



z/OS Basics: JES3 for New Users

David Jones
IBM – JES3 Product Team

August 2, 2010
Session 6818 – 4:30 PM-5:30 PM



SHARE in Boston



Trademarks

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

- IBM®
- MVST™
- Redbooks®
- RETAIN®
- z/OS®
- zSeries®

The following are trademarks or registered trademarks of other companies.

- Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* 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.

SHARE inBoston



Objective

- Session description:
 - This session is intended for those who are new to the platform or just new to JES3. This session will aid those who are trying to find the status of their work in the system by issuing JES3 commands and understanding the results. Once they know the status, they will then be shown how to manipulate the job and/or output to their satisfaction.
- My objective is to provide an overview of the JES3 workflow and identify related JES3 statements and commands.
- Just about any one section or topic can be an entire hour long session!



Acknowledgement

My personal Thank You to Greg Thompson who helped to steer and review the contents of this presentation!



JES3 DSPs

- JES3 is composed of routines called Dynamic Support Programs (DSPs).
- DSPs are scheduled and processed the same as jobs.
- System programmers can alter DSP processing with installation exit routines.
- System programmers can add DSPs or replace existing DSPs.
- Some of the DSPs mentioned are:
 - Internal reader – INTRDR
 - Disk reader – DR
 - Converter/interpreter – CI

SHARE in Boston

5

- *z/OS V1R12.0 JES3 Introduction – Glossary*

dynamic support program (DSP). Multiprogrammed system components that are scheduled by the job segment scheduler (JSS) and implement a specified function. A DSP can be related to job execution, such as main service or output service, or it can be a background utility, such as the dump job facility.

- *ABCs of z/OS System Programming - Volume 13 – Chapter 15*

JES3 is composed of routines called dynamic support programs, (DSPs). With the same mechanisms it uses for processing jobs, JES3 schedules and processes DSPs. JES3 users can write their own dynamic support programs to customize JES3 processing in their system.

Most of the JES3 programs in the global processor is divided into parts called dynamic support programs, or DSPs. There are DSPs for reading job input, for processing jobs, and for writing job output. What distinguishes DSPs from ordinary routines or subroutines is that DSPs are schedulable units. Before a DSP is executed, it must be scheduled by JES3. (DSPs have priorities that govern their position in a JES3 dispatching queue.)

System programmers can alter what DSPs do (with installation exit routines), or they can write new DSPs to supplement or replace the DSPs shipped with JES3.



JES3 FSSs

- Function Subsystems (FSSs) offload function to separate address spaces on the local and global processors.
 - Printers
 - Converter Interpreter

SHARE in Boston

6

- *z/OS V1R12.0 JES3 Introduction – Glossary*

functional subsystem (FSS). An extension of JES that runs in an address space separate from the JES address space. An FSS provides support for a function peripheral to JES processing, such as a peripheral device or other component.

- *ABCs of z/OS System Programming - Volume 13 – Chapter 1*

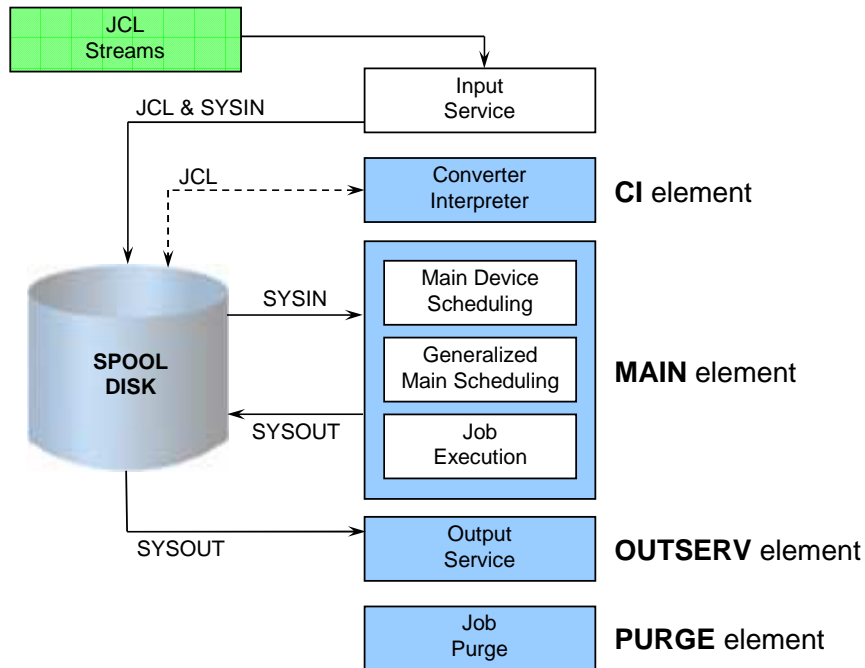
A functional subsystem (FSS) is a collection of programs residing in an address space separate from JES that communicates with JES to provide a JES-related function, such as print processing and converter/interpreter (C/I) processing for JES3. An FSS extends the scope of JES processing. Because an FSS operates in its own address space, it functions independently of JES in several areas.

An FSS is responsible for:

- The management of storage resources that it needs during data set processing including print buffers.
- Its own recovery and serviceability.
- Its performance and accounting measurements.
- The security of its own resources.

JES and the FSS communicate through the functional subsystem interface (FSI) and SSI. The FSI is a one-level interface which provides two way communication. The FSI consists of a set of macro-invoked service routines provided by both JES and the FSS/FSA.

JES3 MVS job flow and scheduler elements



SHARE in Boston

7

- *ABCs of z/OS System Programming - Volume 13 – Chapter 3*

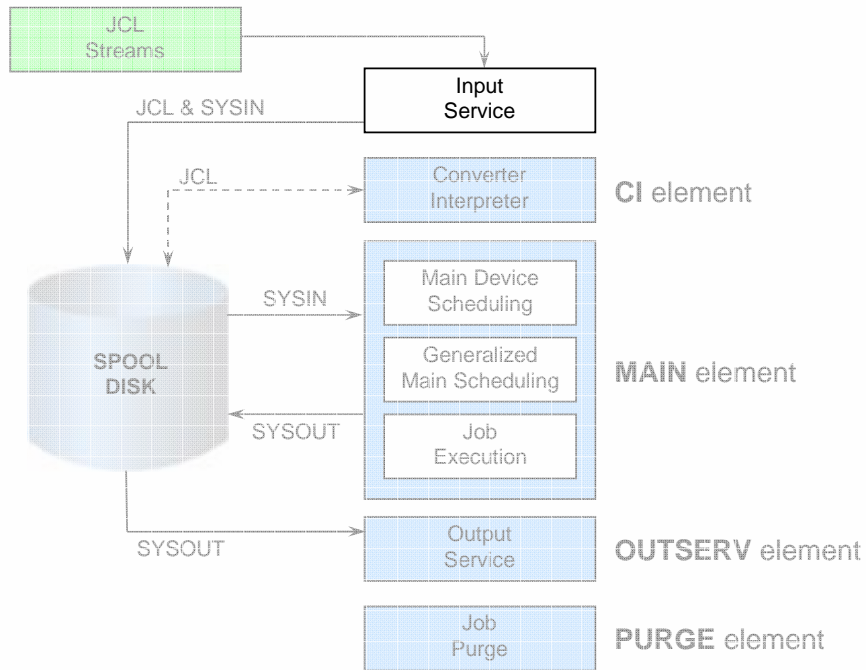
JES3 MVS job flow

JES3 processing performed on behalf of jobs consists of distinct phases. A typical MVS job entering the JES3 complex is set by the input service to go through the following phases (scheduler elements (SE)):

- **Converter/Interpreter (CI)** - Interpretation of the job's JCL to a form usable by MVS (SWA - Scheduler Work Area).
- **Main Device Scheduling (MDS)** - Pre-execution resource allocation to insure I/O resources needed by the job are available, followed by passing (Generalized Main Scheduling - GMS) the job to MVS execution.
- **Output Service** - Output processing that handles the job's SYSOUT data.
- **Purge** - Removal the job from the system.

Scheduler elements

Each of the four phases of a job is represented by a scheduler element (SE) in the JCT entry. Every scheduler element denotes a unit of JES3 work (DSP) JES3 must perform when processing the job. The JES3 job segment scheduler (JSS) selects SEs (perhaps from several jobs) that are ready for processing, obtains an RQ for the job, and builds entries in the function control table (FCT) (if they do not already exist) so that DSP can be dispatched to do the work.



SHARE in Boston

First a few items regarding the first phase of a job workflow – Input Service.



Input Service

- The JES3 function that accepts and queues all jobs
 - Processes JCL control statements
 - Creates scheduler elements that represent work to be executed by JES3
 - Adds jobs to the JES3 job queue
- JES3 global processor reads jobs from:
 - A local card reader
 - A local tape reader
 - A disk reader
 - An internal reader
 - A remote work station
 - Another node in a job entry network
 - MVS START, MOUNT, and LOGON commands

SHARE in Boston

9

- *ABCs of z/OS System Programming - Volume 13 – Chapter 6*

Input service is the JES3 function that accepts and queues all jobs, entering the JES3 system, except those invoked via the *X command. When jobs are sent to a JES3 global, JES3 input service does the following:

- Processes JCL control statements
- Creates scheduler elements that represent work to be executed by JES3
- Adds jobs to the JES3 job queue.

The global processor reads the job into the system from one of the following input sources:

- A local card reader
- A local tape reader
- A disk reader
- A remote work station
- Another node in a job entry network
- The internal reader
- Started tasks

The reader phase of input service reads jobs (JCL and input stream data) and stores them on a spool data set. The only jobs not read by the reader phase are jobs from an internal reader and demand select jobs. These jobs are read directly by the control statement processing phase. Jobs can come from a card reader, a tape unit, a disk reader or from a remote workstation. The reader phase treats jobs from a remote workstation as though the job came from a card reader.

See also *z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4*.

Input Service: Disk reader definition



- The disk reader DSP (DR) processes control statements from a partitioned data set member.
- Specified in the JES3 start procedure:
 - `//JES3DRDS DD DSN=SYS1.JES3DR ,DISP=SHR`
- Or with the DYNALLOC initialization statement:
 - `DYNALLOC,DDN=JES3DRDS,DSN=SYS1.JES3DR`
- Can enter all JES3 operator commands (except `*DUMP` and `*RETURN`)
- Good for production JCL streams.

SHARE in Boston

10

- *ABCs of z/OS System Programming - Volume 13 – Chapter 6*

Disk reader

If the disk reader facility (DR) is desired, specify in the JES3 start procedure:

```
//JES3DRDS DD DISP=SHR,DSN=dsn
```

or use the DYNALLOC statement in the initialization to specify the disk reader data set:

```
DYNALLOC,DDN=JES3DRDS,DSN=dsn
```

The disk reader data set is a the partitioned data sets containing input for the JES3 disk reader facility. The maximum block size for this data set is 3200. Concatenated data sets may be used.

You can enter all JES3 commands (except ***DUMP** and ***RETURN**) from a card reader (CR) or disk reader (DR). You can use the disk reader to enter repetitive commands based on system requirements (such as shift change). Any output messages generated from a card reader or disk reader are displayed at the console from which you called the reader.

One of the main uses of the disk reader is to read in a production JCL stream that you place into the disk reader data set.

Input Service: Disk reader commands



- *X DR,M=mem
 - Call a disk reader to initiate processing of a disk reader data set member.
 - K parameter can be used to keep the reader active (hot disk reader) after reading the member.

```
*x dr,m=wtor
IAT6306 JOB (JOB00019) IS DR          , CALLED BY C3E0SY1
IAT6201 JOB DRDR -01 (JOB00020),01 JOBS,00004 CDS,P=15
IAT7450 JOB DR          (JOB00019) PURGED
IAT6100 (JOB00020) JOB WTORJ      (JOB00021), PRTY=02, ID=*UNKNOWN
IAT7450 JOB DRDR -01 (JOB00020) PURGED
. . .
SY1 WTORJ      IAT2000 JOB WTORJ      (JOB00021) SELECTED SY1      SRVCLASS=DISCRETN
SY1 WTORJ      IEF403I WTORJ - STARTED - TIME=16.33.40
```

SHARE in Boston

11

Use the *CALL,DR command to initiate reading of jobs from a direct-access storage device (DASD).

If the K parameter is specified on the *CALL, *START, or *CANCEL command, the disk reader remains active when an EOF condition is reached, and reading resumes each time an *START command is entered.

Reader processing

Reader processing takes place in the JES3 global address space. The reader phase reads jobs and places the jobs on JES3 SPOOL in batches for later processing. For each reader batch, an input service job is created that prepares and places the job in a queue for further processing.

Example here calls the WTOR member of the disk reader data set SYS1.JES3DR.

JOB00019 is the disk reader (DR) DSP which reads the JCL for one or more jobs from the member and places them on SPOOL. In this case only for the WTOR member,

JOB00020 is the input service job started for the disk reader that places the jobs into a queue for further processing. In this case only for the WTOR member.

JOB00021 is the job that has been started for the WTOR member.

See *z/OS V1R12.0 JES3 Commands* for more details.

Input Service: Disk reader commands



- ***S DR,M=mem**
 - Resume a hot disk reader to initiate processing of a disk reader data set member.
 - Can add or alter parameters specified from the previous disk reader command.
- ***C DR**
 - Cancel disk reading immediately and keep the reader active (hot disk reader).
 - Use KN to completely purge the disk reader.

SHARE in Boston

12

*START,DR command to start a disk reader.

*CANCEL,DR command to cancel a disk reader.

See *z/OS V1R12.0 JES3 Commands* for more details.

Input Service: Internal reader definition



- The internal reader DSP (INTRDR) processes control statements from an input stream...
 - ...created by a job as system output
 - ...from the TSO SUBMIT command
- **OPTIONS** initialization statement
 - INTRDR= specifies maximum active internal readers.

- *ABCs of z/OS System Programming - Volume 13 – Chapter 6*

INTRDR processing

Input may also come from the JES3 internal reader that processes input streams contained in SYSOUT data sets obtained from MVS. The internal reader allows a JES3 output data set to be passed to JES3 input service and be processed as an input stream. In this way, jobs can also be submitted to JES3 from MVS.

START and **MOUNT** commands and TSO/E LOGONs cause jobs to be started from predefined procedures. Input service processes the JCL created for these jobs in the same manner as any other standard job. Jobs initially placed on direct-access storage devices (DASD) and subsequently analyzed by JES3 input service are placed on the JES3 spool.

Input Service: Internal reader commands



- *I X,D=INTRDR
 - Display INTRDR DSP information

```
*i x,d=intrdr
IAT8475 (IQDX) - INTRDR  MXCT=000020 USE=000001 MOD=YES
IAT8472 (IQDX) - INTRDR  IS NOT IN HOLD
```

- *I A,D=INTRDR
 - Display active INTRDR DSPs

```
*i a,d=intrdr
IAT8522 JOB INTRDR  (JOB00009) ACTIVE ON INTRDR          0064.95 MIN
IAT8523 INTRDR COUNTS - MAX=(020,020), ACT=0001, FCT=001, HWATER=0001
IAT8593 INQUIRY ON ACTIVE JOBS COMPLETE,          1 JOB  DISPLAYED
```

*INQUIRY,X,D=INTRDR to display INTRDR DSP information.

In the example, MXCT is maximum possible DSPs in the JES3 global address space – USE is the current active DSPs.

*INQUIRY A,D=INTRDR to display active jobs on the INTRDR DSP.

The active jobs are the jobs being processed by the DSP. For INTRDR, this includes the INTRDR job itself. MAX is the current maximum number of DSPs allowed (same as MAXCNT above) and the initialization value.

See *z/OS V1R12.0 JES3 Commands* for more details.

Input Service: Internal reader commands



- *F X,D=INTRDR,HOLD
- *F X,D=INTRDR,RELEASE
 - Modify INTRDR DSP to stop or start processing
- *F X,D=INTRDR,MC=25
 - Modify INTRDR DSP to maximum count of 25

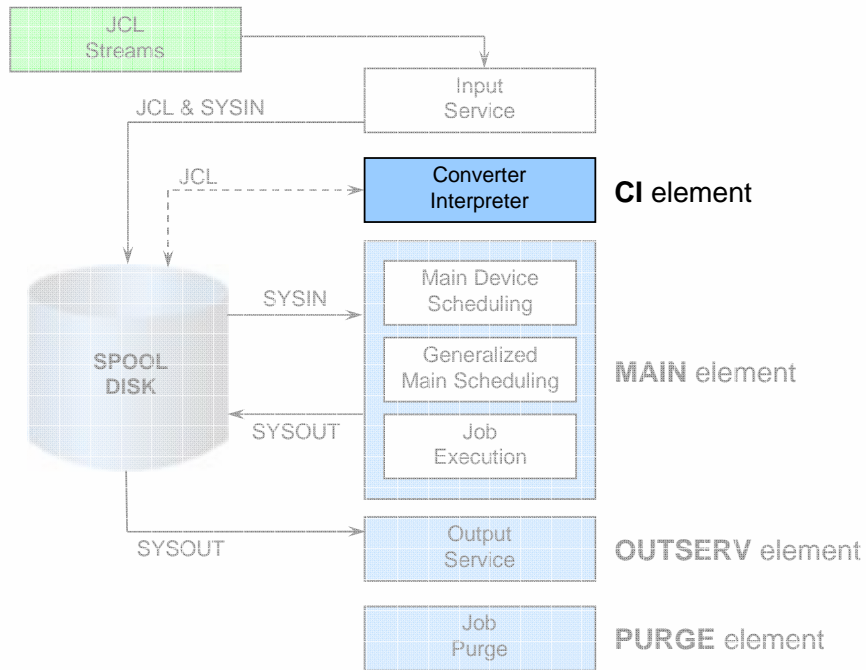
```
*f x,d=intrdr,hold
IAT8497 (MODX) - INTRDR   HELD
*i x,d=intrdr
IAT8475 (IQDX) - INTRDR   MXCT=000020 USE=000001 MOD=YES
IAT8472 (IQDX) - INTRDR   IS IN HOLD
*f,x,d=intrdr,mc=25,release
IAT8484 (MODX) - INTRDR   OLDMXCT=000020 NEWMXCT=25
IAT8497 (MODX) - INTRDR   RELEASED
*i x,d=intrdr
IAT8475 (IQDX) - INTRDR   MXCT=000025 USE=000001 MOD=YES
IAT8472 (IQDX) - INTRDR   IS NOT IN HOLD
```

*MODIFY X,D=INTRDR to modify the INTRDR DSP.

An INTRDR DSP can be held to stop processing and released to continue.

The maximum number of DSPs running in the JES3 global address space can be changed – raised or lowered.

See *z/OS V1R12.0 JES3 Commands* for more details.



SHARE in Boston

After Input Service brings the job in and prepares the job on SPOOL, it is ready for the second step in the workflow – Converter/Interpreter (C/I).



Converter Interpreter

- First scheduler element for a standard job - **CI**.
- Converter Interpreter (C/I) controls the conversion of information from JCL statements to internal control blocks for use by the system.
 - Job is printed and purged if JCL syntax or logic errors are found.
 - Job's device, volume, and data set requirements are collected for pre-execution setup.
- CI in the JES3 global address space.
 - CI and POSTSCAN DSPs.
- CI FSS
 - Can be used to off-load overhead from the JES3 global address space – including on the global!
 - CI DSP for interpreter and prescan phases.

SHARE in Boston

17

- z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4.

Converter/Interpreter (C/I) service controls the conversion of JCL statements to Converter/Interpreter text and then into control blocks. This service comprises primarily the JES3 CI and POSTSCAN dynamic support programs, the C/I subtasks under which the MVS C/I routines run, and the initiator support routines.

C/I service controls the conversion and interpretation of a job's JCL. The three principal phases of C/I service are:

- *Converter/Interpreter phase*: Uses the MVS C/I routines to convert and interpret JCL into scheduler control blocks. At this time, the scheduler control blocks are created in the scheduler work area (SWA).
- *Prescan phase*: Extracts job requirements from the scheduler control blocks for use in the postscan phase. At the end of the prescan phase, the scheduler control blocks are moved from the SWA to JES3 spool.
- *Postscan phase*: Locates data sets and gathers information for use in JES3 device setup.

You can define one or more C/I functional subsystem (FSS) address spaces to perform the converter/interpreter, prescan, and locate (catalog search) portion of the postscan phase for some or all jobs. A C/I FSS can operate on the global or any local main.

Converter Interpreter: CI definition



- **STANDARDS initialization statement**
 - CIBATCH= can be used to limit on which processors C/I for batch jobs is performed (applies to CI FSSs).
 - CIDEMAND= can be used to limit on which processors C/I for demand select jobs is performed (applies to CI FSSs).
 - CICNT= specifies the maximum number of CI DSPs in the JES3 global address space.
 - PSTCNT= specifies the maximum number of POSTSCAN DSPs in the JES3 global address space.
- **CIPARM initialization statement**
 - Specifies the options to be used for the MVS C/I.

Initialization statements in the JES3 INISH deck associated with defining CI.

Use the STANDARDS initialization statement to specify default values for information not provided on other JES3 initialization statements or on the `//*FORMAT` JES3 control statement. It also provides standards to be applied to all jobs entering the system.

See *z/OS V1R12.0 JES3 Initialization and Tuning Reference* for more details.

Converter Interpreter: CI related commands



- *I X,D=CI
 - Display information for CI DSP in the JES3 global address space

```
*i x,dsp=ci
IAT8475 (IQDX) - CI      MXCT=(000009,000009) USE=(000000,000000) MOD=YES
IAT8472 (IQDX) - CI      IS NOT IN HOLD
IAT8487 (IQDX) - ADDR SPACE JCL LIMIT 00000000; JOB JCL LIMIT 00000000
```

- *I Q,D=CI
 - Display jobs waiting for a CI DSP

```
*i q,d=ci
IAT8674 JOB IBMUSERA (JOB00025) P=02 CL=W      CI
IAT8595 INQUIRY ON JOB QUEUE STATUS COMPLETE,  1 JOB  DISPLAYED
```

- *I A,D=CI
 - Display jobs active in CI

```
*i a,d=ci
IAT8522 JOB IBMUSERA (JOB00025) ACTIVE ON CI      0000.01 MIN
IAT8593 INQUIRY ON ACTIVE JOBS COMPLETE,          1 JOB  DISPLAYED
```

SHARE in Boston

19

*INQUIRY,X,D=CI to display CI DSP information.

In the example, MXCT is maximum possible DSPs that can be concurrently active in the JES3 global address space.

USE is the current active DSPs.

First values are the DSPs counts for batch jobs, the second values are the DSP counts for TDO logons and started task.

*INQUIRY Q,D=CI is the list of jobs waiting for the CI DSP.

In the example, we have JOB00025 waiting to be processed by a CI DSP.

This situation was created by holding the CI DSP.

*INQUIRY A,D=CI to display active jobs on the CI DSP.

For the CI DSP this is the number of jobs active in C/I.

This was captured by releasing the CI DSP and then immediately doing the inquiry.

Like the INTRDR DSP, can also use *MODIFY X,D=CI command to HOLD and RELEASE CI activity.

*MODIFY X,D=CI can be used to modify the counts for the CI DSP in the JES3 global address space.

See *z/OS V1R12.0 JES3 Commands* for more details.

Converter Interpreter: Putting it together



- Job active in JES3 global address space CI
 - *CALL DISPLAY command used to display information about jobs in the JES3 job queue.

```
*i a,d=ci
IAT8522 JOB IBMUSERA (JOB00035 ACTIVE ON CI 0000.01 MIN
IAT8593 INQUIRY ON ACTIVE JOBS COMPLETE. 1 JOB DISPLAYED

*call display,j=35
IAT6306 JOB (JOB00036) IS DISPLAY , CALLED BY C3E0SY1
IAT7762 - 02 JOB00035 IBMUSERA SE=(CI-ACTIVE,MAIN-NOSTAT,
IAT7762 - OUTSERV-NOSTAT,PURGE-NOSTAT),JCTMLINS=(SY1,SY2,SY3,
IAT7762 - SY4,SY5,SY6,SY7,SYLOCAL8),
IAT7762 - ORG=ANYLOCAL,CLASS=W,GROUP=JES3WLM,JTRACKS=00000002,
IAT7762 - IORATE=MED,PROCLIB=IATPLBI1,MAINS=(SY1,SY2,SY3,SY4,
IAT7762 - SY5,
IAT7762 - SY6,SY7,SYLOCAL8),LINEST=999,PAGEST=09999999,
IAT7762 - BYTEST=00001500,CRDEST=999,JESMSG LG=LOG,
IAT7762 - OUSID=IBMUSER,
IAT7762 - TUSID=IBMUSER
IAT7450 JOB DISPLAY (JOB00036) PURGED
```

SHARE in Boston

20

Periodically we will try to put together the command information with the position of the job in the workflow. Here we show a job in C/I.

This slide also introduces the *CALL DISPLAY command which can be used to obtain additional job details.

This example we have JOB00035 being processed by a CI DSPI. With *CALL DISPLAY we see the position of the job within the scheduler elements (SE). Here we see again that the job is active in CI. If the job was on the queue waiting for CI, then SE status would be CI-NOSTAT; no status since the job has not entered CI. When the CI portion of the job workflow is done, SE status is CI-COMPLETE.

Converter Interpreter: CI FSS definition



- FSSDEF initialization statement
 - One statement with TYPE=CI for each CI FSS to be created.
 - FFSNAME= identifies the FSS for other commands (e.g. *MODIFY).
 - PNAME= specifies the procedure library member for starting the FSS – CI FSS default is JES3CI.
 - SYSTEM= specifies the JES3 mains on which the FSS address space can run – default is the global processor.
 - DSPCNT= specifies the maximum number of CI DSPs in the FSS address space for batch and demand select jobs.
 - Other parameters as well!

Use the FSSDEF statement to define the characteristics of one or more C/I FSSes, which operate in their own address spaces.

See *z/OS V1R12.0 JES3 Initialization and Tuning Reference* for more details.

Converter Interpreter: CI FSS related commands



- *I F,TYPE=CI
 - Display information for all CI FSSs
- *F F,FSS=fssname...
 - Modify a CI FSS

*INQUIRY F,TYPE=CI to display information about all CI FSSes.

*MODIFY F,FSS= to modify a specific FSS.

See *z/OS V1R12.0 JES3 Commands* for more details.

Converter Interpreter: Putting it together



- Start a CI FSS

```

*i f,type=ci
IAT8701 FSSNAME TYP SYSTEM PROCNAME JOBID STAT T S MD RC
IAT8701 DSP/DEV MAXASST
IAT8702 CIFSS1 C/I SY1 JES3CI NONE INAC N N
IAT8702 002,001 00000000
IAT8702 CIFSS2 C/I SY1 JES3CI NONE INAC N N
IAT8702 005,000 00000000
IAT8702 CIFSS3 C/I NONE JES3CI NONE INAC N N
IAT8702 002,001 00000000
. . .

*f f,fss=cifss1,dsp=(3,3)
IAT8310 MODIFY COMPLETE FOR FSS CIFSS1 - KEYWORD DSP=

*f f,fss=cifss1,st=y
IAT8310 MODIFY COMPLETE FOR FSS CIFSS1 - KEYWORD ST=
IAT6100 ( DEMSEL ) JOB JES3CI (JOB00026), PRTY=15, ID=IBMUSER
IAT6372 FSS CIFSS1 START-UP ON SY1 SUCCESSFUL
SY1 JES3CI IEF403I JES3CI - STARTED - TIME=15.01.16
IXZ0001I CONNECTION TO JESXCF COMPONENT ESTABLISHED,
GROUP NODE1 MEMBER CIFSS1
IAT4452 CI FSS CIFSS1 IS READY FOR WORK

*i f,type=ci
IAT8701 FSSNAME TYP SYSTEM PROCNAME JOBID STAT T S MD RC
IAT8701 DSP/DEV MAXASST
IAT8702 CIFSS1 C/I SY1 JES3CI JOB00026 ACTV N Y
IAT8702 003,003 00000000
. . .

```

SHARE in Boston

23

These CI FSSes were defined with START=NO so all are INAC.

DSP/DEV column shows the DSP count for a CI FSS and shows the device name for a WTR FSS.

The first DSP count value is the number of CI DSPs that can concurrently process batch jobs,

The second value is the number of CI DSPs that can concurrently process demand select jobs (TSO logon and started tasks),

First modify example modified a CI FSS to change the DSP counts.

More important is the second modify which was used to start a CI FSS. The started CI FSS is running JES3CI, from the proclib, in JOB00026.

Converter Interpreter: Putting it together



- Job active on a CI FSS

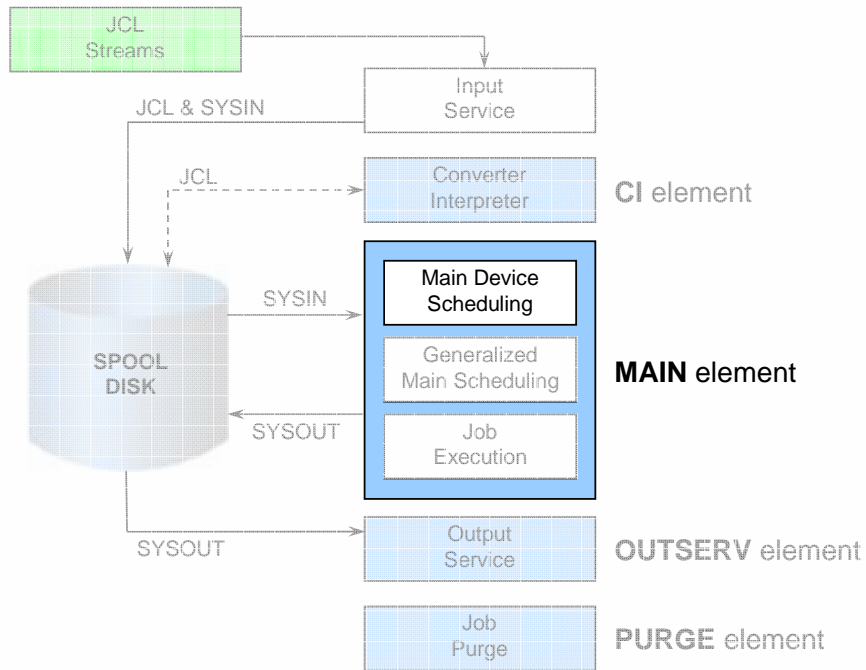
```

*i a,d=ci
IAT8522 JOB IBMUSERA (JOB00038) ACTIVE ON CIFSS1 0000.00 MIN
IAT8593 INQUIRY ON ACTIVE JOBS COMPLETE. 1 JOB DISPLAYED

*call display,j=38
IAT6306 JOB (JOB00039) IS DISPLAY , CALLED BY C3E0SY1
IAT7762 - 02 JOB00038 IBMUSERA SE=(CI-ACTIVE,MAIN-NOSTAT,
IAT7762 - OUTSERV-NOSTAT,PURGE-NOSTAT) .ICTM.LINS=(SY1,SY2,SY3,
IAT7762 - SY4,SY5,SY6,SY7,SYLOCAL8),
IAT7762 - ORG=ANYLOCAL,CLASS=W,GROUP=JES3WLM,JTRACKS=00000002,
IAT7762 - IORATE=MED,PROCLIB=IATPLBI1,MAINS=(SY1,SY2,SY3,SY4,
IAT7762 - SY5,
IAT7762 - SY6,SY7,SYLOCAL8),INDEX=ACTIVE IN C/I IN AN FSS,
IAT7762 - LINES=999,PAGEST=09999999,BYTEST=00001500,
IAT7762 - CRDEST=999,
IAT7762 - JESMSG LG=LOG,OUSID=IBMUSER,TUSID=IBMUSER
IAT7450 JOB DISPLAY (JOB00039) PURGED
  
```

Here again we did a *INQUIRY for the CI DSP and we see we have JOB00035 being processed by a CI DSP – in this case in the running in a CI FSS – CIFSS1 specifically – rather than in the CI in the JES3 global address space as we saw earlier.

Again, with *CALL DISPLAY we see the position of the job within the scheduler elements (SE) – active in CI.



Once C/I processing has been completed without any failures, the job can move to the MAIN SE with Main Device Scheduling.



Main Device Scheduling

- After C/I a standard job is moved to the second scheduler element – **MAIN**.
- Main Device Scheduling (MDS) performs setup which ensures that resources (devices, volumes, and data sets) needed by the job are allocated before the job is processed.

- *z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4.*

JES3 provides a device management facility called the main device scheduler (MDS) that can wholly or partially support the MVS allocation process. The purpose of MDS is to satisfy job resource requirements (the devices, volumes, and data sets needed) before and during job execution, thus allowing execution to proceed without allocation delays. MDS also allows controlled multisystem access to commonly accessible data sets in the loosely coupled environment.

Main Device Scheduling: MDS definition



- STANDARDS initialization statement
 - SETUP= specifies the type of setup processing.
 - NONE specifies that no pre-execution setup is to occur. All devices are allocated, mounted, and deallocated by MVS.
- SETNAME and DEVICE initialization statements
 - Identify the devices to be managed by MDS.
- SETPARAM initialization statement
 - Specifies MDS parameters and indicates how devices are to be managed.
 - Ignored if SETUP=NONE was specified.
 - Parameters...

SHARE in Boston

27

Initialization statements in the JES3 INISH deck associated with MDS definition.

Another use of the STANDARDS initialization statement.

Types of setup:

•Job setup

Before a job executes, JES3 reserves all devices needed and mounts all volumes needed.

After the job starts to execute, it does not have to wait between job steps to have a device allocated or a volume mounted.

•High watermark setup

JES3 considers the needs of each step and reserves the maximum number of each device type required across the steps.

Reduces the number of devices JES3 reserves for a job.

•Explicit setup

Combines the execution advantages of job setup and the device usage advantages of high watermark setup.

See *z/OS V1R12.0 JES3 Initialization and Tuning Reference* for more details.

Main Device Scheduling: MDS definition



- SETPARAM parameters:
 - FETCH= specifies whether MDS is to issue volume fetch messages.
 - NO then ALLOCATE=AUTO is assumed.
 - ALLOCATE= specifies whether automatic allocation immediately follows volume fetch.
 - MANUAL suspends jobs after volume fetch until operator continues them.
 - AUTO specifies MDS attempts allocation of resources and runs system select for SMS-managed resources.
 - SMSSETUP= specifies whether JES3 manages SMS data sets.
 - If not specified, MVS determines whether JES3 manages SMS data sets.

No notes or references for this page.

Main Device Scheduling: MDS related commands



- *I S
 - Display status of all mains in the complex and a summary of the MDS queues.
 - Good first step if JES3 may be sick.
- *I MAIN=*main*
 - Display the information about a main processor, the global processor, or all main processors.

*INQUIRY,S command to display the status of all mains in the complex and a summary of the MDS queues.

*INQUIRY MAIN= command to display information about a main specific processor, the global processor, or all main processors.

See *z/OS V1R12.0 JES3 Commands* for more details.

Main Device Scheduling: Putting it together



- Status of mains

```

*i s
IAT5619 ALLOCATION QUEUE = 0000000 BREAKDOWN QUEUE = 0000000
IAT5619 SYSTEM SELECT QUEUE = 0000000 ERROR QUEUE = 0000000
IAT5619 SYSTEM VERIFY QUEUE = 0000000 FETCH QUEUE = 0000000
IAT5619 UNAVAILABLE QUEUE = 0000000 RESTART QUEUE = 0000000
IAT5619 WAIT VOLUME QUEUE = 0000000 VERIFY QUEUE = 0000000
IAT5619 ALLOCATION TYPE = AUTO
IAT5619 CURRENT SETUP DEPTH - ALL PROCESSORS = 0000000
IAT5619 MAIN NAME STATUS SDEPTH DASD TAPE
IAT5619 SY1 ONLINE IPLD 255,000 00640,00000 00144,00000
IAT5619 SY2 ONLINE IPLD 255,000 00640,00000 00144,00000
IAT5619 SY3 ONLINE NOTIPLD 255,000 00640,00000 00144,00017
. . .
IAT5619 SYLOCAL8 ONLINE NOTIPLD 255,000 00640,00000 00144,00000
IAT5619 DEFERCT=NO
IAT5619 SDEPZERO=WAIT

*i main=all
IAT8643 MAIN INQUIRY RESPONSE
INFORMATION FOR MAINPROC SY1
FMID=HJS7770, STATUS=(ONLINE,CONNECTED,ATTACHED,GLOBAL)
INFORMATION FOR MAINPROC SY2
FMID=HJS7770, STATUS=(ONLINE,CONNECTED,ATTACHED,LOCAL)
INFORMATION FOR MAINPROC SY3
STATUS=(ONLINE,NOT-CONNECTED,DOWN,LOCAL)
. . .
INFORMATION FOR MAINPROC SYLOCAL8
STATUS=(ONLINE,NOT-CONNECTED,DOWN,LOCAL)
MAINPROC INQUIRY RESPONSE COMPLETE
  
```

SHARE in Boston

30

*INQUIRY S shows the status of the mains, but does not identify the global.

Use *I MAIN command get more detailed information about the mains. The status indicators returned for *I MAIN are:

- Online status, in which the following text can appear:
 - ONLINE** = The processor has been varied on line to the global, or was defined (explicitly or by default) as being initially on line to the global at initialization time.
 - OFFLINE** = The processor has been varied off line to the global, or was defined explicitly as being initially on line to the global at initialization time. The status OFFLINE is also displayed if the processor has been flushed.
- Connect status, in which the following text can appear:
 - CONNECTED** = The processor has connected to the global and is capable of receiving jobs.
 - NOT-CONNECTED** = The processor has not connected to the global.
- JES3 status as viewed by JESXCF, in which the following text can appear:
 - DOWN** = Either the processor is completely down, or JES3 has never attached to JESXCF since the processor was IPLed.
 - ATTACHED** = JES3 has initialized and attached to JESXCF.
 - NOT-ATTACHED** = JES3 was previously attached to JESXCF, but has been brought down.
 - JESXCF-ERROR** = The attach status of JES3 could not be obtained because of an error.
- Global status, in which the following text can appear:
 - GLOBAL** = The processor is the JES3 global.
 - LOCAL** = The processor is a JES3 local.

See z/OS V1R12.0 JES3 Messages for more details.

Main Device Scheduling: MDS related commands



- *S *main*,FLUSH
 - Flush jobs currently running from a JES3 local main for a failed processor.
- Recommended even if JES3 local is down for a short time.
 - Until the main is flushed or returns, new jobs may not run.
 - Jobs with job names that duplicate “active” jobs.
 - Data sets with DISP=OLD in “use” would delay other jobs requiring them.

When a system in the JESplex is brought down “normally,” such as an IPL, JES3 is made aware of it. However, if the system goes down abnormally, JES3 may not be aware of it and everything looks fine, but jobs are not running on that system. If the local processor failure is such that JES3 cannot be restarted, then the local processor must be logically removed from the JESplex.

FLUSH clears out JES3 use of the system that is no longer available.

See *z/OS V1R12.0 JES3 Commands* for more details.

Main Device Scheduling: Putting it together



- Flush a system

```

SY1          *ISG178E GLOBAL RESOURCE SERIALIZATION HAS BEEN DISRUPTED.  GLOBAL RESOURCE REQUESTORS WILL BE SUSPENDED.
*SY1        *I6 IXC402D SY2 LAST OPERATIVE AT 17:36:59.  REPLY DOWN AFTER SYSTEM RESET, OR INTERVAL=SSSSS TO SET A REPROMPT
TIME.

*i main=all
IAT8643 MAIN INQUIRY RESPONSE
INFORMATION FOR MAINPROC SY1
FMID=HJS7770, STATUS=(ONLINE,CONNECTED,ATTACHED,GLOBAL)
INFORMATION FOR MAINPROC SY2
FMID=HJS7770, STATUS=(ONLINE,CONNECTED,ATTACHED,LOCAL)
INFORMATION FOR MAINPROC SY3
STATUS=(ONLINE,NOT-CONNECTED,DOWN,LOCAL)
. . .
MAINPROC INQUIRY RESPONSE COMPLETE

16,down
SY1          IEE600I REPLY TO 16 IS:DOWN
. . .

*i main=sy2
IAT8643 MAIN INQUIRY RESPONSE
INFORMATION FOR MAINPROC SY2
FMID=HJS7770, STATUS=(ONLINE,CONNECTED,DOWN,LOCAL)
MAINPROC INQUIRY RESPONSE COMPLETE

*S sy2,flush
IAT2628 **S SY2,FLUSH ' ACCEPTED
IAT2006 PREMATURE JOB TERM - JOB SYSLOG (JOB00017) - CANCELED - SY2
IAT2006 PREMATURE JOB TERM - JOB RACF (JOB00020) - CANCELED - SY2
IAT2006 PREMATURE JOB TERM - JOB VTAM14 (JOB00022) - CANCELED - SY2
IAT2006 PREMATURE JOB TERM - JOB TCAS (JOB00024) - CANCELED - SY2
IAT7001 JOB SYSLOG (JOB00017) IS ON WRITER PRT002(0002),RECORDS=975
IAT7008 JOB SYSLOG(JOB00017) ON PRT002(0002), READ ERROR(DM702) FOR DSN=+MASTER+.SYSLOG.JOB00017.D000000A.?.
IAT7007 JOB SYSLOG (JOB00017) ON WRITER PRT002 (0002), DSN=+MASTER+.SYSLOG.JOB00017.D000000A.?, PURGED.

```

- Continued...

SHARE in Boston

32

First component to detect a system might be down is XCF – message IXC402D in this case.

As evidenced by *I MAIN=ALL, JES3 is unaware at this point that SY2 is down and so any jobs for SY2 are still active and associated resources are still allocated to those jobs,

Replying DOWN to IXC402D gets the system out of the sysplex, but not completely out of the JESplex.

JES3 is aware SY2 is down, but jobs for SY2 are still active and associated resources are still dedicated.

If the system is IPLed and JES3 restarted, then the appropriate cleanup and restart of the jobs will occur.

However, if the system might or will be down for an extended time, then tell JES3 to flush the jobs from that system.

Can also use MVS command D XCF to see what systems are in the SYSPLEX:

```
d xcf
```

```

SY1          IXC334I  15.08.59  DISPLAY XCF 379
          SYSPLEX PLEX1:      SY1          SY2

```


Main Device Scheduling: Putting it together



- Flushed system is IPLed and varied online.

```

*i main=sy2
IAT8643 MAIN INQUIRY RESPONSE
INFORMATION FOR MAINPROC SY2
  FMID=HJS7770, STATUS=(OFFLINE,FLUSHED,DOWN,LOCAL)
MAINPROC INQUIRY RESPONSE COMPLETE

*i main=sy2
IAT8643 MAIN INQUIRY RESPONSE
INFORMATION FOR MAINPROC SY2
  FMID=HJS7770, STATUS=(OFFLINE,FLUSHED,ATTACHED,LOCAL)
MAINPROC INQUIRY RESPONSE COMPLETE

*v sy2,online
IAT8180 SY2      VARIED ONLINE  TO JES3 ON GLOBAL
IAT6100 ( DEMSEL ) JOB SYSLOG   (JOB00058), PRTY=15, ID=+MASTER+
IAT6100 ( DEMSEL ) JOB MOUNT    (JOB00059), PRTY=11, ID=IBMUSER
IAT6100 ( DEMSEL ) JOB RACF     (JOB00061), PRTY=15, ID=IBMUSER
IAT6100 ( DEMSEL ) JOB BPXAS    (JOB00060), PRTY=15, ID=OEDFLT
IAT7001 JOB SYSLOG (JOB00058) IS ON WRITER PRT002(0002),RECORDS=2313
IAT7007 JOB SYSLOG (JOB00058) ON WRITER PRT002 (0002), DSN=+MASTER+.SYSLOG.JOB00058.D0000009.?,
PURGED.
IAT7007 JOB SYSLOG (JOB00058) ON WRITER PRT002 (0002), DSN=+MASTER+.SYSLOG.JOB00058.D000000A.?,
PURGED.

*i main=sy2
IAT8643 MAIN INQUIRY RESPONSE
INFORMATION FOR MAINPROC SY2
  FMID=HJS7770, STATUS=(ONLINE,CONNECTED,ATTACHED,LOCAL)
MAINPROC INQUIRY RESPONSE COMPLETE

```

SHARE in Boston

33

Now SY2 is completely down, flushed, and also OFFLINE.

Need to eventually vary the local main, like a device, online to JES3. Otherwise, JES3 ignores the local main when it tries to reconnect.

With an IPL of SY2 and restart of JES3 as a local, the main will connect with the local and once again be in the JESplex.

The vary online can be done before or after the local main has been restarted. In this case SY2 was IPLed and JES3 restarted before the vary was done.

First inquiry was after the flush (same result after SY2 is IPLed but before JES3 is restarted).

Second inquiry is after SY2 is IPLed and JES3 has been started. Note SY2 is attached.

Vary on and a final inquiry – all back to normal.

If the vary on was done before SY2 was IPLed, then the result would have been:

```

*v sy2,online
IAT8180 SY2      VARIED ONLINE  TO JES3 ON GLOBAL

*i main=all
IAT8643 MAIN INQUIRY RESPONSE
INFORMATION FOR MAINPROC SY1
  FMID=HJS7770, STATUS=(ONLINE,CONNECTED,ATTACHED,GLOBAL)
INFORMATION FOR MAINPROC SY2
  FMID=HJS7770, STATUS=(ONLINE,NOT-CONNECTED,DOWN,LOCAL)

```

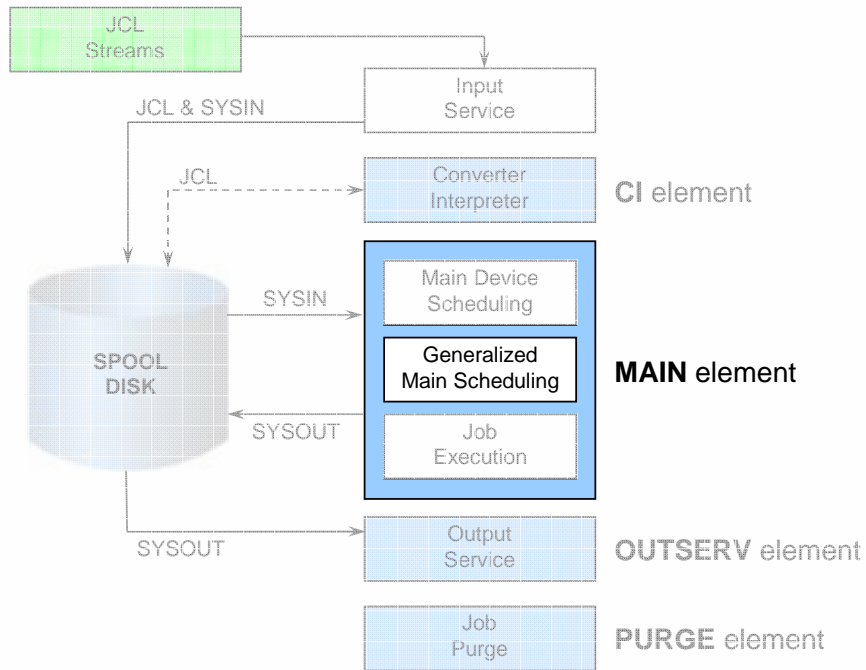
Main Device Scheduling: MDS related commands



- *I S,...
 - Use various parameters and options to display the status of jobs currently in setup or the status of volumes and data sets controlled by MDS.
 - A displays jobs in the allocate queue which are competing for resources.
 - A,SUMM displays summary of jobs in MDS allocation, including the resources that are required and the number of jobs that require each resource.
 - J=*jobo* displays information about a job on an MDS queue.
 - D=*dsn* displays data set status.
 - V=ALL,E displays currently managed volumes and serials.
 - V=*volser*,E displays the status of a volume and all job references to the volume.
 - Many more!

*INQUIRY S has many more parameters! Some of these could be used to determine what is going on with MDS processing of jobs.

See *z/OS V1R12.0 JES3 Commands* for more details.



SHARE in Boston

Still in the MAIN SE, once MDS has determine the JES3 managed resources needed by the job are available, the job moves to Generalized Main Scheduling (GMS).



Generalized Main Scheduling

- Second phase of the second scheduler element – **MAIN**.
- Generalized Main Scheduling (GMS) selects and schedules a job for execution each time an MVS initiator requests work.
- Job selection depends primarily upon initialization parameters specified.
- GMS features:
 - Deadline scheduling
 - Dependent Job Control (DJC)

- *z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4.*

Each time an MVS initiator requests work, generalized main scheduling (GMS) selects and schedules a job for execution. The job that GMS selects depends primarily upon initialization parameters that you have specified.

Deadline scheduling and dependent job control (DJC), additional GMS functions, enable you to control when jobs execute. With deadline scheduling, you specify a deadline by which you want the job to run. JES3 periodically increases the job's selection priority in an attempt to run the job by the specified deadline. DJC allows you to create a network of related jobs.

Generalized Main Scheduling: JES and WLM managed initiators



- JES3 or WLM control of initiators is specified in the job class group.
- WLM managed initiator job selection is based upon the performance goals and classification rules in the WLM policy.
 - Fewer and simpler externals.
 - WLM adapts to changing system conditions to meet the objectives.
- JES managed initiator job selection is driven by how job processing is defined in the GROUP, CLASS, and SELECT initialization statements.
 - More controls to specify the number of initiators, mix of jobs, and so forth based upon resources, not performance goals.
 - The decisions are static and require intervention to be changed.

Generalized Main Scheduling: GMS definition



- MAINPROC initialization statement
 - Defines a processor as a JES3 main.
 - SELECT= Specifies the name of the job selection mode to be initially assigned to the main.
- SELECT initialization statement
 - Defines the scheduling controls associated with a particular job selection mode.
 - Specifies job classes and job class groups that can execute.
 - Additional parameters exist to affect the job selection algorithm.

Initialization statements in the JES3 INISH deck associated with GMS definition.

See *z/OS V1R12.0 JES3 Initialization and Tuning Reference* for more details.

Generalized Main Scheduling: GMS definition



- GROUP initialization statement
 - Defines the characteristics of a JES3 job class group.
 - MODE= specifies WLM or JES managed initiators for the group.
 - EXRESC= specifies execution resources assigned to the group.
- CLASS initialization statement
 - Defines the characteristics of JES3 job classes.
 - GROUP= assigns the job class to a job class group.
 - Other JES3 unique parameters to control job scheduling for the class:
 - MDEPTH, TDEPTH, SDEPTH, MLIMIT, TLIMIT
 - Not covered in this presentation.

One way to think of JES3 GROUP is that it is used to define the initiators available and where. The definition include the mode (JES3 or WLM), the mains the initiators can run on, and a name tat can be used to assign job classes to the GROUP.

CLASS defines the job classes that can be used for jobs. The class specifies the GROUP, or set of initiators , than the job can be scheduled to, but the also the systems where the job class can be scheduled to run on. In addition there are many other JES3 unique options, not covered here, that are used to control the number of jobs scheduled within the class.

See *z/OS V1R12.0 JES3 Initialization and Tuning Reference* for more details.

Generalized Main Scheduling: GMS related commands



- *I G,*main*,... to obtain the status of GMS.
 - Specify ALL for *main* for every processor in the JESplex.
- *I G,ALL,S
 - Displays the current SELECT mode options, but not the job classes or job class groups (not real interesting).
- *I G,ALL,G
 - Displays the execution resource status of the job class groups.
- *I G,ALL,C
 - Displays the status of the job classes.
- *I C=*class*
 - Displays information about job classes.

SHARE in Boston

40

*INQUIRY,G, command to obtain the status of GMS components of JES3.

*INQUIRY CLASS= command to display information about job classes.

See *z/OS V1R12.0 JES3 Commands* for more details.

Generalized Main Scheduling: Putting it together



- Job class groups and job classes.

```

*i g,all,g
IAT8932 GROUP - JES3TEST - STATUS=ON   DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY1
IAT8932 GROUP - JES3HOLD - STATUS=OFF  DI=0000 AI=0000 UI=0000 ALLOC=MAN UNAL=DYN BAR=NO JSPAN=ALL MODE=JES - SY1
IAT8932 GROUP - JES3WLM - STATUS=ON    DI=0000 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=WLM - SY1
IAT8932 GROUP - JS3BATCH - STATUS=OFF  DI=0002 AI=0000 UI=0000 ALLOC=DEM UNAL=DEM BAR=NO JSPAN=ALL MODE=JES - SY1
IAT8932 GROUP - JES3TEST - STATUS=ON   DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY2
IAT8932 GROUP - JES3HOLD - STATUS=OFF  DI=0000 AI=0000 UI=0000 ALLOC=MAN UNAL=DYN BAR=NO JSPAN=ALL MODE=JES - SY2
IAT8932 GROUP - JES3WLM - STATUS=ON    DI=0000 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=WLM - SY2
IAT8932 GROUP - JS3BATCH - STATUS=OFF  DI=0002 AI=0000 UI=0000 ALLOC=DEM UNAL=DEM BAR=NO JSPAN=ALL MODE=JES - SY2
.
.
IAT8932 GROUP - JES3TEST - STATUS=ON   DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SYLOCAL8
IAT8932 GROUP - JES3HOLD - STATUS=OFF  DI=0000 AI=0000 UI=0000 ALLOC=MAN UNAL=DYN BAR=NO JSPAN=ALL MODE=JES - SYLOCAL8
IAT8932 GROUP - JES3WLM - STATUS=ON    DI=0000 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=WLM - SYLOCAL8
IAT8932 GROUP - JS3BATCH - STATUS=OFF  DI=0002 AI=0000 UI=0000 ALLOC=DEM UNAL=DEM BAR=NO JSPAN=ALL MODE=JES - SYLOCAL8
IAT8599 INQUIRY ON GMS COMPLETE

*i g,all,c
IAT8934 CLASS - A - STATUS=ON - GRP=JES3TEST - SY1
IAT8934 CLASS - B - STATUS=ON - GRP=JES3TEST - SY1
IAT8934 CLASS - C - STATUS=ON - GRP=JES3HOLD - SY1
IAT8934 CLASS - TEMP - STATUS=OFF - GRP=JES3TEST - SY1
IAT8934 CLASS - W - STATUS=ON - GRP=JES3WLM - SY1
IAT8934 CLASS - WLMCLASS - STATUS=ON - GRP=JES3WLM - SY1
IAT8934 CLASS - A - STATUS=ON - GRP=JES3TEST - SY2
IAT8934 CLASS - B - STATUS=ON - GRP=JES3TEST - SY2
IAT8934 CLASS - C - STATUS=ON - GRP=JES3HOLD - SY2
IAT8934 CLASS - TEMP - STATUS=OFF - GRP=JES3TEST - SY2
IAT8934 CLASS - W - STATUS=ON - GRP=JES3WLM - SY2
IAT8934 CLASS - WLMCLASS - STATUS=ON - GRP=JES3WLM - SY2
.
.
IAT8934 CLASS - A - STATUS=ON - GRP=JES3TEST - SYLOCAL8
IAT8934 CLASS - B - STATUS=ON - GRP=JES3TEST - SYLOCAL8
IAT8934 CLASS - C - STATUS=ON - GRP=JES3HOLD - SYLOCAL8
IAT8934 CLASS - TEMP - STATUS=OFF - GRP=JES3TEST - SYLOCAL8
IAT8934 CLASS - W - STATUS=ON - GRP=JES3WLM - SYLOCAL8
IAT8934 CLASS - WLMCLASS - STATUS=ON - GRP=JES3WLM - SYLOCAL8
IAT8599 INQUIRY ON GMS COMPLETE
  
```

SHARE inBoston

41

Display job class groups.

Note WLM or JES3 managed groups.

Note STATUS indicates whether or not enabled for the main.

Group JS3BATCH is not enabled.

Display job classes.

Note STATUS indicates whether or not enabled for the main.

Class TEMP is not enabled.

JES3 supports class names up to 8 characters.

JOB JCL statement does not.

Add another JECL statement to use job classes greater than one character:

```
/**MAIN CLASS=TEMP
```

Generalized Main Scheduling: Putting it together



- Enabled is not all it seems to be.

```

*i c=a
IAT8609 CLASS INQUIRY INFORMATION
INFORMATION FOR CLASS A
GROUP=JES3TEST (JES), SPART=NONE, DEFAULT=NO
DEFINED ON SY1, SY2, SY3, SY4, SY5, SY6, SY7, SYLOCAL8
ENABLED ON SY1, SY2, SY3, SY4, SY5, SY6, SY7, SYLOCAL8
LIMIT SYSTEM/CLASS          MAXIMUM    CURRENT
SDEPTH ----                255        0
SPIN=NO

*i c=b
IAT8609 CLASS INQUIRY INFORMATION
INFORMATION FOR CLASS B
GROUP=JES3TEST (JES), SPART=NONE, DEFAULT=NO
DEFINED ON SY1, SY2, SY3, SY4, SY5, SY6, SY7, SYLOCAL8
DISABLED ON ALL SYSTEMS
LIMIT SYSTEM/CLASS          MAXIMUM    CURRENT
SDEPTH ----                255        0
SPIN=NO

*i c=temp
IAT8609 CLASS INQUIRY INFORMATION
INFORMATION FOR CLASS TEMP
GROUP=JS3BATCH (JES), SPART=NONE, DEFAULT=NO
DEFINED ON SY1, SY2, SY3, SY4, SY5, SY6, SY7, SYLOCAL8
ENABLED ON SY1, SY2, SY3, SY4, SY5, SY6, SY7, SYLOCAL8
LIMIT SYSTEM/CLASS          MAXIMUM    CURRENT
SDEPTH ----                255        0
SPIN=NO
  
```

SHARE in Boston

42

Display job class information for 3 different classes.

Class A is enabled on all systems, and the job class group it is assigned to, JES3TEST, is also enabled on all systems. Hence jobs with class A can be scheduled to all systems.

Class B is disabled on all systems. Hence jobs with class B cannot be scheduled even though the job class group it is assigned to, JES3TEST, is enabled on all systems.

Class TEMP is enabled on all systems, but the job class group it is assigned to, JS3BATCH, is disabled on all systems. Hence jobs with class TEMP cannot be scheduled.

Have to look carefully at both job class and job class group to determine if jobs with a specific class can be scheduled.

The job class group is the set of initiators – if they are not enabled, then anything assigned to them cannot be scheduled.

The job classes are assigned to initiators.

Generalized Main Scheduling: Putting it together



- Job waiting for something to be enabled.

```

*i j=47
IAT8674 JOB IBMUSERA (JOB00047) P=02 CL=TEMP      MAIN
IAT8699 INQUIRY ON JOB STATUS COMPLETE,          1 JOB DISPLAYED

*x display,j=47
IAT6306 JOB (JOB00048) IS DISPLAY  CALLED BY C3EUSY1
IAT7762 - 02 JOB00047 IBMUSERA SE (CI-COMplete,MAIN-NOSTAT)
IAT7762 - OUTSERV-NOSTAT,PURGE-NOSTAT),JCCMAINS=(SY1,SY2,SY3,
IAT7762 - SY4,SY5,SY6,SY7,SYLOCAL8),
IAT7762 - SEVCLASS=DISCRETN,ORG=ANYLOCAL,CLASS=TEMP,
IAT7762 - GROUP=JS3BATCH,TRACKS=00000008,IORATE=MED,
IAT7762 - POCLES=EXPER
IAT7762 - WAITING FOR A MAIN/CLASS/GROUP,LINEST=999,
IAT7762 - PAGES1=09999999,BYTES1=00001300,CRDEST=999,
IAT7762 - JESMSG LG=LOG,
IAT7762 - OUSID=IBMUSER,TUSID=IBMUSER
IAT7450 JOB DISPLAY (JOB00048) PURGED
  
```

For the *I result we see MAIN rather than MAIN(EXECUTING-SY1), so it is not executing in an initiator. *X DISPLAY gives more detail. The job submitted with class TEMP is waiting for something to be enabled. It will run when we later enable the job class group.

Generalized Main Scheduling: GMS related commands



- *F G,*main*,... to modify the status of GMS.
 - Specify ALL for *main* for every processor in the JESplex.
- *F G,ALL,G,*group*,MODE,JES | WLM
 - Change initiator management for *group* (must use ALL).
- *F G,*main*,G,*group*,INIT,*nnn*
 - Change the number of dedicated initiators for a JES managed group.
- *F G,*main*,G,*group*,ALLOC,*opt*
 - Change group initiator allocation option for a JES managed group.
- *F G,*main*,G,*group*,ON|OFF
 - Enable or disable a group

*MODIFY,G, command to alter the status of the GMS component of JES3.

See *z/OS V1R12.0 JES3 Commands* for more details.

Generalized Main Scheduling: Putting it together



- Modify the initiator management (must use ALL).

```

*i g,all,g,jes3wlm
IAT8932 GROUP - JES3WLM - STATUS=ON  MODE=WLM - SY1
IAT8932 GROUP - JES3WLM - STATUS=ON  MODE=WLM - SY2
. . .
IAT8932 GROUP - JES3WLM - STATUS=ON  MODE=WLM  SYLOCAL8
IAT8599 INQUIRY ON GMS COMPLETE

*f g,all,g,jes3wlm,mode,jes
IAT8103 MODE CHANGED TO JES FOR GROUP JES3WLM

*i g,all,g,jes3wlm
*i g,all,g,jes3wlm
IAT8932 GROUP - JES3WLM - STATUS=ON . . . MODE=JES - SY1
IAT8932 GROUP - JES3WLM - STATUS=ON . . . MODE=JES - SY2
. . .
IAT8932 GROUP - JES3WLM - STATUS=ON . . . MODE=JES  SYLOCAL8
IAT8599 INQUIRY ON GMS COMPLETE

```

SHARE in Boston

45

Change initiator management from WLM to JES3.

Not real interesting, but could be done. This is why when defining a GROUP as WLM, even JES3 related parameters can/should be specified. They will not be used for a WLM managed group, but come into play if the group management is changed.

Note that ALL must be specified for the main when changing the mode!

Generalized Main Scheduling: Putting it together



- Modify the number of dedicated initiators for a JES managed group.

```

*i g,all,g,jes3test
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY1
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY2
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY3
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY4
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY5
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY6
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY7
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SYLOCAL8
IAT8599 INQUIRY ON GMS COMPLETE

*f g,sy3,g,jes3test,init,10
IAT8456 GMS MODIFY - COMPLETE NO ERRORS - SY3

*i g,all,g,jes3test
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY1
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY2
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0010 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY3
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY4
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY5
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY6
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SY7
IAT8932 GROUP - JES3TEST - STATUS=ON DI=0003 AI=0000 UI=0000 ALLOC=DEM UNAL=MAN BAR=NO JSPAN=ALL MODE=JES - SYLOCAL8
IAT8599 INQUIRY ON GMS COMPLETE
  
```

An example of changing the number of dedicated initiators for a JES3 managed GROUP. Maybe a need to increase the number of jobs that could concurrently run on a particular system for some reason. Possibly because a set of much needed resources are available on that system. Maybe to help distribute the workload more. Many possible reasons.

Generalized Main Scheduling: Putting it together



- About that job with CLASS=TEMP
 - Enable the job class group.

```

*x display,j=47
IAT6306 JOB (JOB00048) IS DISPLAY , CALLED BY C3E0SY1
IAT7762 - 02 JOB00047 IBMUSERA SE=(CI-COMplete,MAIN-NOSTAT,
IAT7762 - OUTSERV-NOSTAT,PURGE-NOSTAT),JCTMANS=(SY1,SY2,SY3,
IAT7762 - SY4,SY5,SY6,SY7,SYLOCAL8),
IAT7762 - SYVCLASS=DISCRETN,ORG=ANYLOCAL,CLASS=TEMP,
IAT7762 - GROUP=JS3BATCH,JSRACKS=00000008,IORATE=MED,
IAT7762 - PROCLIB=IATPLD11,
IAT7762 - WAITING FOR A MAIN/CLASS/GROUP,LINEST=999,
IAT7762 - PAGEST=09999999,BYTEST=00001500,CRDEST=999,
IAT7762 - JESMSGLOG=LOG,
IAT7762 - OUSID=IBMUSER,TUSID=IBMUSER
IAT7450 JOB DISPLAY (JOB00048) PURGED

*f g,sy3,g,js3batch,on
IAT8456 GMS MODIFY - COMPLETE NO ERRORS - SY3

*f g,sy2,g,js3batch,on
IAT8456 GMS MODIFY - COMPLETE NO ERRORS - SY2
IAT6100 ( DEMSEL ) JOB INITJES3 (JOB00049), PRTY=15, ID=INIT
IAT2001 OS MVS WAITING FOR WORK - SY2
IAT7450 JOB INITJES3 (INT00049) PURGED
IAT7001 JOB IBMUSERA (JOB00047) IS ON WRITER PRT002(0002),RECORDS=18

*i g,sy2,g,js3batch
IAT8932 GROUP - JS3BATCH - STATUS=ON DI=0002 AI=0000 UI=0000 ALLOC=DEM UNAL=DEM BA=NO JSPAN=ALL MODE=JES - SY2
IAT8599 INQUIRY ON GMS COMPLETE
  
```

SHARE in Boston

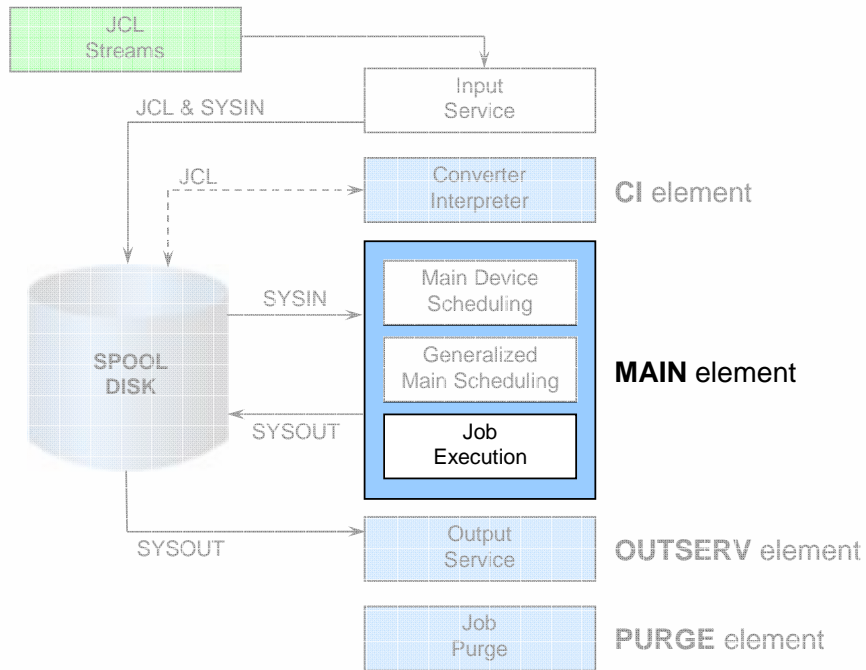
47

Job submitted with class TEMP is waiting because we found the job class group JS3BATCH disabled on all systems.

Enabling JS3BATCH SY3 did nothing as SY3 is not online!

Enabling JS3BATCH on SY2 resulted in an initiator being started, JOB INITJES3 (JOB00049), job 47 running, then the initiator being purged.

Fell into an example of a JES3 initiator being allocated and unallocated on demand.



SHARE in Boston

Once all the resources are made available with MDS, and the job is scheduled to run somewhere, GMS, is ready for execution.



Job Execution

- Last phase of the second scheduler element – **MAIN**.
- The job selected processed (executed) by the MVS initiator.
- While the job is executing, JES3 commands can be used to check status, modify the job, and cancel the job.
- After the job execution, MDS Breakdown stage makes the resources available for another job.

No notes or references for this page.

Job Execution: Job related commands



- *I A,...
 - Display the status of active jobs.
 - Many options to select which active jobs are displayed.
 - D= used to display jobs being processed by a DSP.
- *I B,...
 - Display the number of jobs backlogged.
 - Jobs that have been scheduled for main service or are waiting to be scheduled for main service.
 - Options exist to select which backlogged jobs are displayed.
- *I J=...
 - Display the status of specific jobs
 - Provides different detail than *I A.
 - Can specify a pattern such as * to get all jobs.

SHARE in Boston

50

*INQUIRY,A command to display the status of active jobs.

*INQUIRY,B command to display job backlogged.

*INQUIRY,J command to display the status of one or more jobs.

See *z/OS V1R12.0 JES3 Commands* for more details.

Job Execution: Putting it together



- Active jobs.

```

*i a
IAT8524 JOB RACF      (JOB00004)      ON SY1      003089.28 MIN
IAT8524 JOB SYSLOG   (JOB00002)      ON SY1      003089.28 MIN
IAT8524 JOB VTAMJ4   (JOB00012) VTAMJ4 ON SY1      003088.55 MIN
IAT8524 JOB TCAS     (JOB00013) TCAS   ON SY1      003088.33 MIN
IAT8524 JOB JES3CI   (JOB00041) CIFSS1 ON SY1      000007.60 MIN
IAT8524 JOB SYSLOG   (JOB00017)      ON SY2      003081.75 MIN
IAT8524 JOB RACF     (JOB00020)      ON SY2      003081.58 MIN
IAT8524 JOB VTAMJ4   (JOB00022) VTAMJ4 ON SY2      003081.30 MIN
IAT8524 JOB TCAS     (JOB00024) TCAS   ON SY2      003081.13 MIN
IAT8499 NO JOBS ACTIVE ON SY3
IAT8499 NO JOBS ACTIVE ON SY4
IAT8499 NO JOBS ACTIVE ON SY5
IAT8499 NO JOBS ACTIVE ON SY6
IAT8499 NO JOBS ACTIVE ON SY7
IAT8499 NO JOBS ACTIVE ON SYLOCAL8
IAT8593 INQUIRY ON ACTIVE JOBS COMPLETE,      9 JOBS DISPLAYED

*i a,d=all
IAT8522 JOB JES3     (JOB00000) ACTIVE ON OUTSERV      TIME NAVAIL
IAT8522 JOB JES3     (JOB00000) ACTIVE ON OUTSERV      TIME NAVAIL SCHED FOR
IAT8522 JOB MONITOR   (JOB00001) ACTIVE ON MONITOR          3089.32 MIN
IAT8522 JOB INTRDR   (JOB00011) ACTIVE ON INTRDR          3088.62 MIN
IAT8522 JOB MOUNT     (JOB00003) ACTIVE ON WTR      PRT003 3089.23 MIN
IAT8522 JOB IBMUSERA (JOB00044) ACTIVE ON CIFSS1          0000.00 MIN
IAT8593 INQUIRY ON ACTIVE JOBS COMPLETE,      6 JOBS DISPLAYED
  
```

SHARE in Boston

51

Examples of *INQUIRY for active jobs and jobs being processed by DSPs.

We've seen the *I A command before for INTRDR and CI DSPs. This lists all the DSPs!

Job Execution: Putting it together



- All jobs.

```
*i j=*
IAT8674 JOB MONITOR (JOB00001) P=15 MONITOR (ACTIVE)
IAT8674 JOB SYSLOG (JOB00002) P=15 CL=W MAIN (EXECUTING-SY1)
IAT8674 JOB RACF (JOB00004) P=15 CL=W MAIN (EXECUTING-SY1)
IAT8674 JOB BPXAS (JOB00005) P=15 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB BPXAS (JOB00006) P=15 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB BPXAS (JOB00007) P=15 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB BPXAS (JOB00008) P=15 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB RUNON (JOB00009) P=15 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB INTRDR (JOB00011) P=15 INTRDR (ACTIVE)
IAT8674 JOB VTAMJ4 (JOB00012) P=15 CL=W MAIN (EXECUTING-SY1)
IAT8674 JOB TCAS (JOB00013) P=15 CL=W MAIN (EXECUTING-SY1)
IAT8674 JOB IBMUSER (JOB00014) P=15 CL=W MAIN (EXECUTING-SY1)
IAT8674 JOB SYSLOG (JOB00017) P=15 CL=W MAIN (EXECUTING-SY2)
IAT8674 JOB BPXAS (JOB00019) P=15 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB RACF (JOB00020) P=15 CL=W MAIN (EXECUTING-SY2)
IAT8674 JOB RUNON (JOB00021) P=15 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB VTAMJ4 (JOB00022) P=15 CL=W MAIN (EXECUTING-SY2)
IAT8674 JOB TCAS (JOB00024) P=15 CL=W MAIN (EXECUTING-SY2)
IAT8674 JOB JES3CI (JOB00041) P=15 CL=W MAIN (EXECUTING-SY1)
IAT8674 JOB MOUNT (JOB00003) P=11 CL=W OUTSERV (ACTIVE ON WTR)
IAT8674 JOB MOUNT (JOB00018) P=11 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB IBMUSERA (JOB00015) P=02 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB IBMUSERA (JOB00037) P=02 CL=W HOLD=(OP) MAIN
IAT8674 JOB IBMUSERA (JOB00042) P=02 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB IBMUSERA (JOB00043) P=02 CL=W OUTSERV (PENDING WTR)
IAT8674 JOB IBMUSERA (JOB00044) P=02 CL=W OUTSERV (PENDING WTR)
IAT8699 INQUIRY ON JOB STATUS COMPLETE, 26 JOBS DISPLAYED
```

SHARE in Boston

52

All jobs displayed here. Note there is no N= parameter to limit the number of lines displayed!

This Includes the current scheduler element and status.

One job held by the operator and next step is MAIN – no status so it is waiting on a queue to enter MAIN.

Job Execution: Putting it together



- Held job details.

```
*x display,j=37
IAT6306 JOB (JOB00045) IS DISPLAY CANCELED BY C3E0S11
IAT7762 - 02 JOB00037 IBMUSERA SE=(CI-COMPLETE,MAIN-NOSTAT,
IAT7762 - OUTSERV-NOSTAT,PURGE-NOSTAT),JCTMANS=(SY1,SY2,SY3,
IAT7762 - SY4,SY5,SY6,SY7,SYLOCAL8),
IAT7762 - SRVCLASS=DISCRETN,ORG=ANYLOCAL,CLASS=W,
IAT7762 - GROUP=JES3WLM,JTRACKS=00000008,IORATE=MED,
IAT7762 - PROCLIB=IATPLB11,LINEST=999,
IAT7762 - PAGEST=09999999,BYTEST=00001500,CRDEST=999,
IAT7762 - JESMSG LG=LOG,OUSID=IBMUSER,TUSID=IBMUSER,HOLD=OPR
IAT7450 JOB DISPLAY (JOB00045) PURGED
```

*X DISPLAY,J= for the held job.

Again shows it is held by the operator.

The CI element is complete and it has not yet been in MAIN – no status.

Job Execution: Putting it together



- Backlog.

```

*i b
IAT8688 FUNCTION ACTIVE WAITING
IAT8688 CI 00000000 00000001
IAT8688 INTRDR 00000001 00000000
IAT8688 MAIN 00000009 00000000
IAT8688 MONITOR 00000001 00000000
IAT8688 OUTSERV 00000012 00000000
IAT8619 INQUIRY ON BACKLOG COMPLETE

*i j=25
IAT8674 JOB IBMUSERA (JOB00025) P=02 CL=W CI
IAT8699 INQUIRY ON JOB STATUS COMPLETE, 1 JOB DISPLAYED

*x display,j=25
IAT6306 JOB (JOB00027) IS DISPLAY CALLED BY C3E0SY1
IAT7762 - 02 JOB00025 IBMUSERA SE=(CI-NOSTAT,MAIN-NOSTAT,
IAT7762 - OUTSERV-NOSTAT,PURGE-NOSTAT),JCTMANS=(SY1,SY2,SY3,
IAT7762 - SY4,SY5,SY6,SY7,SYLOCAL8),
IAT7762 - ORG=ANYLOCAL,CLASS=W,GROUP=JES3WLM,JTRACKS=00000002,
IAT7762 - IORATE=MED,PROCLIB=IATPLBI1,
IAT7762 - WAITING FOR A DSP TO BECOME AVAILABLE,LINEST=999,
IAT7762 - PAGEST=09999999,BYTEST=00001500,CRDEST=999,
IAT7762 - JESMSGLG=LOG,
IAT7762 - OUSID=IBMUSER,TUSID=IBMUSER
IAT7450 JOB DISPLAY (JOB00027) PURGED
  
```

SHARE in Boston

54

Here the CI DSP was held so that we could show a backlog example.

The *I B indicates one job is waiting for CI.

Since we knew what job was submitted, a search for which job(s) were waiting was not needed.

*I J for a specific job shows that the CI would be the next schedule element – there is no status shown here.

Again using *CALL DISPLAY for the job provides more explicit detail.

Job Execution: Putting it together



- One last one, got a couple jobs waiting in Allocate.

```

*i j=vtamj4
IAT8674 JOB VTAMJ4 (JOB00016) P=15 CL=W MAIN(ALLOCATE)
IAT8674 JOB VTAMJ4 (JOB00017) P=15 CL=W MAIN(EXECUTING-SY2)
IAT8674 JOB VTAMJ4 (JOB00021) P=15 CL=W MAIN(ALLOCATE)
IAT8699 INQUIRY ON JOB STATUS COMPLETE, 3 JOBS DISPLAYED

*i s
IAT5619 ALLOCATION QUEUE = 0000002 BREAKDOWN QUEUE = 0000000
IAT5619 SYSTEM SELECT QUEUE = 0000000 ERROR QUEUE = 0000000
IAT5619 SYSTEM VERIFY QUEUE = 0000000 FETCH QUEUE = 0000000
IAT5619 UNAVAILABLE QUEUE = 0000000 RESTART QUEUE = 0000000
IAT5619 WAIT VOLUME QUEUE = 0000000 VERIFY QUEUE = 0000000
IAT5619 ALLOCATION TYPE = AUTO
IAT5619 CURRENT SETUP DEPTH - ALL PROCESSORS = 0000000
IAT5619 MAIN NAME STATUS SDEPTH DASH TAPE
IAT5619 SY1 ONLINE IPLD 255,000 00640,00000 00144,00000
IAT5619 SY2 ONLINE IPLD 255,000 00640,00000 00144,00000
.
.
IAT5619 DEFERCT=NO
IAT5619 SDEPZERO=WAIT

*x display,j=16
IAT6306 JOB (JOB00028) IS DISPLAY , CALLED BY C3E0SY1
IAT7762 -15 JOB00016 VTAMJ4 SE=(CI-COMplete,MAIN-ACTIVE,
.
.
IAT7762 -INDEX=0D MDS ALLOC PROCESSING,LINEST=999,
IAT7762 -PAGEST=09999999,BYTEST=00001500,CRDEST=999,
IAT7762 -JESMSG LG=LOG,RQJSTAT=(
IAT7762 -SY1,FAILED PREALLOCATION SCAN),
.
.
IAT7762 -MDS FAILED TO ALLOCATE THE FOLLOWING ON SY1,
IAT7762 -1 OUT OF 1 REQUESTS FOR DEVICE=3390
IAT7450 JOB DISPLAY (JOB00028) EUROBE

```

SHARE in Boston

55

Had a problem starting VTAM during one of my IPLs of SY1. Reason does not really matter, but in my environment I had a device problem.

JOB00016 was the first attempt on SY1 and JOB00021 was a second attempt – neither worked. The provided me with the perfect opportunity to show the MDS queues with *I S and see that two jobs are on the allocation queue – waiting for a resource.

Doing a *CALL DISPLAY for one of the jobs shows that the job is active in MAIN and provided some clues that there was a problems allocating a resource.

Resolving the issue is not important here, just an example of the kind of information available using different commands.

Job Execution: Job related commands



- *F J=...
 - Can modify or cancel one or more jobs with various parameters.
 - H and R can be used to hold jobs (operator hold status) and release jobs.

```
*i j=38
IAT8674 JOB IBMUSERA (JOB00038) P=15 CL=W          MAIN(EXECUTING-SY1)
IAT8699 INQUIRY ON JOB STATUS COMPLETE,           1 JOB  DISPLAYED

*f j=38,h
IAT8080 JOB IBMUSERA (JOB00038) HELD

*i j=38
IAT8674 JOB IBMUSERA (JOB00038) P=15 CL=W          HOLD=(OP) MAIN(EXECUTING-SY1)
IAT8699 INQUIRY ON JOB STATUS COMPLETE,           1 JOB  DISPLAYED

*f j=38,r
IAT8080 JOB IBMUSERA (JOB00038) RELEASED
```

- RUN parameter specifies that a job, in a WLM managed group, waiting to be selected for execution be run now.
 - Some scheduling criteria will still apply.

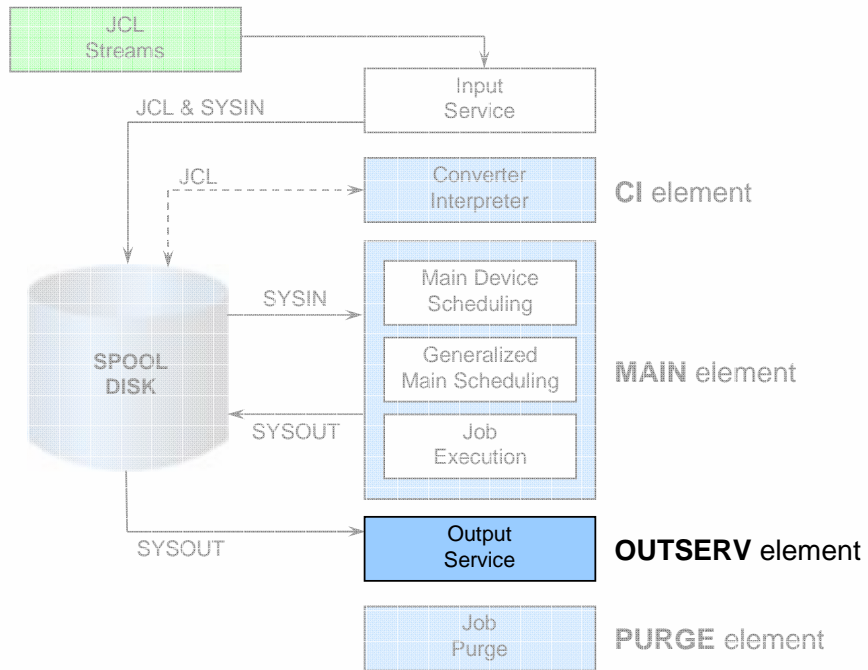
SHARE in Boston

56

*MODIFY J= to modify a job, hold a job, cancel a job, and more.

The RUN parameter specifies that a job waiting to be selected for execution be run now. This command is allowed only for jobs that are in WLM-managed groups and are waiting to be selected for execution. This will ignore some schedule criteria for the job, but not all, hence the job may still not execute immediately.

See *z/OS V1R12.0 JES3 Commands* for more details.



After execution the job moves to the OUTSERV element where output from the job is handled.



Output Service

- After execution a standard job moves to the third scheduler element – **OUTSERV**.
- Output service executes on the global and processes SYSOUT data sets.
- JES3 output service performs three distinct functions:
 - Queueing output
 - Scheduling output
 - Writing output

- *z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4.*

Output service executes on the global and processes SYSOUT data sets destined for:

- JES3 managed devices
- PSF managed devices v system application printer interface (SAPI) applications
- NJE (BSC and SNA) v Process SYSOUT (PSO)

The output service driver receives control after a job completes breakdown in main service, after a job spins off an output data set, or after JES3 spins off an output data set. JES3 output service performs three distinct functions:

- Queueing output
- Scheduling output
- Writing output

Output Service: Queueing output



- SYSOUT data sets are placed on four queues:
 - Q=WTR – *Output service writer queue*
 - SYSOUT data sets to be processed by JES3 managed devices.
 - Q=HOLD – *Output service hold queue*
 - SYSOUT data sets to be processed by system routines other than JES3 managed devices.
 - Q=BDT – *MVS/BDT work queue*
 - Network job and SYSOUT streams to be transmitted in a JES3 SNA/NJE network.
 - Q=TCP – *TCP/IP/NJE work queue*
 - Network job and SYSOUT streams to be transmitted in a JES3 TCP/IP/NJE network.

SHARE in Boston

59

- *ABCs of z/OS System Programming - Volume 13 – Chapter 11*

Q=WTR Output service writer queue: This queue contains data sets waiting for output processing by JES3 managed devices. Output service automatically selects data sets for processing based on their selection characteristics such as output class, output priority, and output device-related requirements. You can use JES3 commands to place these data sets in operator-hold status. You can also use JES3 commands to modify a data set's selection characteristics or move the data set to the HOLD queue.

Q=HOLD Output service hold queue: This queue contains data sets that are awaiting output processing by other than JES3-managed devices. These data sets must be processed by the function for which they are held (system application printer interface (SAPI) application, external writer or TSO). The function that processes the data set can then change data set characteristics, release it for JES3 processing, or cause JES3 to purge it. If necessary, the operator can force a JES3 writer to process the data set or issue a modify (*F) command to move the data set to the WTR for JES3 device processing.

Q=BDT MVS/BDT work queue: This queue contains SNA/NJE networking job or networking system output streams. MVS/BDT sends these job or system output streams to the proper node within a SNA/NJE network. You can use JES3 operator commands to hold, release, or cancel networking requests from the queue.

Q=TCP TCP/IP/NJE network jobs: This queue can be either network job streams or network SYSOUT streams. This command might have a length of 126 characters if the command is issued from an input device that permits that command length.

See also *z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4*.

Output Service: Scheduling output



- Output Scheduling Element (OSE)
 - A control block that describes the characteristics or processing options of one or more output data sets for the same job.
 - Output service scheduling uses writer scheduling parameters to search the OSEs for the first perfect-fit OSE or for the OSE which best fits the requirements of the writer needing work.
 - Find the best match of output characteristics between the SYSOUT data sets and the device.

- *ABCs of z/OS System Programming - Volume 13 – Chapter 11*

Output scheduling element (OSE)

Output service uses a basic work unit, the OSE, to represent all the output service characteristics of the output data sets of a job. Output service characteristics include format, print size, paper requirements, number of copies, and other specifications that tailor the output to the user's needs. Every job is associated with at least one OSE; an OSE represents one or more data sets with similar output requirements. After OSE construction, the OSEs are queued

Output Service: Writing output



- JES3 writer support is provided within the JES3 global address space.
 - Output data sets destined for JES3 managed devices such as print and punch.
- Output writer FSS
 - Device dependent routines that run in a separate address space and use the functional subsystem interface (FSI) to communicate with JES3 for output processing.
 - JES3 controls output scheduling for FSS-controlled devices.
- External Writer/SAPI applications
 - Routines executing in MVS jobs using the subsystem interface (SSI) to select/access/process output data sets.

SHARE in Boston

61

•z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4.

JES3 writer support consists of a writer driver, writer scheduling (selection) routines, device-dependent routines, command-processing routines (also called message-processing routines) and spool-access routines (for print and punch writers).

In most cases, the writer support is provided within the JES3 global address space. Certain devices, however, use device-dependent routines that operate in a separate address space called an output writer functional subsystem (FSS) address space. In this case, the writer driver and the command-processing routines operate in the JES3 global address space and communicate with the output writer FSS using the functional subsystem interface (FSI). The device-dependent routines, also called a functional subsystem application (FSA), and the spool-access routines operate in the output writer FSS.

External writer/SAPI application routines execute in an address space other than the JES3 address space. This type of writer is functionally independent of JES3 and operates as a completely separate MVS job. However, the external writer/SAPI application interacts with JES3, through the subsystem interface, to request data sets for processing. A subset of the output service scheduling function called PROCESS SYSOUT and system application printer interface are invoked as a result of this kind of request.

No attempt is made by output service to schedule external writers/SAPI applications as a result of constructing OSEs requiring their services; it is the responsibility of the operator to start external writers as required. JES3 posts started writers that are waiting for work when incoming jobs require processing.

Output Service: Output service definition



- OSEs are the composite of information from...
- OUTSERV initialization statement
 - Specifies default values and standards for the output service element (OSE) to be used on output devices.
- SYSOUT initialization statement
 - Defines SYSOUT class characteristics.
- `//*FORMAT JES3` control statement
 - Another JECL statement.
 - Can be either specific (DDNAME specified) or nonspecific.
- `//OUTPUT JCL` statement
 - Specifies processing options for a SYSOUT data set.

SHARE in Boston

62

- *z/OS V1R12.0 JES3 Initialization and Tuning Guide* – Chapter 4.

The merge order used to create OSEs:

1. The output service resident data area (OSD), which contains information from the OUTSERV initialization statement.
2. A non-specific FORMAT statement overrides the OUTSERV initialization statement.
3. The SYSOUT class table (SCT), which contains information from the SYSOUT initialization statement.
4. The in-storage job data set output control block (JDO), which contains information from the spooled OUTPUT JCL statement.
5. The job data set control block (JDS), which contains information from the SYSOUT DD JCL statement.
6. A specific FORMAT statement overrides all the other statements. initialization statement.

Note: If a DD has both a direct FORMAT statement and an OUTPUT statement that apply, then both are used to create separate OSEs.

Output Service: Writer definition



- **DEVICE** initialization statement
 - Specifies device characteristics and tells JES3 how to use that device.
- **FSSDEF** initialization statement
 - Used to define a WTR FSS that runs in a separate address space.
 - One statement with TYPE=WTR for each output WTR FSS to be created.
 - FSSNAME= associates the FSS with a DEVICE statement.

Initialization statements in the JES3 INISH deck associated with Output Service definition.

See *z/OS V1R12.0 JES3 Initialization and Tuning Reference* for more details.

Output Service: Output related commands



- *I U...
 - Display job output in a JES3 system.
 - Dependent upon using the Q= parameter!
 - WTR, HOLD, BDT, and TCP queues (WTR is default).
 - Additional parameters dependent upon the Q= selected.
 - *I U by itself does not give you much of anything.
 - Default is 10 detail lines, use N= to get more or ALL.
- *MODIFY U...
 - Modify job output in a JES3 system.
 - Also dependent upon using the Q= parameter!
 - Q= not specified is ALL output.
 - Additional parameters dependent upon the Q= selected.

SHARE in Boston

64

*INQUIRY,U command to display job output in a JES3 system.

*MODIFY,U command to modify job output in a JES3 system.

Why not O for output instead of U, O is for console commands.

The job output can be at various places within the system and your selection of the proper "Q=" keyword value on the *INQUIRY,U and *MODIFY,U commands dictates what output you want.

•Q=WTR is the default for *INQUIRY

•Q=ALL is the default for *MODIFY

Parameters available depend upon the Q= used. Some are common to all queues, some are common for some queues, some are unique for the queue.

See *z/OS V1R12.0 JES3 Commands* for more details.

Output Service: Putting it together



- Job in OUTSERV

```

*x display,j=29
IAT6306 JOB (JOB00030) IS DISPLAY , CALLED BY C3E0SY1
IAT7762 - 02 JOB00029 IBMUSERA SE=(CI-COMLETE,MAIN-COMLETE,
IAT7762 - OUTSERV-ACTIVE, URG-NOSTAT) ,JCTMAINS=(SY1,SY2,SY3,
IAT7762 - SY4,SY5,SY6,SY7,SYLOCAL8) ,
IAT7762 - SRVCLASS=DISCRETN,ORG=ANYLOCAL,CLASS=W,
IAT7762 - GROUP=JES3WLM,JTRACKS=00000006,IORATE=MED,
IAT7762 - PROCLIB=IATPLB11,MAINS=(
IAT7762 - SY1,SY2,SY3,SY4,SY5,SY6,SY7,SYLOCAL8) ,
IAT7762 - INDEX=WAITING FOR WRITER, INEST=999,PAGEST=09999999,
IAT7762 - BYTEST=00001500,
IAT7762 - CRDEST=999,JESMSG LG=LOG,OUSID=IBMUSER,TUSID=IBMUSER
IAT7450 JOB DISPLAY (JOB00030) PURGED

*i u,j=29
IAT8131 JOB IBMUSERA (JOB00029), T=PRT, L=22, PG=0, SR=22, BY=24504.
IAT8119 NUMBER OF JOBS FOUND : 1
  
```

SHARE in Boston

65

Using *CALL DISPLAY we see that JOB00029 is done executing, is in the OUTSERV SE, and is waiting for a writer.

Let's see what we get on Q=WTR with *I U,J=

We don't get much information other than that the job was found on Q=WTR.

Can we find out how many SYSOUT data sets there are for a job without listing everything?

I have not really found a way to do that.

Output Service: Putting it together



- Queued output details.

```
*i u, j=29, dd=?, n=3
IAT8131 JOB IBMUSERA (JOB00029), T=PRT, L=4, PG=0, SR=4, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.STEP0001.SYSPRINT(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D000000E.?.
IAT8131 JOB IBMUSERA (JOB00029), T=PRT, L=3, PG=0, SR=3, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.STEP0001.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D000000F.?.
IAT8131 JOB IBMUSERA (JOB00029), T=PRT, L=4, PG=0, SR=4, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.STEP0002.SYSPRINT(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000011.?.
IAT8119 NUMBER OF JOBS FOUND : 1

*i u, j=29, dd=?, q=hold
IAT8131 JOB IBMUSERA (JOB00029), L=7, PG=0, SR=7, BY=16336,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.JESMSG(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000002.JESMSG.
IAT8131 JOB IBMUSERA (JOB00029), L=74, PG=0, SR=74, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.JESJCL(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000003.JESJCL.
IAT8131 JOB IBMUSERA (JOB00029), L=64, PG=0, SR=64, BY=12252,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.JESYSMSG(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000004.JESYSMSG.
IAT8131 JOB IBMUSERA (JOB00029), L=3, PG=0, SR=3, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.STEP0003.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000015.?.
IAT8131 JOB IBMUSERA (JOB00029), L=3, PG=0, SR=3, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.STEP0004.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000018.?.
IAT8119 NUMBER OF JOBS FOUND : 1

*i u, j=29, dd=?, q=bdt
IAT8121 NO OUTPUT FOR SELECTED OPTIONS, OSE NOT FOUND
```

SHARE in Boston

66

Additional parameters on *I U,J=, like DD=?, will provide more details – essentially asking what are the DD names here.

Note that N=3 was used here to limit the amount of output. The default is 10.

Inquiry's done on WTR and HOLD queues find all the output for this job. Checking both the BDT and TCP queues would find no output – no output was expected on these queues for this job. But note that the reply message, IAT8121, indicates OSE NOT FOUND – this is why we mentioned what an OSE is. Sometimes the text is JOB NOT FOUND. Either way, no output matches the inquiry.

Now it was expected that output on Q=WTR should be printing. Why not?

Output Service: Putting it together



- Details for the data set and device.

```

*i u, j=29, dd=.STEP0001.SYSPRINT, req=all
IAT8131 JOB IBMUSERA (JOB00029), P=2, D=PRT002, SL=(NAVAIL), T=PRT,
IAT8131 JOB IBMUSERA (JOB00029), F=1PRT, C=6, U=PN, CH=GS10,
IAT8131 JOB IBMUSERA (JOB00029), FL=NONE, CM=(NONE,0), SS=C, CL=A,
IAT8131 JOB IBMUSERA (JOB00029), ID=IBMUSER, OB=NONE, H=N, PM=LINE,
IAT8131 JOB IBMUSERA (JOB00029), L=4, PG=0, SR=4, BY=4004,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, IP=N,
IAT8131 JOB IBMUSERA (JOB00029), DD=.STEP0001.SYSPRINT(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D000000E.?.
IAT8119 NUMBER OF JOBS FOUND : 1

*i d, d=prt002
IAT8562 PRT002 0002 (AC ) WTR F=1PRT U=PN DYN=Y
IAT8562 PRT002 0002 TIMEOUT=0
IAT8562 PRT002 0002 H=N B=N M=A
IAT8562 PRT002 0002 L=0+ DGRPY=N
IAT8562 PRT002 0002 C=6 PM=LINE
IAT8562 PRT002 0002 WS=(P,CL,PM,L,D)
IAT8562 PRT002 0002 DGRP=LOCAL
IAT8562 PRT002 0002 JOB SYSLOG (JOB00002)
IAT8572 0002 (AC ) SY1
IAT8572 0002 (AC ) SY2
IAT8572 0002 (AC ) SY3
IAT8572 0002 (AC ) SY4
IAT8572 0002 (AC ) SY5
IAT8572 0002 (AC ) SY6
IAT8572 0002 (AC ) SY7
IAT8572 0002 (AC ) SYLOCAL8
IAT8500 INQUIRY ON DEVICES COMPLETE
  
```

SHARE in Boston

67

Here detailed the information for a specific DD name is obtained. Instead of using additional parameters like DD=, the REQ=ALL parameter was used to get more details for the data set. It can be used for just about any output inquiry, but you'll get many more lines of data!

For this output we see it is going to device PRT0002 and not held. So what's up?

Used *INQUIRY D to check the device for any clues.

(AC) indicates device is Active and job SYSLOG (JOB00002) is active on the device.

I'm confused. It looks like some job may be consuming the printer?

If you could physically see the printer, likely it is not doing anything.

Output Service: Putting it together



- Displaying outstanding requests.

```

D R,KEY
SY1 IEE112I 17.10.08 PENDING REQUESTS 130
CNT KEY CNT KEY CNT KEY
 2 WTR
D R,KEY=WTR
SY1 IEE112I 17.08.27 PENDING REQUESTS 124
RM=0 IM=5 CEM=1 EM=2 RU=0 IR=0 AMRF
ID:R/K T MESSAGE TEXT
19002 I *IAT7030 JOB MOUNT (JOB00003) ON PRT003 (0003), SETUP
F=1PRT, U=PN, FCB=6.
18002 I *IAT7030 JOB SYSLOG (JOB00002) ON PRT002 (0002), SETUP
F=1PRT, U=PN, FCB=6.

*S PRT002
*IAT7032 SETPRT SUB-TASK INVOKED FOR PRT002 (0002)
*&SY1 R= JES3 13 IEC120A M 0002,PN
-R 13,PN
&SY1 R= IEE600I REPLY TO 13 IS;PN
IAT7007 JOB SYSLOG (JOB00002) ON WRITER PRT002 (0002), DSN=
IAT7007 +MASTER+.SYSLOG.JOB00002.D0000009.?, PURGED.
IAT7007 JOB SYSLOG (JOB00002) ON WRITER PRT002 (0002), DSN=
IAT7007 +MASTER+.SYSLOG.JOB00002.D000000A.?, PURGED.
IAT7007 JOB SYSLOG (JOB00002) ON WRITER PRT002 (0002), DSN=
IAT7007 +MASTER+.SYSLOG.JOB00002.D000000B.?, PURGED.
IAT7001 JOB TCAS (JOB00013) IS ON WRITER PRT002(0002),RECORDS=14
IAT7007 JOB TCAS (JOB00013) ON WRITER PRT002 (0002), DSN=
IAT7007 IBMUSER.TCAS.JOB00013.D0000009.?, PURGED.
. . .

```

SHARE in Boston

68

So while trying to figure this out, someone mentioned an MVS command that might be useful. This provides a lists of outstanding messages.

```
D R,KEY=WTR
```

KEY=keyname

The system requests those messages that are identified by a one to eight-character keyname, such as those messages issued by the specified dynamic support program (DSP) of JES3.

Using just KEY summarizes outstanding requests for all JES3 DSPs.

Specifying a DSP like WTR gives a list of messages outstanding for that message.

IAT7030 indicates device needs to be started. These messages were also at the top of the operator screen – but I paid no attention to them as I rarely print output.

So the printer had to be started with *S PRT002 and output flies!

The point of this is not that messages on the operator console (a given) but that sometimes it takes using multiple JES3 and MVS commands to determine the overall picture of what is going on.

Output Service: Q=HOLD vs held output



- Q=HOLD is for SYSOUT data sets to be processed by system routines other than JES3 managed devices.
- When does JES3 output end up on Q=HOLD?
 - Output using a SYSOUT class with HOLD=TSO.
 - TSO user can access output with TSO/E OUTPUT command.
 - Output using a SYSOUT class with HOLD=RSVD and // JOB statement where MSGCLASS specifies a SYSOUT class with HOLD=RSVD.
 - TSO user can access output with TSO/E OUTPUT command.
 - SYSOUT classes with HOLD=EXTWTR
 - Held for an external writer.
 - Explicitly moved to Q=HOLD
 - *MODIFY with NQ=HOLD

SHARE in Boston

69

Output on Q=HOLD is output that is selectable and held for processing by a system function (TSO user, external writer). Not for processing by a JES3 managed device (WTR), BDT, or TCP.

•z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4.

Output service hold queue (Q=HOLD): This queue contains data sets that are awaiting output processing by other than JES3-managed devices. These data sets must be processed by the function for which they are held (system application printer interface (SAPI) application, external writer or TSO). The function that processes the data set can then change data set characteristics, release it for JES3 processing, or cause JES3 to purge it. If necessary, the operator can force a JES3 writer to process the data set or issue a modify (*F) command to move the data set to the WTR for JES3 device processing.

Output Service: Q=HOLD vs held output



- Held output is output in a hold state and is not selectable for processing.
 - Can be on any of the four output queues.
- When does JES3 output end up in a hold state?
 - JCL DD statement with HOLD=YES
 - H=(USER) state
 - Different from JES2 which treats like OUTDISP=HOLD and puts the output on the JES2 hold queue.
 - Explicitly held by operator
 - H=(OPER) state
 - *MODIFY with NH=HOLD
 - JES3 subsystem placed the data set in hold
 - H=(SYS)

SHARE in Boston

70

Held output is output on any Q= that is in a hold state – when in a hold state, the output is not selectable for processing/writing.

Hold states are USER, OPER, SYS.

No specific references for this slide.

Output Service: Putting it together



- Output using JCL DD statement with HOLD=YES.

```
*i u,j=29,dd=?,req=all
IAT8131 JOB IBMUSERA (JOB00029), P=2, D=PRT002, SL=(NAVAIL), T=PRT,
IAT8131 JOB IBMUSERA (JOB00029), F=1PRT, C=6, U=PN, CH=GS10,
IAT8131 JOB IBMUSERA (JOB00029), FL=NONE, CM=(NONE,0), SS=C, CL=A,
IAT8131 JOB IBMUSERA (JOB00029), ID=IBMUSER, OB=NONE, H=(USER),
IAT8131 JOB IBMUSERA (JOB00029), PM=LINE, L=3, PG=0, SR=3, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, IP=N,
IAT8131 JOB IBMUSERA (JOB00029), DD=.STEP0002.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000012.?.
IAT8119 NUMBER OF JOBS FOUND : 1

*F,U,j=29,dd=.STEP0002.SYSUT2,NH=N
IAT8122 CHANGED JOB IBMUSERA (JOB00029) DATA SET=.STEP0002.SYSUT2
IAT8118 NUMBER OF DATA SETS MODIFIED : 1
IAT7001 JOB IBMUSERA (JOB00029) IS ON WRITER PRT002(0002),RECORDS=3

*i u,j=28,dd=.STEP0001.SYSUT2,req=all
IAT8131 JOB IBMUSERA (JOB00028), P=2, D=PRT002, SL=(NAVAIL), T=PRT,
IAT8131 JOB IBMUSERA (JOB00028), F=1PRT, C=6, U=PN, CH=GS10,
IAT8131 JOB IBMUSERA (JOB00028), FL=NONE, CM=(NONE,0), SS=C, CL=A,
IAT8131 JOB IBMUSERA (JOB00028), ID=IBMUSER, OB=NONE,
IAT8131 JOB IBMUSERA (JOB00028), H=(OPER,USER), PM=LINE, L=3, PG=0,
IAT8131 JOB IBMUSERA (JOB00028), SR=3, BY=4084, COPIES=1, IP=N,
IAT8131 JOB IBMUSERA (JOB00028), DD=.STEP0001.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00028), DSN=IBMUSER.IBMUSERA.JOB00028.D000000F.?.
IAT8119 NUMBER OF JOBS FOUND : 1
```

SHARE in Boston

71

Still have one data set for job 29 on Q=WTR.

JCL had a DD statement with HOLD=YES which results in a USER HOLD of the output.

If the user wants the output printed, the modify command is used to release the hold.

When an operator uses the modify command to hold output, H=(OPER) would be seen.

You may actually see both: H=(OPER,USER) in which case doing a release with NH=N releases both!

Output Service: Putting it together



- Output on Q=HOLD.

```
*i u, j=29, dd=? q=hold
IAT8131 JOB IBMUSERA (JOB00029), L=7, PG=0, SR=7, BY=16336,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=..JESMSGLG(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000002.JESMSGLG.
IAT8131 JOB IBMUSERA (JOB00029), L=74, PG=0, SR=74, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=..JESJCL(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000003.JESJCL.
IAT8131 JOB IBMUSERA (JOB00029), L=64, PG=0, SR=64, BY=12252,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=..JESYSMSG(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000004.JESYSMSG.
IAT8131 JOB IBMUSERA (JOB00029), L=3, PG=0, SR=3, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.STEP0003.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000015.?.
IAT8131 JOB IBMUSERA (JOB00029), L=3, PG=0, SR=3, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, DD=.STEP0004.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000018.?.
IAT8119 NUMBER OF JOBS FOUND : 1
```

Let's turn our attention back to the output on Q=HOLD for JOB00029.

Output Service: Putting it together



- Detail for output on Q=HOLD.

```
*i u,j=29,dd=.STEP0003.SYSUT2,q=hold,req=all
IAT8131 JOB IBMUSERA (JOB00029), P=2, D=ANYLOCAL, DSID=*, SL=(NAVAI),
IAT8131 JOB IBMUSERA (JOB00029), W=*, F=1PRT, C=6, U=PN, CH=PS10,
IAT8131 JOB IBMUSERA (JOB00029), FL=NONE, CM=(NONE,0), SS=C, CL=T,
IAT8131 JOB IBMUSERA (JOB00029), ID=IBMUSER, OB=NONE, H=N, PM=LINE,
IAT8131 JOB IBMUSERA (JOB00029), L=3, PG=0, SR=3, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, IP=N,
IAT8131 JOB IBMUSERA (JOB00029), DD=.STEP0003.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000015.?.
IAT8119 NUMBER OF JOBS FOUND : 1

*i u,j=29,dd=.STEP0004.SYSUT2,q=hold,req=all
IAT8131 JOB IBMUSERA (JOB00029), P=2, D=LOCAL, DSID=*, SL=(NAVAI),
IAT8131 JOB IBMUSERA (JOB00029), W=*, F=1PRT, C=6, U=PN, CH=PS10,
IAT8131 JOB IBMUSERA (JOB00029), FL=NONE, CM=(NONE,0), SS=C, CL=W,
IAT8131 JOB IBMUSERA (JOB00029), ID=IBMUSER, OB=NONE, H=N, PM=LINE,
IAT8131 JOB IBMUSERA (JOB00029), L=3, PG=0, SR=3, BY=4084,
IAT8131 JOB IBMUSERA (JOB00029), COPIES=1, IP=N,
IAT8131 JOB IBMUSERA (JOB00029), DD=.STEP0004.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00029), DSN=IBMUSER.IBMUSERA.JOB00029.D0000018.?.
IAT8119 NUMBER OF JOBS FOUND : 1

*i u,j=32,q=hold,dd=.STEP0003.SYSUT2,cl=?,h=?
IAT8131 JOB IBMUSERA (JOB00032), CL=T, H=(OPER), L=3, PG=0, SR=3,
IAT8131 JOB IBMUSERA (JOB00032), BY=4084, COPIES=1,
IAT8131 JOB IBMUSERA (JOB00032), DD=.STEP0003.SYSUT2(1),
IAT8131 JOB IBMUSERA (JOB00032), DSN=IBMUSER.IBMUSERA.JOB00032.D0000015.?.
IAT8119 NUMBER OF JOBS FOUND : 1
```

SHARE in Boston

73

Using REQ=ALL to get more details there is no real indication of why the output is on Q=HOLD.

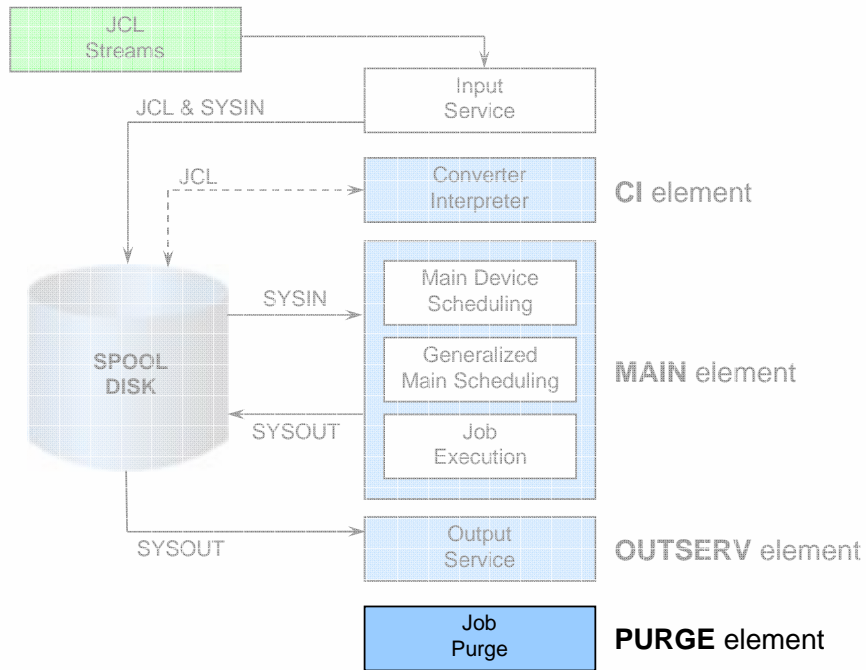
We only know from the initialization statements that:

- Class T is defined by SYSOUT,CLASS=T,HOLD=TSO
- Class W is defined by SYSOUT,CLASS=W,HOLD=EXTWTR,DEST=LOCAL

So again, sometimes it takes cross referencing information from multiple sources to come up with a complete picture.

Note also H=N in both cases but the output could also have been held as seen here for JOB00032 where an operator hold exists.

For JOB00032, TSO OUTPUT does display the output for DD=.STEP0003.SYSUT2(1), but afterward the data set remains on Q=HOLD with H=(OPER).



Finally the job is done – executed and all output processed – so it's time to remove it from the system.



Job Purge

- The last processing step for all jobs is the final scheduler element – **PURGE**.
- Purge releases the JES3 SPOOL and resident storage used for the job.
- JES3 writes SMF record type 25 which contains device allocation information.
 - One record for each job that completed C/I processing.
 - One record for each job that uses a private catalog.
 - One record for each MDS dynamic unallocation request.
- JES3 writes SMF record type 26 which contains final job accounting information.
- An operator message is issued for the purged job.

THE END OF THE JOB!

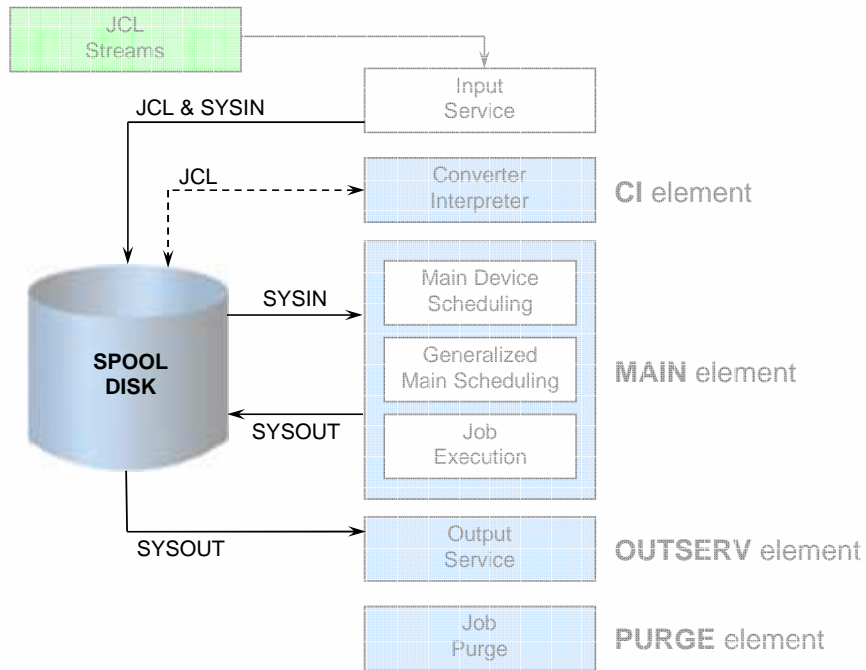
SHARE in Boston

75

•z/OS V1R12.0 JES3 Initialization and Tuning Guide – Chapter 4.

Purge is the last processing function for a job in the JES3 system. It releases all JES3 DASD space assigned to the job, making it available for allocation to subsequent jobs. A message is issued to the operator indicating that the job has been purged from the system.

At the conclusion of purge, the JES3 installation accounting routine writes out a type 26 SMF record. This record contains times and dates of the processing on the processor writing the record. Since the clocks on the various processors are not synchronized, elapsed times should all be taken from the same types of records (for example, from type 26 records). Otherwise, timing differences in the records may result.



We referred to SPOOL a few times. We'll cover a few highlights as this could be yet another whole session!



SPOOL

- SPOOL is the buffer between routines and devices for input and output.
- SPOOL is also the storage space for JES3 control blocks and data.
- SPOOL consists of one or more single extent DASD data sets.
- Can logically group data sets into SPOOL partitions.
- Track groups are the basic unit of allocation.
 - Track group is a group of SPOOL records.
 - A spool record is the size of one JES3 buffer.
- Single Track Table (STT) contains JES3 control block.

SHARE in Boston

77

- *ABCs of z/OS System Programming - Volume 13 – Chapter 4*

The spool data management function controls the JES3 spool and data. The term "spool" used in the JES3 environment is an acronym for simultaneous-peripheral-operations-online and refers to that set of direct access storage devices (DASD) especially set aside for JES3. Spool data management routines control the allocation, access, and deallocation of space on these designated direct access storage devices (DASD).

The single track table (STT) is a section of spool space used exclusively for JES3 control blocks not associated with a particular job, such as control blocks used to track JES3 functions and to save status. The allocation mechanism of the single track table is by record as opposed to track group, in contrast to the rest of JES3 spool space allocation.

You need to be aware of where STT is allocated when changing SPOOL devices.

SPOOL: Definition



- Data sets defined to JES3 via...
 - DD statement in JES3 start procedure
 - DYNALLOC initialization statement
 - Easier method for changing your spool configuration
- SPOOL partitions defined via...
 - SPART initialization statement
 - SPART parameter on TRACK and FORMAT initialization statements assigns data sets to a partition.
- SPOOL allocation defined via...
 - BUFFER initialization statement defines buffer size (BUFFSIZE) and default track group size (GRPSZ).
 - SPART defines track group size for the partition.

SHARE in Boston

78

Initialization statements in the JES3 INISH deck associated with SPOOL definition.

Any of the initialization statements preceding the ENDJSAM statement which indicates the end of the JES3 spool initialization statements.

See *z/OS V1R12.0 JES3 Initialization and Tuning Reference* for more details.

SPOOL: Related commands



- *I Q,... for various SPOOL inquiries
 - S for overall summary
 - DD= for SPOOL data set information
 - SP= for SPOOL partition information
- *F Q,... to modify SPOOL data sets and partitions
 - Reassign data sets to partitions.
 - Reassign partition overflows.
 - Can suspend and resume allocations from SPOOL data sets.
 - Various means with different affects on jobs with data in the affected data set.
 - Can also hold and release jobs related to SPOOL data sets.

SHARE in Boston

79

*INQUIRY,Q for various spool inquiries.

*MODIFY,Q to modify SPOOL data sets and partitions.

See *z/OS V1R12.0 JES3 Commands* for more details.

SPOOL: Putting it together



- SPOOL status

```

*i q,s
IAT8530 1,005 GRPS, 870 LEFT ( 87%); 0 UNAVAIL, 0 DRAINED

*i q,dd=all
IAT8513 SPOOL1 PART1 525 GRPS, 390 LEFT ( 74%) STT
IAT8513 SPOOL2 PART2 120 GRPS, 120 LEFT (100%)
IAT8513 SPOOL3 PART3 120 GRPS, 120 LEFT (100%)
IAT8513 SPOOL4 PART4 120 GRPS, 120 LEFT (100%)
IAT8513 SPOOL5 PART5 120 GRPS, 120 LEFT (100%)
IAT8611 INQUIRY ON SPOOL DATA SET STATUS COMPLETE

*i q,sp=all
IAT8980 DRAINED HAS NO SPOOL DATA SETS
IAT8980 UNAVAIL HAS NO SPOOL DATA SETS
IAT8509 PART1 : 525 GRPS, 390 LEFT ( 74%); MIN 3%, MRG 5%, DEF, INIT, OVI
IAT8509 PART2 : 120 GRPS, 120 LEFT (100%); MIN 3%, MRG 5%, OVFL
IAT8509 PART3 : 120 GRPS, 120 LEFT (100%); MIN 3%, MRG 5%, OVFL, OVIN
IAT8509 PART4 : 120 GRPS, 120 LEFT (100%); MIN 3%, MRG 5%, OVFL, OVIN
IAT8509 PART5 : 120 GRPS, 120 LEFT (100%); MIN 3%, MRG 5%, OVFL, OVIN
IAT8980 PART6 HAS NO SPOOL DATA SETS
IAT8607 INQUIRY ON SPOOL PARTITION STATUS COMPLETE

*I Q DD=ALL
IAT8513 SPLX301 PART1 12,375 GRPS, 12,102 LEFT ( 98%), STT
IAT8513 SPLX302 PART1 12,375 GRPS, 12,106 LEFT ( 98%), STT
IAT8513 SPLX321 PART2 12,375 GRPS, 12,310 LEFT ( 98%)
IAT8513 SPLX322 PART3 12,375 GRPS, 1,856 LEFT ( 15%)
IAT8513 SPLX323 PART1 33,750 GRPS, 33,477 LEFT ( 99%), STT
IAT8611 INQUIRY ON SPOOL DATA SET STATUS COMPLETE

```

SHARE in Boston

80

*I Q,S was is used to inquire about overall SPOOL usage.

I* Q,DD=ALL was used to obtain detail about SPOOL data sets.

Note that SPOOL1 contains STT records. Need to be careful when deleting or moving SPOOL data sets containing STT records.

SPOOL data sets can be grouped into SPOOL partitions. *I Q,SP=ALL provides details for the partitions. It does not identify which data sets are in a partition, that can be seen with the previous inquiry where the partition name is listed for each data set.

Final example is an inquiry about SPOOL data sets that was done on a different system (i.e. no relation to the previous inquiries). This shows multiple data sets in a partition and one partition running low on available space.

SPOOL: Top 10 list



- The “PIG” command
 - Shows the jobs which are the 10 largest users of SPOOL.
 - To free up the space, cancel or dump job the offenders.

```
*I Q SP=ALL,U
IAT8583 JES3PART TOTAL IN USE BY JES3 28 TRKGPS
IAT8583 JES3PART TOTAL IN USE BY JOBS 50,396 TRKGPS
IAT8587 JES3PART USERS FOUND= 5 JES3 1 247 JOBS: 10 DISPLAYED
IAT8527 JES3PART: JOB EATSPOL2 (J0394781) 28,948 TRKGPS, 50%
IAT8527 JES3PART: JOB CLSR01D (J0368596) 9,543 TRKGPS, 10%
IAT8527 JES3PART: JOB DIOTST1 (J0338386) 5,426 TRKGPS, 9%
IAT8527 JES3PART: JOB T0413780 (JOB42514) 1,273 TRKGPS, 2%
IAT8527 JES3PART: JOB CLSR01D (J0293978) 791 TRKGPS, 1%
IAT8527 JES3PART: JOB RMF (JOB39973) 466 TRKGPS, 1%
IAT8527 JES3PART: JOB RMF (J0166009) 395 TRKGPS, 1%
IAT8527 JES3PART: JOB CLSR01B (J0368224) 295 TRKGPS, 1%
IAT8527 JES3PART: JOB CLAGNTD (JOB39999) 237 TRKGPS, <1%
IAT8527 JES3PART: JOB CLSR01DS (J0379076) 132 TRKGPS, <1%
IAT8591 INQUIRY ON SPOOL SPACE USAGE COMPLETE
```

SHARE in Boston

81

Finally, if a SPOOL is running low on available space, one could find the top ten largest uses of SPOOL. Ten are shown as the default for *I Q,SP=ALL,U is N=10. You could show more or less, but that does not affect the operation of the command – just the number of lines output.

This is often referred to as the “PIG” command because it shows the pigs or because it is a pig to run? You decide:

Note in JES3 Commands publication:

This might be a long-running command. This command can also prevent other JES3 commands from being processed if job priorities or classes are specified on another command.

See *z/OS V1R12.0 JES3 Commands* for more details.

References



- z/OS V1R12.0 publications are now available at <http://www-03.ibm.com/systems/z/os/zos/bkserv/r12pdf/>
- *z/OS V1R12.0 JES3 Initialization and Tuning Guide* (SA22-7549-09)
 - **Chapter 4 on “JES3 Job Management”**
- *z/OS V1R12.0 JES3 Commands* (SA22-7540-10)
 - Chapters 4 through 7 on “Controlling...” along with command descriptions.
- *z/OS V1R12.0 JES3 Initialization and Tuning Reference* (SA22-7550-09)
 - JES3 initialization statements
- *ABCs of z/OS System Programming - Volume 13* (SG24-7717)
 - <http://www.redbooks.ibm.com/abstracts/sg247717.html>
 - Details on JES3 externals and internals (control blocks), initialization statements, and operator commands.

SHARE in Boston

82

Here are a number of references that were included in the presentation.

Chapter 4 of the *z/OS V1R12.0 JES3 Initialization and Tuning Guide* is a very good starting point (referred to often in the notes and while preparing the slides).

Chapter 4 through 7 of the *z/OS V1R12.0 JES3 Commands* all have descriptions and overviews that are very useful. And of course the publications documents all the JES3 commands!

z/OS V1R12.0 JES3 Initialization and Tuning Reference describes the various JES3 initialization statements.

The *ABCs of z/OS System Programming - Volume 13* is a Redbook that not only describes various portions of JES3 and JES3 services, but also goes into more details about the internals of JES3. Don't be overwhelmed by it as referring to it for specific topics provides addition insight and details not available in the standard publications.



JES3 Education

- Use SHARE!
 - Past proceedings.
 - JES3 Basics: Introduction to JES3 Parts 1 & 2
Denver 2009 sessions 2351 & 2352 – Greg Thompson
 - Let us know what you want to know!

SHARE in Boston

83

Denver 2009 session 2351 and 2352 provide a very good introduction into JES3. It provides a bigger overview of JES3, including the workflow, where as this presentation focused on providing more details about the job workflow.



JES3 Education

- IBM Training: ITSO zSeries JES3 Workshop
 - **5 day class starting 9/13/2010 – Chicago, IL**
 - Search from IBM home page or the Education catalog for 'ITS80'
 - http://www-03.ibm.com/services/learning/ites.wss/us/en?pageType=course_description&courseCode=ITS80
- IBM Education Assistant JES3 modules
 - [z/OS V1.8 JES3 Networking TCP/IP NJE](#)
 - [z/OS V1R10 JES3 ease of use, application enablement, spool browse](#)

SHARE in Boston

84

http://www-03.ibm.com/services/learning/ites.wss/us/en?pageType=course_description&courseCode=ITS80

ITS80 description:

This course covers the design, installation, and management of a JES3 system, both externally and internally. All features and functions of JES3 will be discussed. Coverage includes the functions and features of all releases currently available, including the current release z/OS JES3 V1.9.

Wrap up!



Questions?

Discussion?

Thank You!

SHARE in Boston

85