>
<th>IN</th>
<th>AO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ZOWESVR9</th>
<th>*OMVSEX</th>
<th>IZUSVR</th>
<th>IN</th>
<th>AO</th>
<th>ZOWESVR3</th>
<th>*OMVSEX</th>
<th>IZUSVR</th>
<th>IN</th>
<th>AO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ZOWESVR6</th>
<th>*OMVSEX</th>
<th>IZUSVR</th>
<th>IN</th>
<th>AO</th>
<th>ZOWESVR9</th>
<th>*OMVSEX</th>
<th>IZUSVR</th>
<th>IN</th>
<th>AO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>RMFGAT</th>
<th>RMFGAT</th>
<th>IEFPROC</th>
<th>NSW</th>
<th>SO</th>
<th>MQVIEW</th>
<th>MQVIEW</th>
<th>MQVIEW</th>
<th>NSW</th>
<th>SO</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>MQNVSSI</th>
<th>MQNVSSI</th>
<th>NETVIEW</th>
<th>NSW</th>
<th>S</th>
<th>RSS</th>
<th>RSS</th>
<th>RSS</th>
<th>RSS</th>
<th>NSW</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TCPIP</th>
<th>TCPIP</th>
<th>TCPIP</th>
<th>NSW</th>
<th>SO</th>
<th>TSO</th>
<th>TSO</th>
<th>STEP1</th>
<th>OWT</th>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TWST</th>
<th>TWST</th>
<th>OPCT</th>
<th>NSW</th>
<th>S</th>
<th>VAMP</th>
<th>VAMP</th>
<th>VAMP</th>
<th>OWT</th>
<th>S</th>
</tr>
</thead>
</table>

| DJ27MSTR | DJ27MSTR | IEFPROC | NSW | S | DK27MSTR | DK27MSTR | DK27MSTR | IEFPROC | NSW | S |
Issue MVS Command with grep

```
steven@mbp~$ zowe console issue cmd "d a,l" --zosmf-p 27sh | grep CHIN
QM27CHIN QM27CHIN PROCSTEP IN SO IM3HIRLM IM3HIRLM DXRJPROC NSW S
QJ27CHIN QJ27CHIN PROCSTEP IN SO DK27DBM1 DK27DBM1 IEFPROC NSW S
QG27CHIN QG27CHIN PROCSTEP IN SO FTPD21 STEP1 SYSTASK OWT AO
steven@mbp~$ zowe console issue cmd "d a,l" --zosmf-p 27sh | grep CHIN
```
List data sets

```
stevens-mbp:~ steven$ zowe zos-files list data-set "STEVENH.*"
STEVENH.DEMO.COBOL
STEVENH.DEMO.JCL
STEVENH.DEMO.MIGRATED
STEVENH.FLATFILE.NO.CONTENT
STEVENH.FLATFILE.WITH.CONTENT
STEVENH.JCL
STEVENH.JUNK
STEVENH.PDSE
STEVENH.SPF.ISPPROF
STEVENH.SPFLOG1.LIST
STEVENH.TEST
STEVENH.TEST.JCL
STEVENH.USER.LOG
STEVENH.VSAM
STEVENH.VSAM.DATA
STEVENH.VSAM.INDEX
```
List data set members

```
stevens-mbp:~ steven$ zowe zos-files list all-members "STEVENH.JCL"
FLATFILE
JUNK
SHORTJOB
VSAM
stevens-mbp:~ steven$
```
List jobs

stevens-mbp:~ steven$ zowe jobs list jobs --owner "*" --prefix "ZOWE*"
STC35722  ZOWEXXC ACTIVE
STC35639  ZOJEJC ACTIVE
STC35577  ZOJEJCL ACTIVE
STC35412  ZOJEJR ACTIVE
STC35366  ZOWESH ACTIVE
STC34094  ZOWECCS ACTIVE
STC33951  ZOWEASS ACTIVE
stevens-mbp:~ steven$
Get Job status

```
stevens-mbp:~ steven$ zowe jobs view job-status-by-jobid "JOB36103"
jobid:  JOB36103
retcode: CC 0000
jobname: SHORTJ
status:  OUTPUT
stevens-mbp:~ steven$
```
List Job spool files

```
stevens-mbp:~ steven$ zowe jobs list spool-files-by-jobid "JOB36103"
2 JESMSGLG  JES2
3 JESJCL    JES2
4 JESYSMSG  JES2
stevens-mbp:~ steven$ 
```
View Job spool file

```
15.18.50 JOB36103 15.18.50 JOB36103 15.18.50 JOB36103 $HASP373 SHORTJ STARTED - INIT 2 - CLASS A - SYS MV3B
15.18.50 JOB36103 IEF403I SHORTJ - STARTED
15.18.55 JOB36103 --TIMINGS (MINS.)--
15.18.55 JOB36103 ----PAGING COUNTS---
15.18.55 JOB36103 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB CLOC
15.18.55 JOB36103 SHORTJ STEP1 00 194 .00 .00 .00 .0
8 352 0 0 0 0 0 1
15.18.55 JOB36103 IEF404I SHORTJ - ENDED
15.18.55 JOB36103 -SHORTJ ENDED. NAME- TOTAL CPU TIME=
 .00 TOTAL ELAPSED TIME=.08
15.18.55 JOB36103 $HASP395 SHORTJ ENDED - RC=0000
0------- JES2 JOB STATISTICS -------
- 13 MAY 2019 JOB EXECUTION DATE
- 3 CARDS READ
- 40 SYSOUT PRINT RECORDS
```
Visual Studio Code Extension for Zowe

The Visual Studio Code (VSC) Extension for Zowe lets you interact with data sets that are stored on IBM i/OS mainframes. You can explore data sets, view their contents, make changes, and upload the changes to the mainframe. Interacting with data sets from VSC can be more convenient than using command-line interfaces or 3270 emulators.

Important: To use the VSC Extension for Zowe, you must install Zowe CLI version 2.8.8 or later.

The VSC Extension for Zowe is powered by Zowe CLI. The extension demonstrates the potential for plug-ins powered by Zowe.

Contents
- Prerequisites
- Configuration and usage tips
- Sample use cases

Tip: For information about how to install the extension from a VSIX file and run system tests on the extension, see the Developer README file that is located in the docs folder of this repository.

Prerequisites
After you install the Zowe extension, meet the following prerequisites:

- Install Zowe CLI on your PC.

Important: To use the VSC Extension for Zowe, you must install Zowe CLI version 2.8.8 or later.
Mainframe Assembler & Zowe: “Hello World” Example

Friday, March 15, 8:30 AM - 9:30 AM
☆ [Zowe] Modern to Mainframe (Testing Automation and Orchestration)
Room: Room 101A
Session Number: 24023
Speaker(s)
Dan Kelosky.
DEMOPDS="STEVENH.DEMO.JCL"
ZOSMF_PROFILE=3bsh
# Check and see if pdp already exists
MATCHES='zowe zos-files list data-set "$DEMOPDS" --zosmf-p $ZOSMF_PROFILE --response-format=json | jq -r '.data.apiResponse.returnedRows'
if [ $MATCHES -gt 0 ]; then
  echo "Data set $DEMOPDS already exists, deleting"
  zowe zos-files delete data-set -f "$DEMOPDS" --zosmf-p $ZOSMF_PROFILE
fi

zowe zos-files create data-set-classic $DEMOPDS --zosmf-p $ZOSMF_PROFILE
zowe zos-files upload stdin-to-data-set "$DEMOPDS(INPUT)" <<< $1 --zosmf-p $ZOSMF_PROFILE
zowe zos-files upload stdin-to-data-set --zosmf-p $ZOSMF_PROFILE "$DEMOPDS(COPY)" <<EOF
//COPY JOB 123456, 'TSTRADM', NOTIFY='TSTRADM',
//    CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)
//STEP1
//EXEC PGM=IEBGENER
//SYSPRINT DD SYSPUT=A
//SYSIN
//DD DUMMY
//SYSUT1
//DD DISP=SHR,DSN=$DEMOPDS(INPUT)
//SYSUT2
//DD DISP=SHR,DSN=$DEMOPDS(OUTPUT)
//STEP2
//EXEC PGM=AOPBATCH,PARM='sleep 5'
EOF

JOBID='zowe jobs submit data-set "$DEMOPDS(copy)" --zosmf-p $ZOSMF_PROFILE --response-format-json | jq -r '.data.jobid'

echo "JOBID is $JOBID"

i="0"
while [ $i -lt 5 ]
do
  sleep 1s
  STATUS='zowe jobs view job-status-by-jobid $JOBID --response-format-json --zosmf-p $ZOSMF_PROFILE | jq -r '.data.status'
  if [ "$STATUS" = "OUTPUT" ]; then
    echo "Job $JOBID has now completed"
    i=5
  else
    echo "Waiting for job output to complete. Current status is $STATUS"
  fi
  i=$[i+1]
done
Data set STEVENH.DEMO.JCL already exists, deleting
Data set deleted successfully.
Data set created successfully.
success: true
from: stdin
to: STEVENH.DEMO.JCL(INPUT)

Data set uploaded successfully.
success: true
from: stdin
to: STEVENH.DEMO.JCL(COPY)

Data set uploaded successfully.

JOBID is JOB36947
Waiting for job output to complete. Current status is ACTIVE
Waiting for job output to complete. Current status is ACTIVE
Waiting for job output to complete. Current status is ACTIVE
Job JOB36947 has now completed
Job JOB36947's spool files:
2  JESMSGLG  JES2
3  JESJEL  JES2
4  JESYSMSG  JES2
101  SYSPRINT  STEP1


1  JES2  JOB LOG -- SYSTEM MV3B -- NO
   DE W IN M V S 3 B
0
15.12.48  JOB36947  ----  MONDAY, 20 MAY 2019 ----
15.12.48  JOB36947  IRR0101 USERID STEVENH IS ASSIGNED TO THIS JOB.
15.12.50  JOB36947  ICH700011 STEVENH LAST ACCESS AT 15:12:35 ON MONDAY, MAY 20, 2019
15.12.50  JOB36947  $HASP373 COPY - STARTED - INIT 2 - CLASS A - SYS MV3B
15.12.50  JOB36947  IEF403I COPY - STARTED
15.12.50  JOB36947  --TIMINGS (MIN
S.)--  ----PAGING COUNTS---
Web Desktop

Zowe.org
We are building more than just technology, we are building a community
This is a procedure to start the ZOE web server and Node server. This procedure requires a WebSphere Liberty Angel procedure to be running, such as z/OSMF procedure "IZUANG".

Invoke this procedure, specifying the path where the ZOE server is installed on your system.

ZOESVR,SRVRPATH='/u/chunli/zowe/0.8.3/explorer-server'

SRVRPATH - The path to the HFS directory where the Atlas server was installed.

/export export SYMLIST=

Start the node server

ZOESVR PRB SRVRPATH='/u/chunli/zowe/0.8.3/explorer-server'

SRVRPATH - The path to the HFS directory where the Atlas server was installed.

/export export SYMLIST=

Optional logging parameters that can be configured if required
INSTALL_DIR=/u/stevenh/zowe-0.9.3/

ZOSMF_VERSION=""

ZOS_RELEASE=`INSTALL_DIR/scripts/opercmd 'd iplinfo' | grep RELEASE`

ZOS_VRM= echo $ZOS_RELEASE | sed 's+$RELEASE z/OS \(........\).*$'

if [[ $ZOS_VRM == "02.03.08" ]]
then
    ZOSMF_VERSION=2.3.0
elif [[ $ZOS_VRM == "02.02.08" ]]
then
    ZOSMF_VERSION=2.2.0
fi

echo $ZOSMF_VERSION

echo $ZOSMF_DOC_URL
INSTALL_DIR=/utilities/zowe/7.9.3/

if [[ "$205_VARM" == "02.03.00" ]]
then
  #elmet_f2 $205_VARM
  ZOSMF_VERSION="2.3.0"
else
  #elmet_f1 $205_VARM
  ZOSMF_VERSION="2.2.0"
fi

echo "$205_VERSION"
echo "$205_DOC_URL"
API Mediation Layer

Zowe.org
We are building more than just technology, we are building a community
Why do I need an API gateway? - endpoint consolidation
Why do I need an API gateway? - credential challenges

Client A

3270 client B

HTTPs:8080

TSO

Server A

Web

Server E

REST API

Server C

Command Line

Client D

Another HTML

Client E

HTTPs:26504

HTTPs:32070

HTTPs:1234

HTTPs:32070

HTTPS:23

HTTPS:26504

HTTPS:8080

Telnet:23

Welcome to z/OS

Why do I need an API gateway? - credential challenges
Why do I need an API gateway? - certificate management
Why do I need an API gateway?
What is the Zowe API Mediation Layer?

Zowe Desktop

- VT Terminals
- TN 3270
- MVS Explorer
- JES Explorer
- USS Explorer

API Gateway

- API Catalog
- API Discovery
- File API
- JES API

- z/OSMF
  - REST APIs
  - Workflows
  - TSO Commands
  - MVS Commands
  - Files, USS, JES

8544
- ZLUX
- TN3270
- VT

8542
- zssServer
- X-MEM

8543
- ZWESIS01

7554
Extending Zowe API gateway services?

In progress

API Gateway

REST APIs
Workflows
TSO Commands
MVS Commands
Files, USS, JES

Existing z/OS Address space with REST API
Static Registration

New REST API
Dynamic Discovery with Springboot Eureka

z/OSFS
MVS UI
JES UI
USS UI

8542
zssServer

X-MEM
ZWESIS01

8544
ZLUX

TN3270
VT

8554

7552
API Catalog

7553
API Discovery

7554

7443
z/OSFS

8547
File API

8545
JES API

8548
MVS UI

8546
JES UI

8550
USS UI
Dynamic discovery service based on Netflix Eureka framework

API servers can be statically defined through .yaml files or else REST API calls to the gateway.

API Catalog lists API servers on its “Southbound edge”

Available API services

- API Mediation Layer API
  The API Mediation Layer for z/OS internal API services. The API Mediation Layer provides a single point of access to mainframe REST APIs and offers enterprise cloud-like feature...
  - All services are running

- z/OS Datasets services
  IBM z/OS Datasets REST services
  - All services are running

- z/OS Jobs services
  IBM z/OS Jobs REST services
  - All services are running

- z/OSMF services
  IBM z/OS Management Facility REST services
  - All services are running

My Server APIs
Expose z/OS Connect EE APIs in Zowe API Mediation Layer

By openmainframe | May 27, 2019 | Blog

Exposing the MQ REST API via the Zowe API Mediation Layer

MattLeming
Published on 17 May 2019 / Updated on 20 May 2019
# API Catalog

## Data Sets APIs

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POST</strong></td>
<td>/api/v1/datasets</td>
<td>Create a data set</td>
</tr>
<tr>
<td><strong>DELETE</strong></td>
<td>/api/v1/datasets/{dataSetName}</td>
<td>Delete a data set or member</td>
</tr>
<tr>
<td><strong>GET</strong></td>
<td>/api/v1/datasets/{dataSetName}/content</td>
<td>Get the content of a sequential data set, or PDS member</td>
</tr>
<tr>
<td><strong>PUT</strong></td>
<td>/api/v1/datasets/{dataSetName}/content</td>
<td>Sets the content of a sequential data set, or PDS member</td>
</tr>
<tr>
<td><strong>GET</strong></td>
<td>/api/v1/datasets/{dataSetName}/members</td>
<td>Get a list of members for a partitioned data set</td>
</tr>
<tr>
<td><strong>GET</strong></td>
<td>/api/v1/datasets/{filter}</td>
<td>Get a list of data sets matching the filter</td>
</tr>
</tbody>
</table>

## System APIs

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GET</strong></td>
<td>/api/v1/datasets/username</td>
<td>Get current userid</td>
</tr>
</tbody>
</table>

## Unix Files APIs
SLACKBOTTEST

#slackbot-test

- colin.stone
- general
- random
- slackbot-test

DIRECT MESSAGES
- slackbot
- colin.stone (you)
- andersl
- itvmir

Atlas-mainframe APP 1:29 PM
Abend or Abnormality noticed on system

Message #slackbot-test

4:57 AM
Job Started - Owner: JCAIN Name: ATlj0001 JobId: JOB09972

Job Finished: Owner: JCAIN Name: ATlj0001 JobId: JOB09972 RC: ABEND SOC9

- JESMSGLG
- JESJCL
- JESYSMSG
- SYSOUT
- CEEDUMP

Job Finished: Owner: STONECC Name: TSTJCICS JobId: JOB09971 RC: CC 0000

- JESMSGLG
- JESJCL
- JESYSMSG
Zowe futures

Enterprise installer through SMP/E
More CLI plugins for base z/OS. CICS, DB2, IMS, MQ, zCEE, …
More tools’ REST API catalogs in the API gateway. CICS, IMS, zCEE,
Bottom up configuration
Multi Factor Authentication and SSO using JWT
SDK – App Store
How to become a Zoweian
#zowe-user

**Tuesday, March 26th**

- **zbotzer**: 2:45 PM
  - joined #zowe-user.

- **hogstrom**: 9:18 PM
  - ZLC topics for tomorrow, please drop them in the ZLC Channel.

- **Mike Rosek**: 11:15 PM
  - joined #zowe-user along with Atsushi Manabe.

**Yesterday**

- **Atsushi Manabe**: 10:22 AM
  - Hi all,
  - I would like to install Zowe to IBM Z environment without Internet connection. However, the version of NPM that is the prerequisite of Zowe is V5.4 or later, and I could not find a z/OS version that can be installed locally. Does anyone have experience?

**Today**

- **Nana Yamamoto**: 3:44 AM
  - joined #zowe-user.