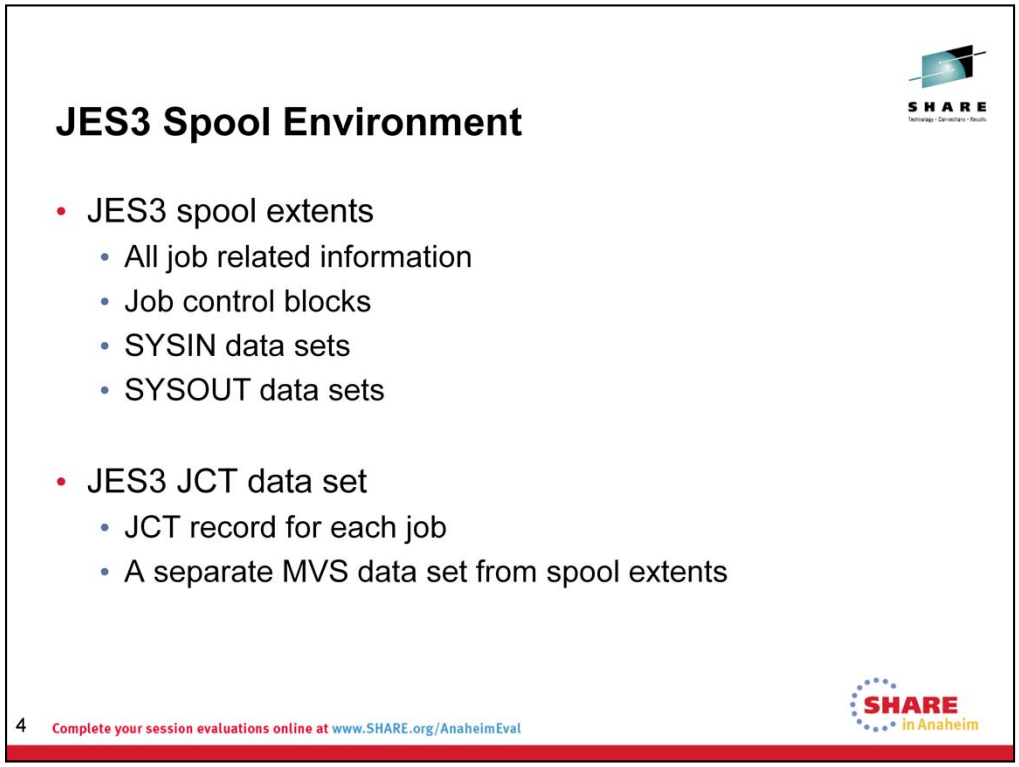

#SHAREorg
SHARE
JES3 Spool Management
Greg Thompson
IBM System Test
March 13, 2014
Session Number 15091
JES3
SHARE in Anaheim
Copyright (c) 2014 by SHARE Inc.
Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-nc-sa/3.0/

Trademarks

- **The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.**
- **IBM®**
- **MVS**
- **JES2**
- **JES3**
- **RACF®**
- **z/OS®**
- **zSeries®**
- **The following are trademarks or registered trademarks of other companies.**
- * Registered trademarks of IBM Corporation
- * All other products may be trademarks or registered trademarks of their respective 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.
- **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.

Objective

- Session abstract:
  - Ever wonder how JES3 spool is used and how it can be managed? In this session, a speaker from the JES3 team will present how JES3 utilizes the spool volumes and the controls available. Also covered will be the new features added in z/OS V2.1 and V1.13 that make it easier for you to manage your JES3 spool volumes.



Spool extents

JES3 allocates space on spool to a job upon entry to the system and deallocates spool space when the job leaves the system. An initial quantity of spool space is allocated to hold job-related control blocks and a job's SYSIN and SYSOUT data. Additional spool space may be required during the execution of a job to hold SYSOUT data.

JES3 records several types of data on the spool data sets:

Information (originally taken from initialization statements) necessary to initialize JES3 in the global and local processors.

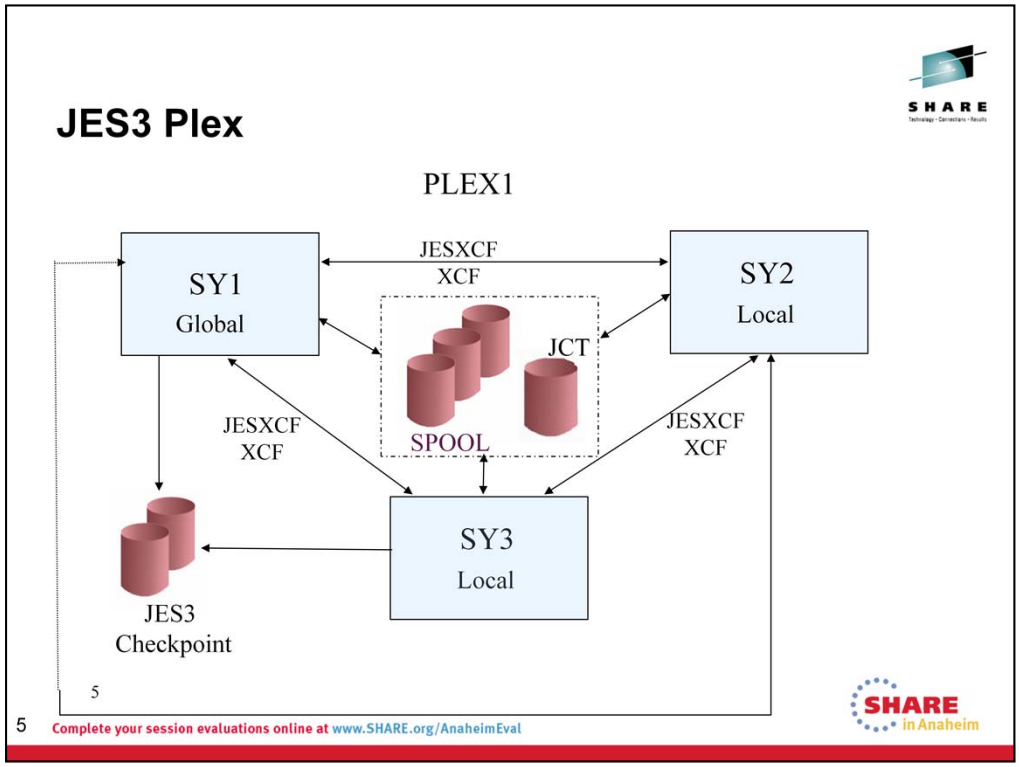
JES3 control blocks that define the scheduling and operational characteristics of the jobs.

SYSIN (DD * or DD DATA) data sets and SYSOUT data sets for jobs.

JCT data set

The job control table (JCT) is a major JES3 control block on a DASD device. The data set is defined in the JES3 start procedure. The DDNAME is JES3JCT. The JES3 JCT data set does not reside within the DASD space allocated for the spool extents and is allocated by MVS during JES3 start.

The JCT data set contains a JCT entry or every job in the JES3 complex. Many JES3 functions, such as inquiry and modify, access the JCT. But the JES3 function that uses the JCT the most is the job segment scheduler (JSS).



JES3 Plex

A sysplex is a set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. A sysplex can have up to 32 MVS systems. The MVS cross system coupling facility (XCF) provides authorized application programs intra- and inter-processor communication (send and receive data) services in the sysplex. XCF services are available both in single and multisystem environments. Each sysplex is given an XCF sysplex name (in the visual PLEX1) and each system in the sysplex has a unique name as well. A JES3 sysplex must be fully contained within a sysplex and may or may not be the primary subsystem for all the MVS systems in the sysplex. Note that JES3 cannot be a secondary subsystem unlike JES2.

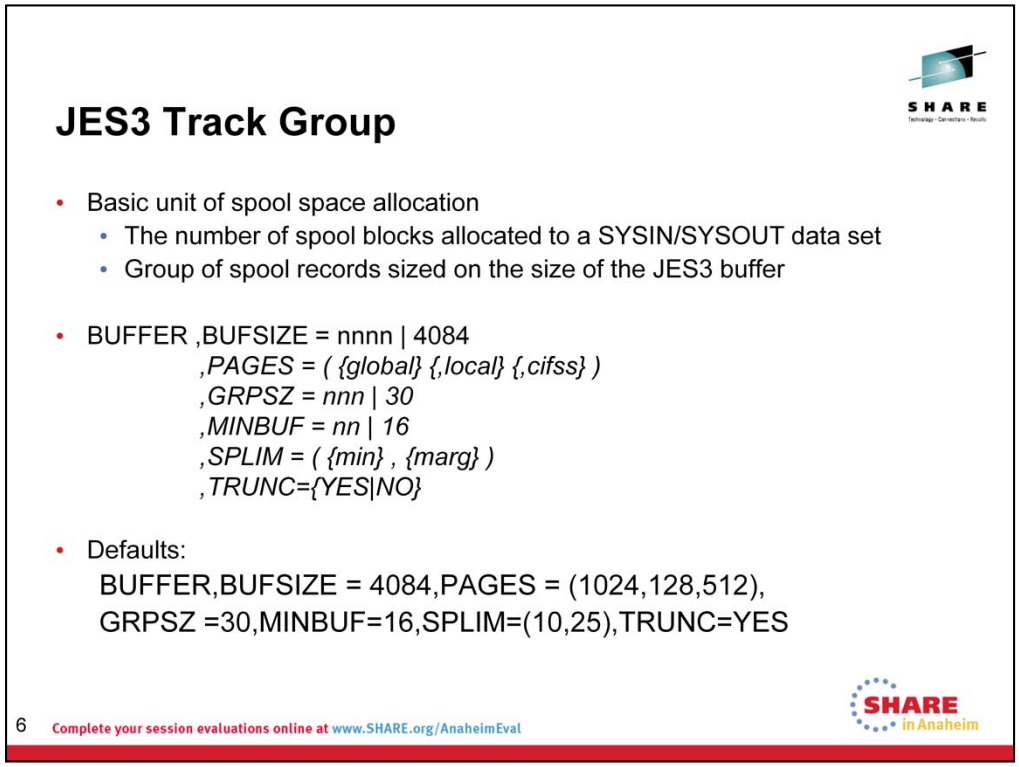
XCF and JESXCF

In a JES3 complex one of the JES3 processors is designated as the focal point for the entry and distribution of jobs and for the control of resources needed by the jobs. That processor, called the *global* processor, distributes work also to all other processors, called *local* processors. The global processor and the local processors communicate using the MVS cross-system coupling facility (XCF) through the JES common coupling services (JESXCF) component. Among other things, the JES3 inter-processor communication is used to extend the intra-processor SSI communication from local processors to the global which provides the services for most of the SSI calls.

JES2, on the other hand, has no global concept and each member of the MAS takes turns accessing the spool and uses the checkpoint to control the access of the spool. Therefore access of the checkpoint is much more active in JES2 and it is why JES2 can use the coupling facility for its checkpoint.

The JES3 global processor manages jobs and resources for the entire complex and selects jobs with readily available resources for MVS execution. The resources that JES3 manages include processors, I/O devices, spool volumes, the job queue, the

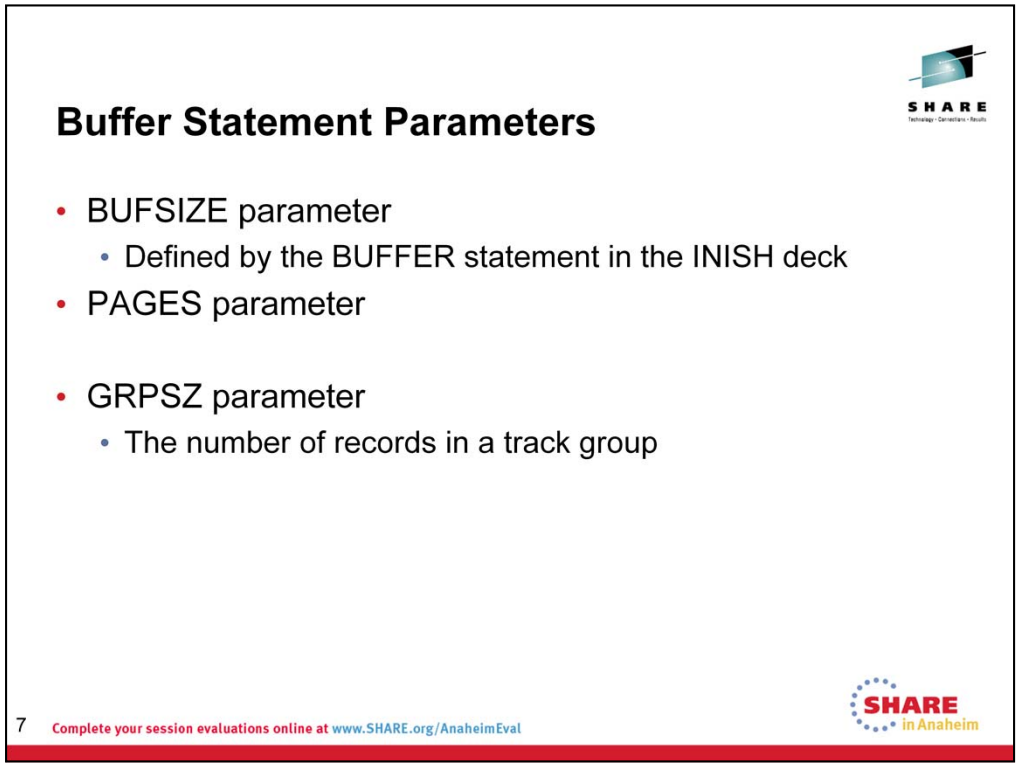
checkpoint data set, and sysin/sysout data. To avoid job execution delays that result when resources are not available, JES3 ensures that they are available before selecting the job for processing.



A track group is the number of records that JES3 treats as a unit when allocating spool space. You can specify the size of a track group using the GRPSZ parameter, choosing a value from 1 to 999. The GRPSZ parameter can be used on two initialization statements:

•The BUFFER statement, if you want to define a “default” track group size for spool partitions that do not have an explicit track group size specification, or if you do not define any spool partitions

•Use the SPART statement which defines spool partitioning, if you want to override the BUFFER statement and tailor the track group size to the type of data in the spool partition being defined. More on spool partitioning later.



JES3 allocates one or more track groups to a job when the job needs spool space. The number of track groups allocated on each request depends on the number defined using the TRKGRPS parameter for the job's SYSOUT class, //*MAIN JES3 control statement, job class, or assigned processor.

BUFSIZE

The size of records on the spool data sets (except the JCT) is defined by the BUFSIZE parameter on the BUFFER initialization statement, and it may range from 1952 to 4084. The recommended buffer size is 4084 bytes. A cold start is required to change this parameter.

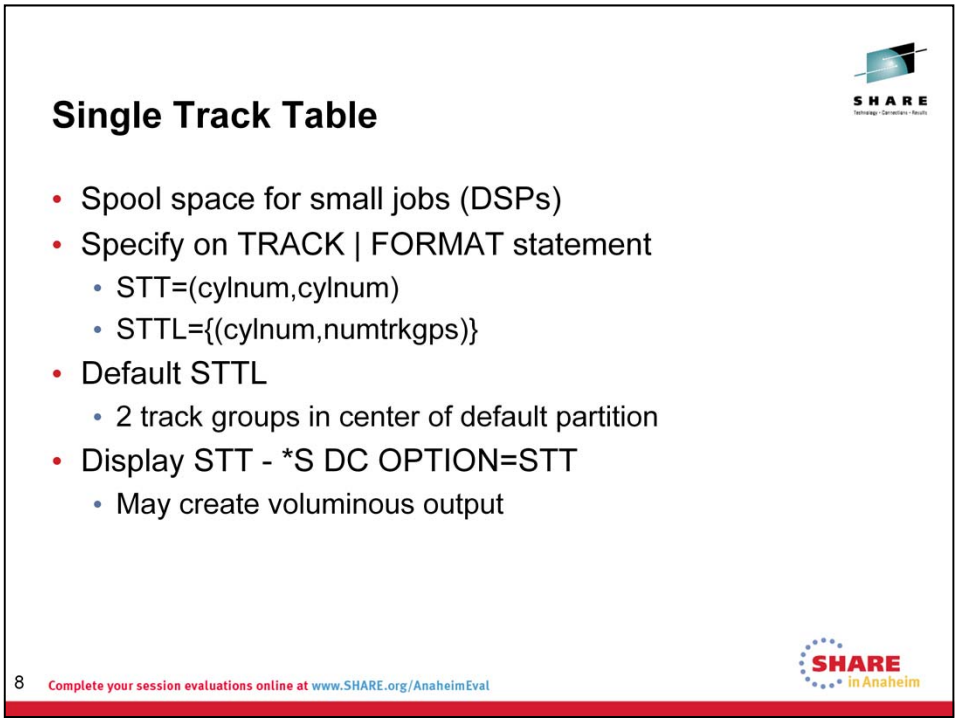
PAGES

The amount of virtual storage allocated to buffers is specified by the PAGES parameter, and acquired dynamically (AGETMAIN) at JES3 initialization. If the number of available buffers goes below a value calculated from the MINBUF parameter, JES3 identifies the condition by adding a user-specified identifier to all console messages. JES3 uses buffers to read and write data from and to spool. A buffer is an area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written.

GRPSZ

The GRPSZ parameter on the BUFFER and SPART initialization statements specifies the number of spool records in each track group. (The BUFSIZE parameter determines the size of each spool record.) The number must not be greater than 999. JES3 rounds the specified value up to the number of records in the nearest whole physical track for the selected spool device type. Track group specification is important for spool space utilization. Each job entering the system is initially assigned a track group and during its life in the system is assigned

additional track groups. Do not over specify this parameter. The default value is okay.



Single track table

The single track table 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.

There are two types of STTs used in the JES3 environment:

• The job control table (JCT) STT

• The system STT

Job control table single track table (JCT STT)

The job control table single track table (JCT STT) is built during JES3 initialization. The JCT STT is a fixed length table and contains a bit map used to allocate JCTs.

System single track table (STT)

The system STT is used to allocate single-record files (SRFs) on a individual record basis. The system programmer specifies the spool data sets that contain the STTs to be allocated using the STT= or STTL= parameter on the TRACK or FORMAT statement. The first system STT is called the primary STT while STTs that are built later are called expansion STTs. The primary STT and any expansion STTs are linked to form a single linked list.

FORMAT,DDNAME=SPOOL1,**STT**=(400,401)

FORMAT,DDNAME=SPOOL2,**STTL**=(400,2)

STT Specifies the range of cylinders you want allocated to the single track table (STT). This range must be within the extent allocated to the data set. The value of *cylnum* specifies an absolute cylinder number. (Absolute cylinder numbers are device-dependent; the component description for the device describes the numbering scheme.)

STTL Specifies the location and number of track groups to allocate to the single track table (STT). These track groups must be within the extent allocated to the data set. The value for *cylnum* specifies an absolute cylinder number indicating the beginning cylinder number of the STT allocation in this extent. (Absolute cylinder numbers are device-dependent; the component description for the device describes the numbering scheme.) The value for *numtrkgps* specifies the number of track groups to allocate to this extent, beginning with the first track group that is located completely in cylinder *cylnum*. The maximum number of track groups that may be allocated to the STT is 9999.

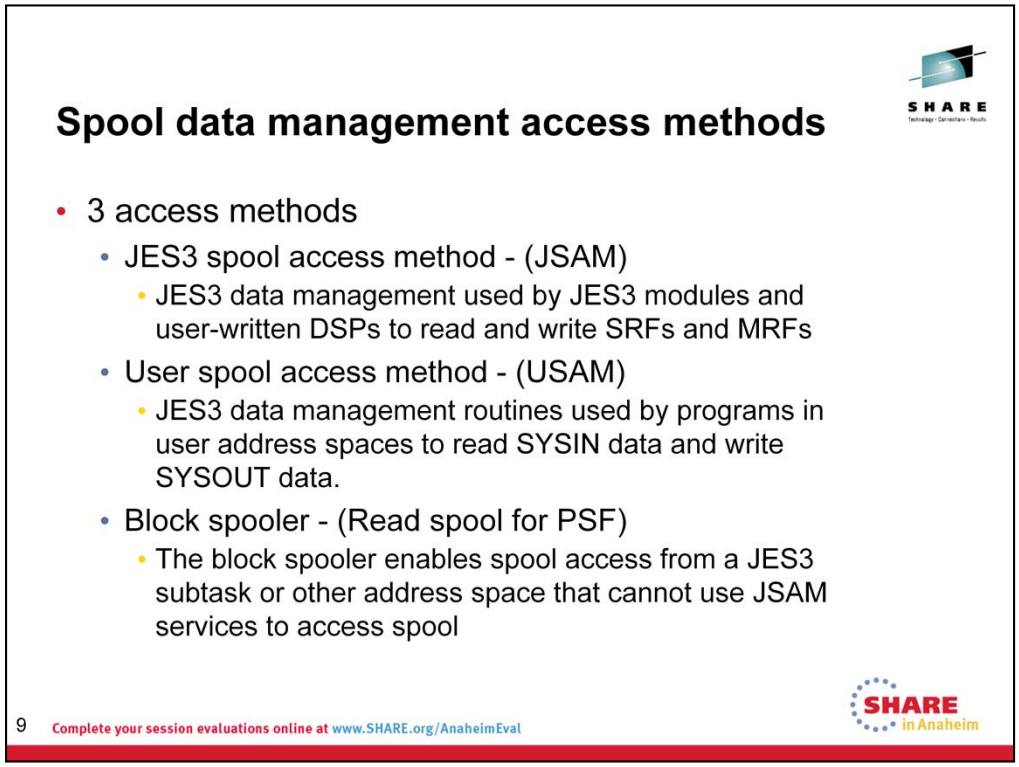
STT dynamic expansion

If all STT space is used, the STT is dynamically expanded. STT space is accounted for on a record basis via a control block called the STT. In the STT control block, there is one bit for each STT record.

Replacing or deleting a spool data set

If you replace a spool data set, JES3 cancels all jobs with data on the replaced spool data set. If the replaced data set contains STT records, JES3 might lose information which could result in the loss of jobs in the system. STT records include information such as the status of devices, DJC network data, deadline queue data, volume unavailable data, dynamic allocation checkpoint data, output service checkpoint data, JESNEWS, device fencing data, virtual unit status, GMS status, and FSS checkpoint data. If STT data is lost, JES3 issues messages that allow you to take the appropriate recovery actions.

If you delete a spool data set, JES3 cancels all jobs in the system that have spool data or allocation tables on the affected data set. Try not to delete a data set that contains important information (for example, the single track table (STT) or the JESNEWS data set). If this information is lost, the system issues messages giving you the opportunity to take appropriate actions.



Spool data management access methods

JES3 uses two techniques for storing spooled data: single-record files and multi-record files. Control blocks are recorded as single-record files (SRFs). This means that a control block is recorded as a single buffer image (a block of data on a spool volume). While most of the control blocks will fit into a single buffer image, some (due to their variable lengths and multiple parts) may extend across multiple buffers. When this occurs, the buffers are chained together into chained single-record files.

Multi-record files (MRFs) consist mainly of JCL, SYSIN, and SYSOUT data records, placed into spool buffers, chained together, and written to spool.

The recording and retrieval of spooled data are the responsibilities of three special access methods:

• JES spool access method (JSAM), used by JES3 modules and user-written DSPs to read and write SRFs and MRFs.

• User spool access method (USAM), used by programs in the user address spaces to read SYSIN data and write SYSOUT data.

• The block spooler. The block spooler is used in FSS address spaces to read data buffers for PSF printers.

Any main in the complex can perform I/O to a spool data set. Requests for spool space can originate from the global or a local, but only the global manages spool space.

JSAM

JSAM is a collection of routines invoked by DSPs by macro calls to acquire spool space, create spool files, read and write records, and to purge SRFs and MRFs when they are no longer needed. JSAM buffer pool management procures space for SRFs and MRFs in JES3 private storage.

The single track table 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.

USAM

USAM provides user access to SYSIN data and the creation of SYSOUT data sets. MVS data management access method macros (BSAM, QSAM) pass control to JES3 by the compatibility interface to allocate spool space for output data sets. USAM thereby provides for the opening, closing, reading, and writing of data sets to JES3 spool.

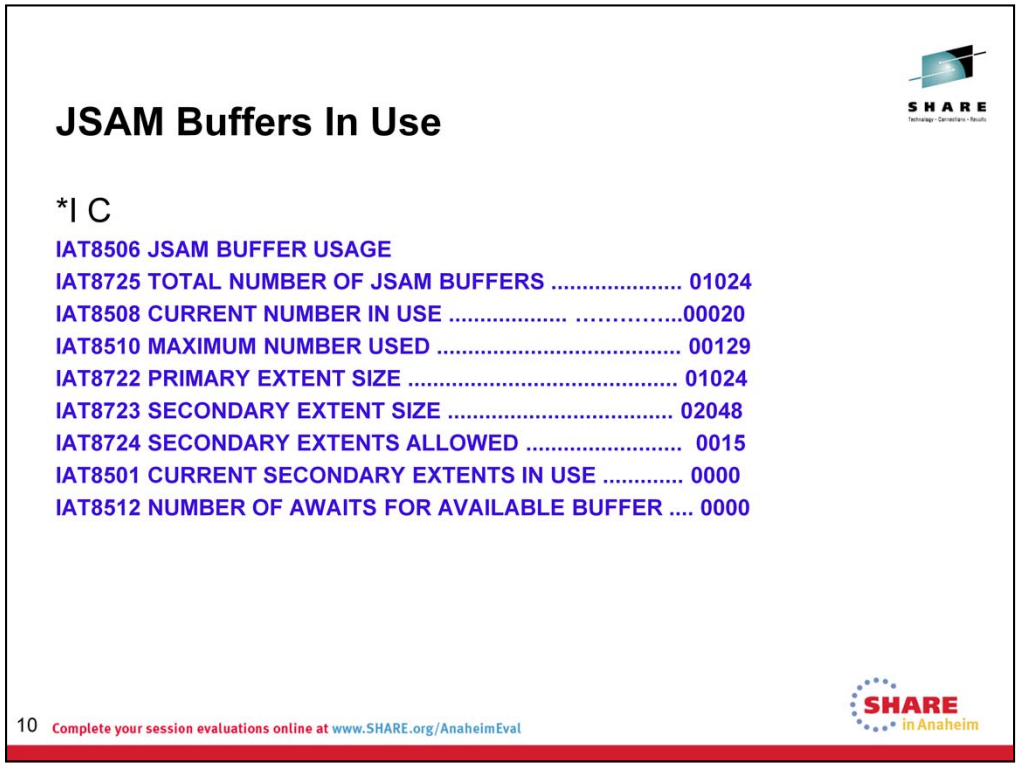
USAM buffer pool management involves user buffer pools. The user storage buffer pools are inside the user address space and consist of either one page of virtual storage for a SYSIN data set or multiple pages for SYSOUT data sets. The contents of user storage buffers are not transmitted directly to or from spool volumes, but rather are moved to or from the USAM protected buffer pools that reside in the common service area (CSA) or in a JES3 auxiliary address space (AUX).

Block spooler

The block spooler enables a program to access spool data sets from a JES3 subtask or other address space that cannot use JSAM services to access spool.

For example, writer FSS input routines invoke the block spooler to read blocks of SYSOUT data from spool and to read and write writer checkpoint records that are used by FSS writers. The block spooler reads data one track at a time (when possible) into buffers in the USAM buffer pool.

The block spooler is also used to read the LVS for locate subtasks and read and write the JST for MDS subtasks.



JSAM buffers in use

Use the *I C command to display the current status of the JSAM buffer pool. The display includes statistics such as current and maximum use counts for these pools.

Secondary extent size

JES3 automatically expands the size of the JSAM buffer pool when it detects a buffer shortage. For each expansion, JES3 increases the size of the buffer pool by one half of the value you specify on this parameter up to a maximum of four times or 32,767, whichever occurs first. For additional information about determining the size of the JSAM buffer pool in *z/OS JES3 Initialization and Tuning Guide*, SA22-7549.

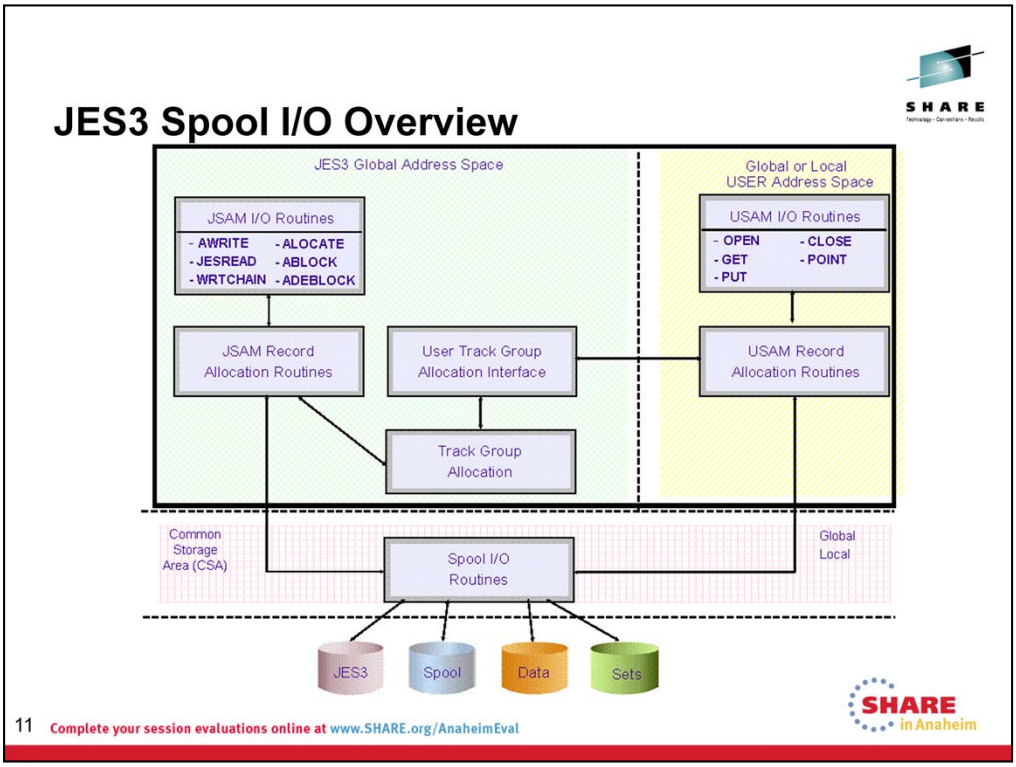
When JES3 determines that all of the buffers in the primary allocation have been exhausted, it automatically expands the buffer pool by allocating additional buffers. Although JES3 provides a secondary allocation of buffers each time an out-of-buffer condition exists, JES3

will expand the buffer pool up to a maximum of 4 times or 32767 bytes, whichever occurs first.

The number of buffers that JES3 allocates for each secondary allocation is one-half the number specified on the PAGES= keyword of the BUFFER initialization statement.

JES3 frees the secondary allocations of buffers when the number of available buffers is greater than or equal to the number of buffers in the secondary allocation plus the acceptable minimum number of JSAM buffers. To find the minimum number for your installation, divide

the total number of JSAM buffers (the primary allocation buffers from the PAGES= keyword plus all secondary buffer allocations) by the MINBUF= parameter.



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).

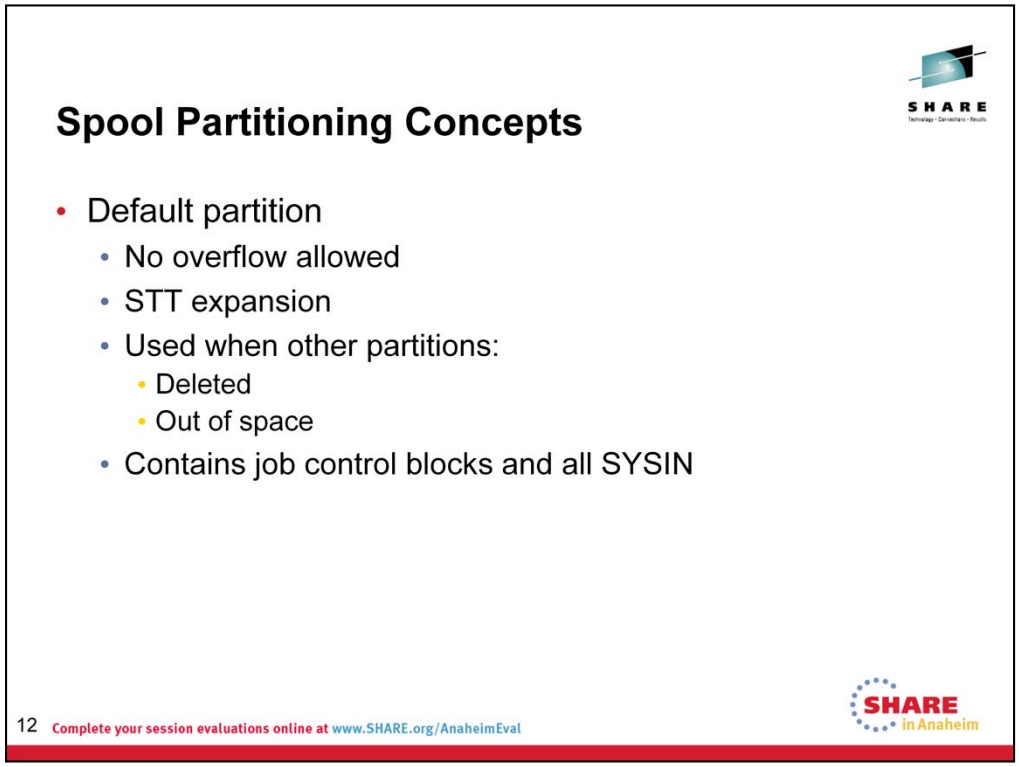
The management of spool space involves the assigning of spool space to jobs and JES3 functions and the recording of the ownership of that space. Spool space comprises data sets on specified DASD assigned to JES3. The following summary examines the different gradations of JES3 spool space from the smallest to the largest segment.

•A *record* is a physical block on a disk; the single smallest unit of spool space.

•A *track group* is a contiguous group of records within a spool data set. The track group is the basic unit of spool space management. (The JES3 initialization stream provides options to control how spool space will be allocated and options to define the size of a track group.)

•A *spool data set* (extent) consists of contiguous tracks on DASD devices. Each data set is generally an entire DASD volume.

•A *partition* is a group of spool data sets. The partition is used by spool space management to manage track groups within the data sets.



Partitioning concepts

Using SPART initialization statements, you can define up to 1024 spool partitions. Additionally, you can identify one of the partitions as the default partition by specifying DEF=YES on a SPART statement.

You do not need to define spool partitions or specify a default partition. If you do not define a spool partition (do not include SPART statements in the initialization stream), JES3 creates a minimum of three spool partitions. JES3 assigns spool data sets to one of the following partitions:

DRAINED Result of an operator drain, hold, or cancel command.

UNAVAIL For data sets unavailable to JES3 during JES3 initialization.

DEFAULT JES3 names the default partition JES3PART.

If you define partitions but not a default partition, JES3 uses as the default partition the partition defined on the first SPART statement in the initialization stream.

Default partition

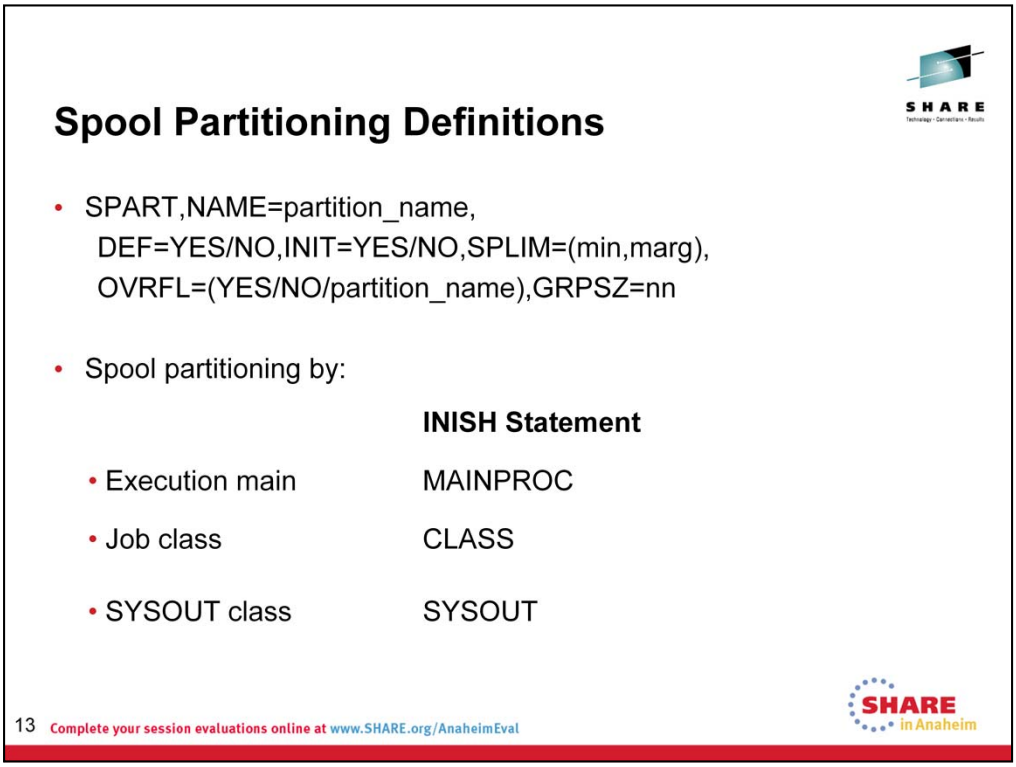
The default spool partition always contains:

- JES3 spool access method (JSAM) single and multi-record files
- Job input (SYSIN) data
- JES3 control blocks created by input service

It may also contain output spool data for:

- Jobs requesting a spool partition with no free space that overflows into the default partition
- Jobs requesting a spool partition that has been deleted

- Jobs that do not request a spool partition and for which the job class and processor have no spool partition designation



A *partition* is a group of spool data sets. The partition is used by spool space management to manage track groups within the data sets.

The SPART initialization statement has a parameter that defines the action to be taken when the spool space in that partition is exhausted.

SPART,...OVRFL=(yes|no|spart),...

Where:

yes This specifies that the spool partition may overflow into the default spool partition. This is the default for this parameter.

no This specifies that the spool partition may not overflow. This only affects allocations requested by USAM. Jobs requesting additional spool space will wait. JSAM

requests to the partition will overflow into the default spool partition.

spart This specifies the name of the spool partition into which spool space allocation will overflow.

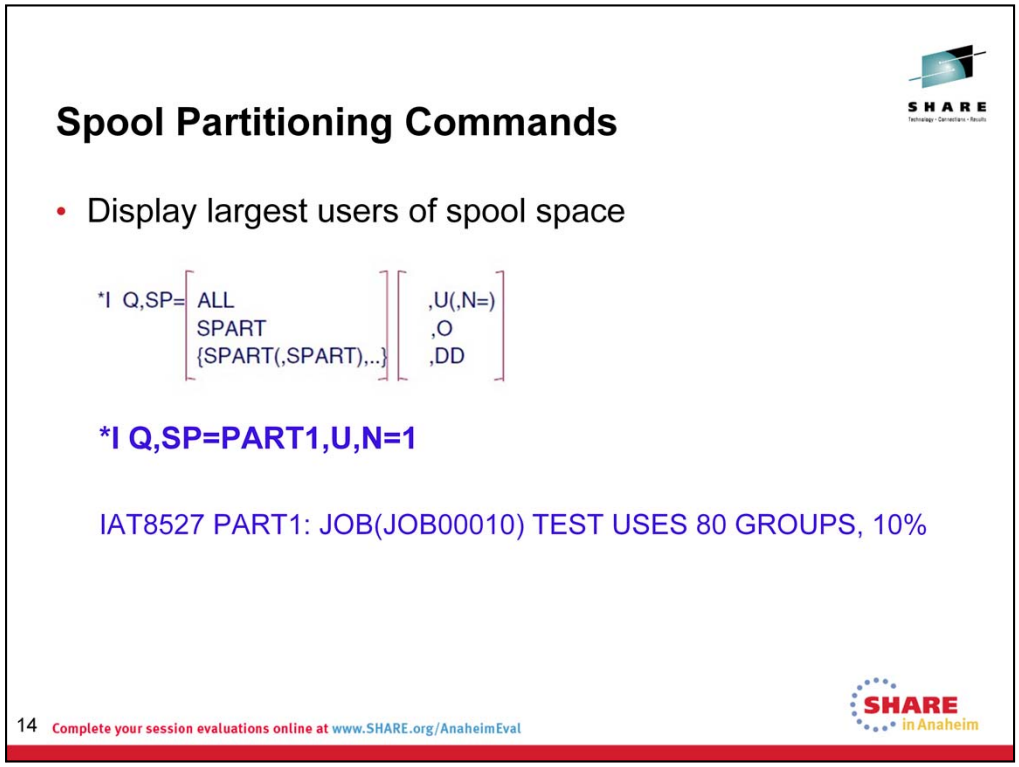
When implementing spool partitioning:

• If OVRFL=YES is not specified on any SPART statement, all out-of-space allocation will overflow into the default partition. Overflow into other than the default partition is not permitted.

• The OVRFL parameter is ignored if it is specified on a default spool partition definition.

• Code OVRFL=NO to prevent output data sets designated for one partition to

overflow into another partition. If it becomes necessary to increase spool space in an out-of-space partition, another spool extent can be moved into the partition.



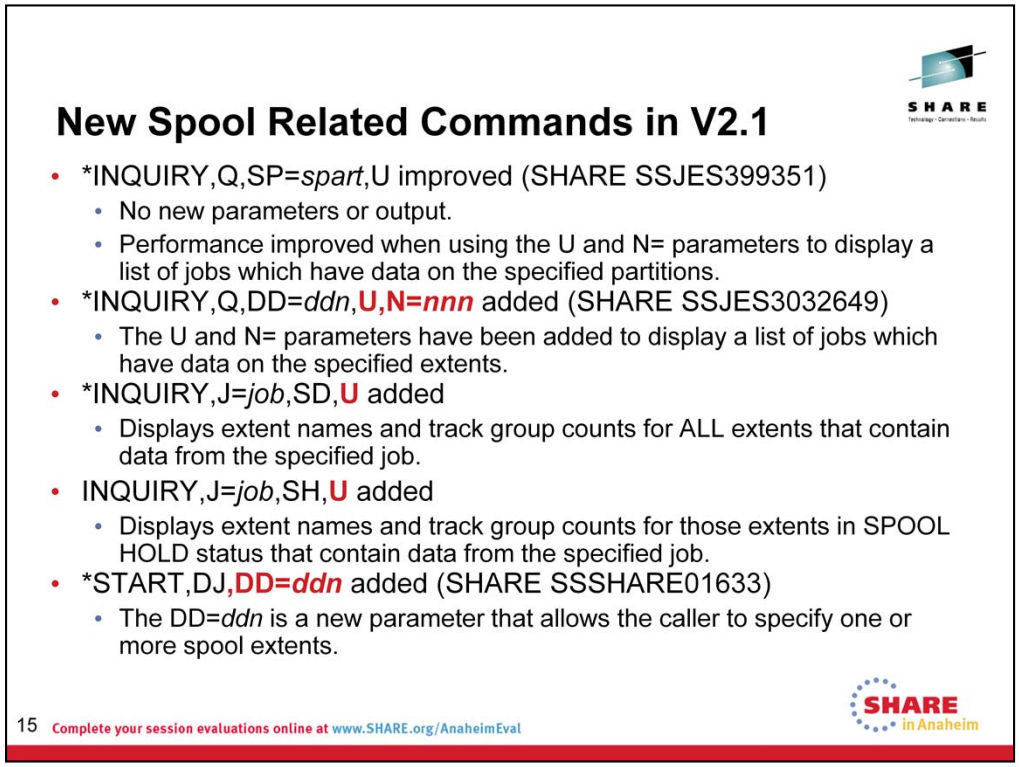
Partitioning commands

You can use the ***I Q** command to identify the spool data sets assigned to a particular partition and to determine if the partition is an overflow partition, the default partition, or the initialization partition. You can also display the size of a partition, the amount of space that is currently available, and the largest users of spool space.

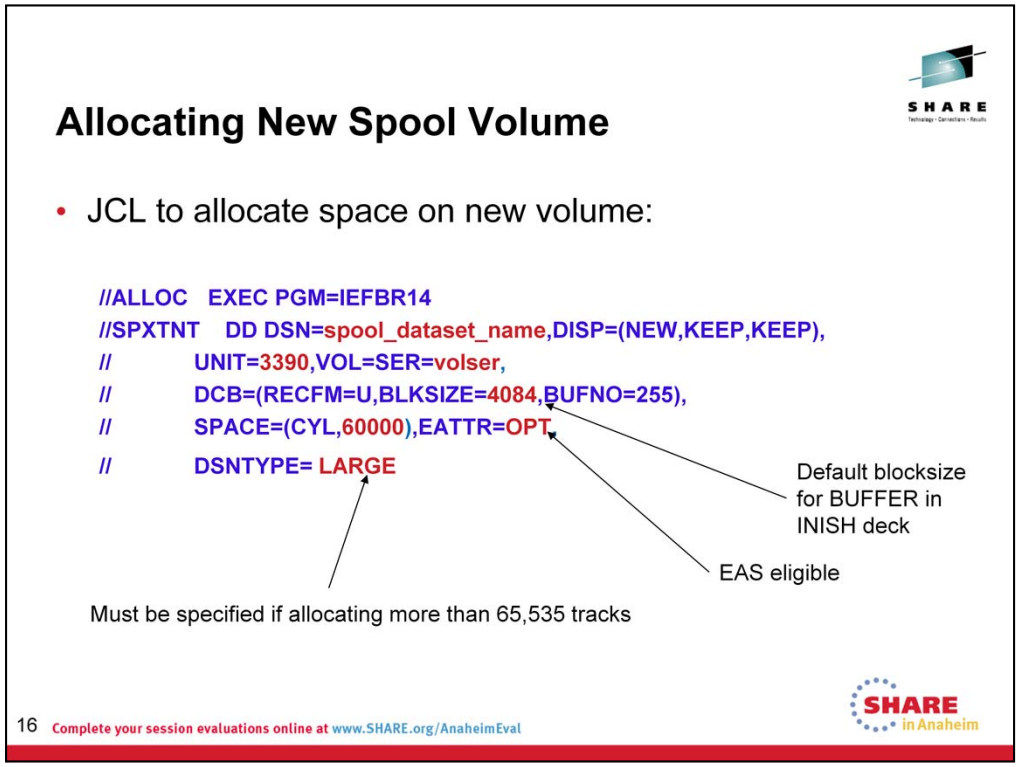
You might want to use this command to help determine if a performance problem is the result of JES3 using a high percentage of the available spool space in one or more spool partitions.

Use the ***I Q** command to display:

- A list of jobs waiting for a DSP
- The names of the spool data sets assigned to a spool partition and whether the partition is defined as the default partition, the initialization partition, or an overflow partition
- The size of the partition and the amount of space currently available
- The status of a partition, and users of the largest spool space
- The status of a spool data set and the name of the spool partition the data set belongs to
- All the defective tracks currently known to JES3
- The amount of space available on all the JES3 spool data sets in the complex



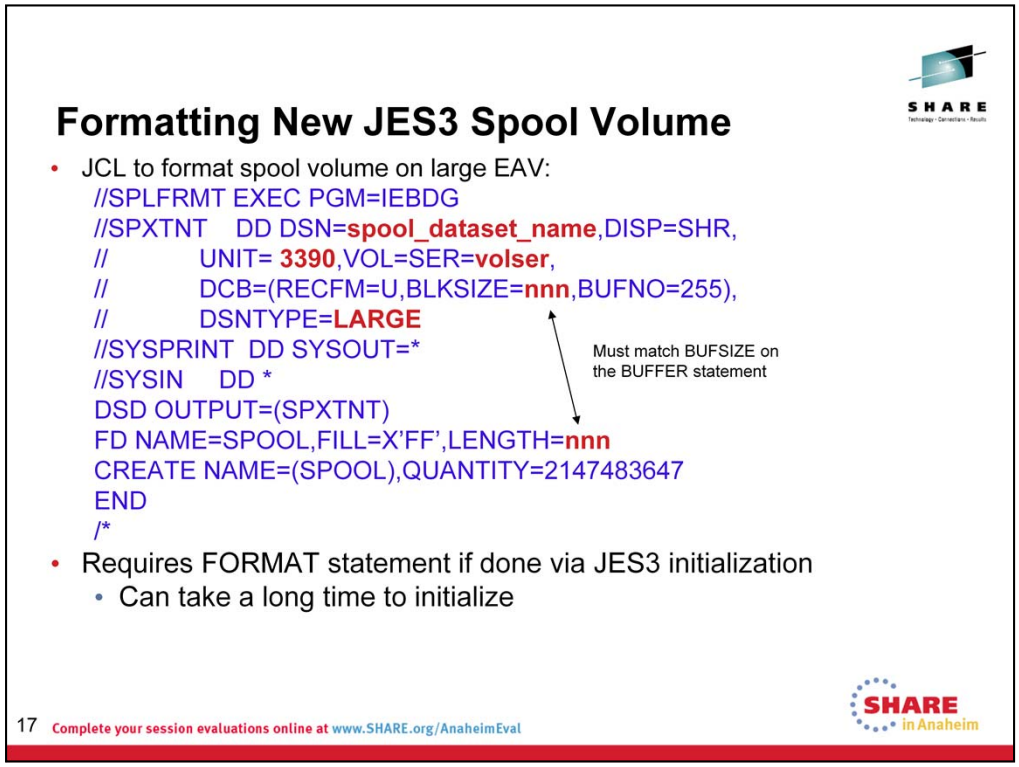
More detail of these commands can be found in the JES3 V2.1 Update session.



A spool data set can be greater than 65,535 tracks only if you specify the DSNTYPE=LARGE keyword on the JCL DD statement that is used to create the data set.

A spool data set can be created with extended attributes so that it is EAS-eligible; then the spool date set can exist in the Extended Addressing Space (EAS) of an Extended Address Volume (EAV). A spool data set can be EAS-eligible only if you specify the EATTR=OPT keyword on the JCL DD statement used to create the data set.

Creating or using a data set with extended attributes requires z/OS V1R12 or later releases. A data set with extended attributes and residing in the EAS cannot be opened on earlier-level systems.

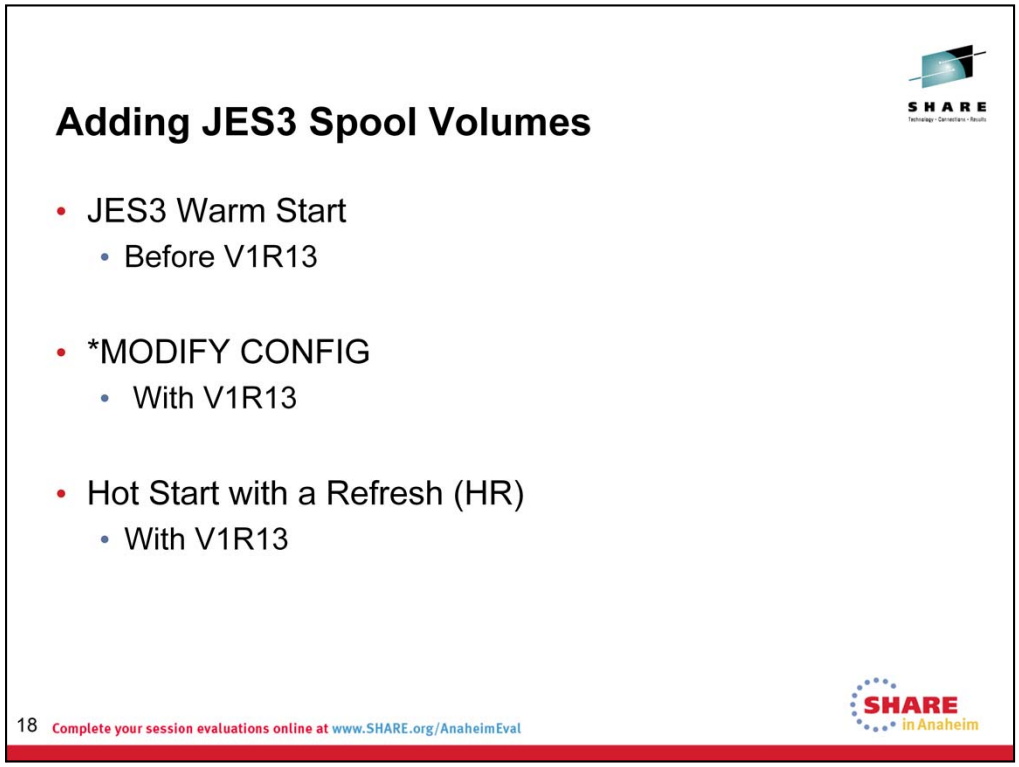


The value of the variable *nnn* must equal the value of the BUFSIZE= parameter on the BUFFER initialization statement. The variable *nnn* appears on both the SPXTNT DD statement and on the FD utility program control statement.

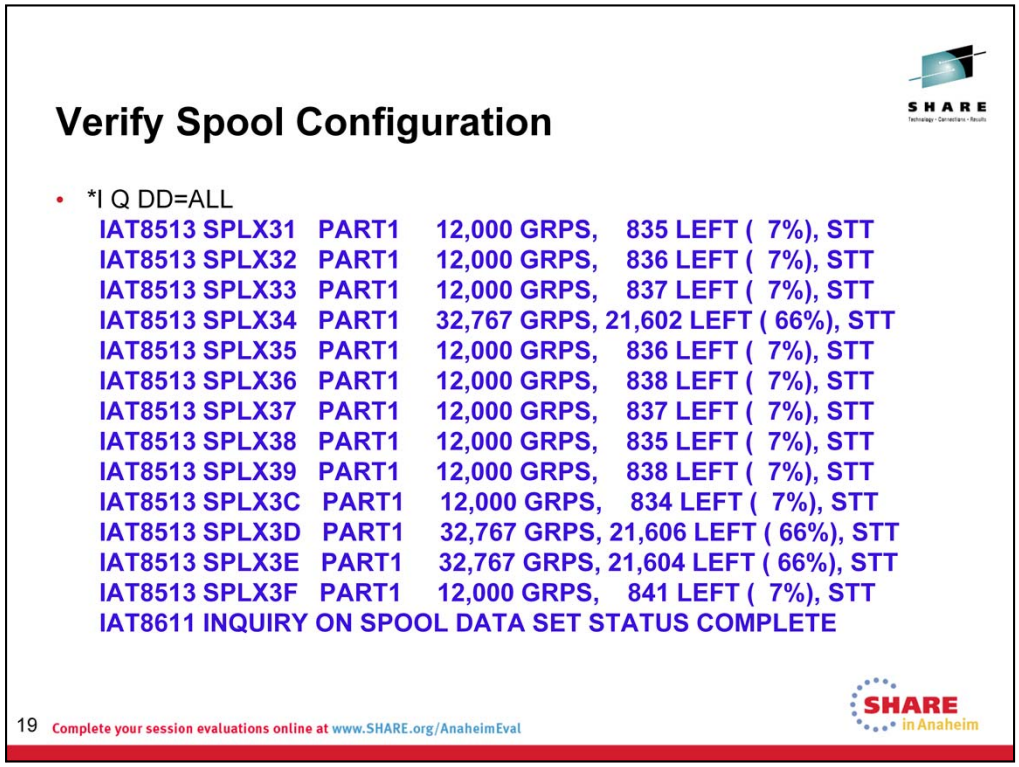
The value of the QUANTITY=2147483647 on the CREATE statement should be specified to ensure that IEBDG formats the entire data set.

If IEBDG successfully formats the entire spool data set, the formatting job ends with an abend code of D37. In addition, MVS issues message IEC031I. Ignore the corrective action specified in the message.

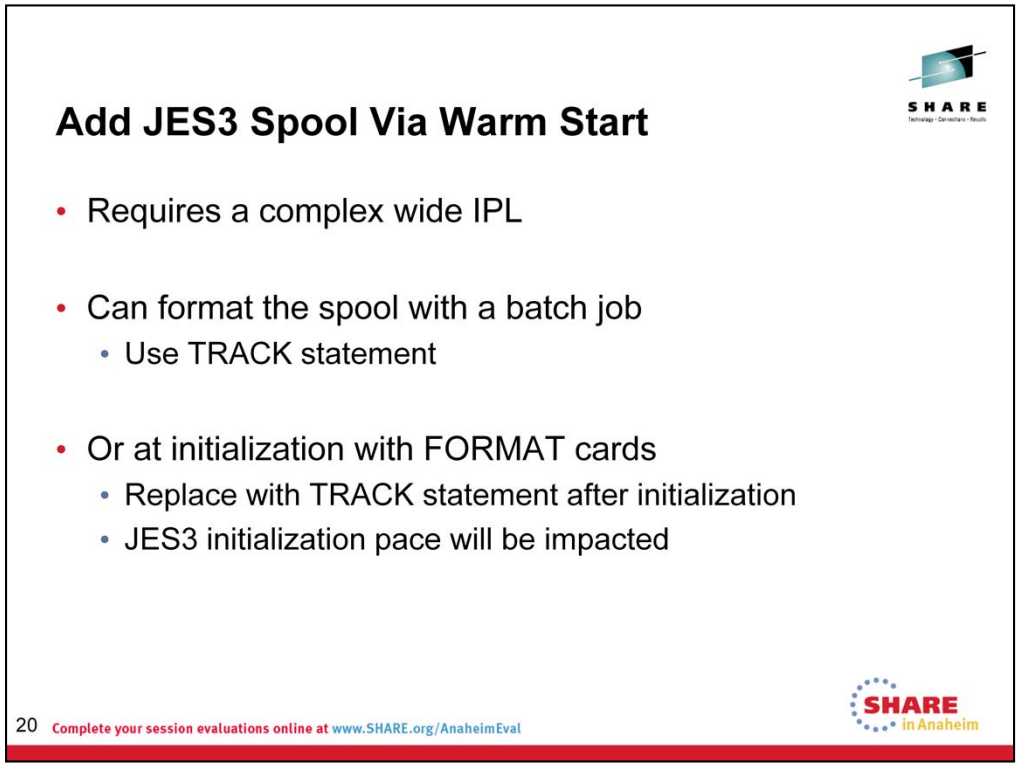
The format can also be done via JES3 initialization by using the FORMAT statement in the INISH deck. This process can be very slow and will inhibit a quick initialization of JES3.



Lets review the different methods to add and spool volumes, starting with the method used before the new implementations for ADD.



Ensure the current configuration is understood. Note is the above example that the Single Track Table (STT) is on all volumes.



The type of start you use depends on why you are formatting the spool data set:

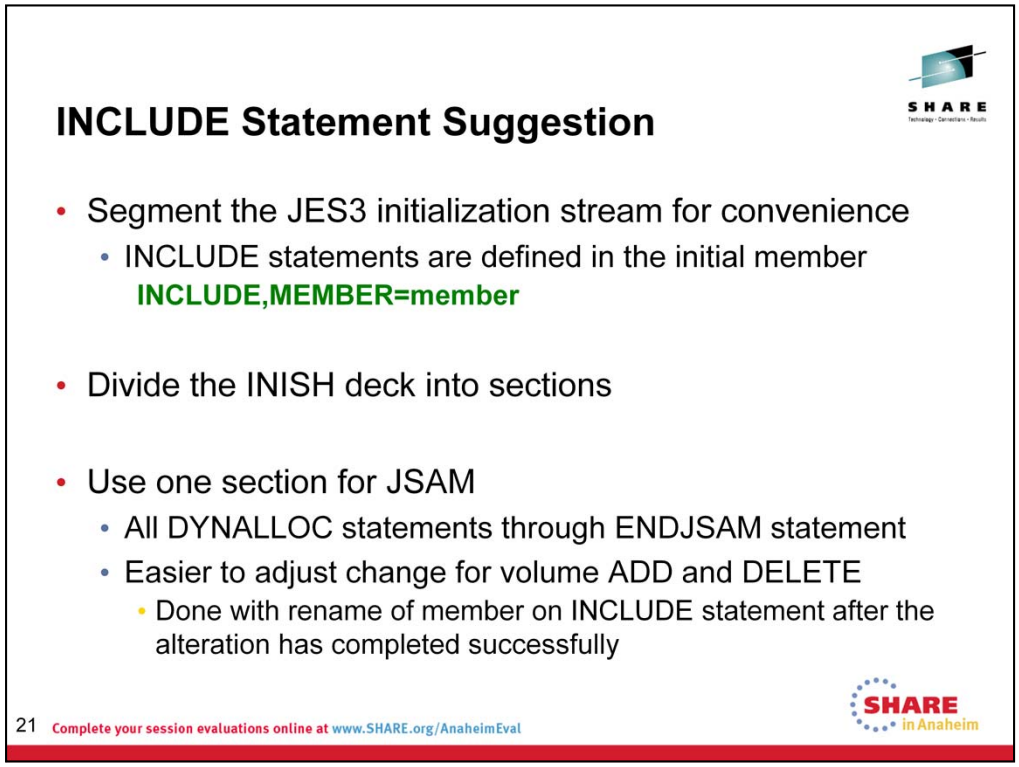
If you have changed the BUFSIZE= parameter on the BUFFER statement, use a cold start (C). (In this case, you must format all spool data sets.)

If you are replacing a spool data set, use a warm start to replace a spool data set (WR). If you also want an analysis of the jobs in the job queue, use a warm start with analysis to replace a spool data set (WAR).

If you are adding a spool data set, use a warm start (W) or a warm start with analysis (WA).

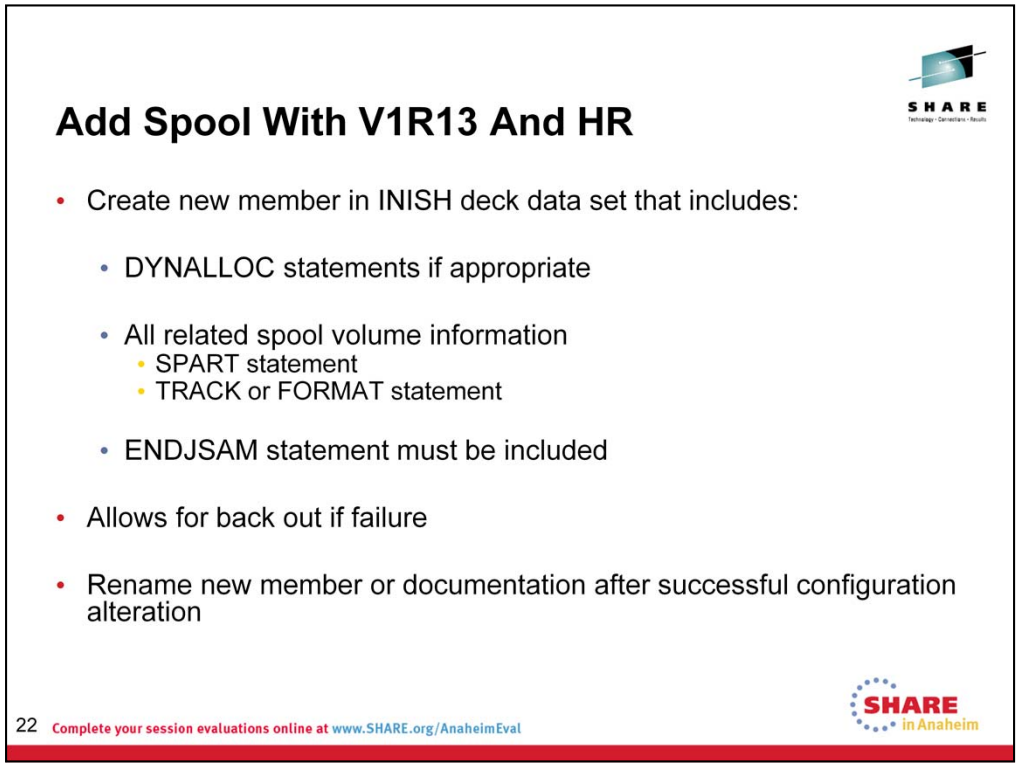
After JES3 processes the initialization stream, replace the FORMAT statement with a TRACK statement. If the FORMAT statement contained the STT or STTL parameter, also code this parameter on the TRACK statement.

If you use a warm start and the initialization stream contains a FORMAT statement for a spool data set that is already formatted, JES3 issues a warning message. JES3 continues with initialization, however, and does not reformat the spool data set.

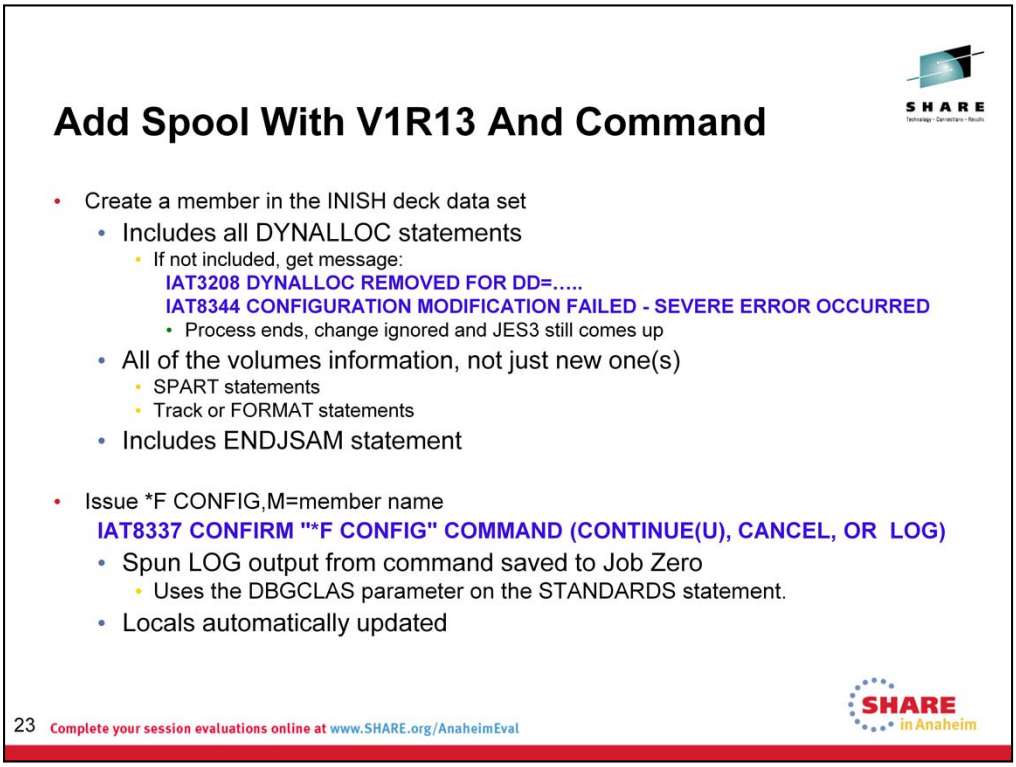


Use the INCLUDE statement to include a *member* in the initialization stream member. Different sections of the initialization stream can be put into different members and included in the primary initialization stream member. The *member* is the PDS member name within the data set specified on the JES3IN DD statement in the JES3 procedure to be included. Up to four member levels can be used (the primary initialization stream member and up to three INCLUDE level members) with the exception of the INCLUDE that contains the DYNALLOC statements where only a single level of nesting is supported.

Note: The INCLUDE statement cannot be used if JES3IN DD data set is concatenated, and the members to be included are in a data set other than the first in the concatenation.



In z/OS JES3 V1R13, one can add a spool volume by issuing a command. Create a new member in the allocated INISH deck data set that includes all of the DYNALLOC statements currently used (including statements for PROCLIBS), all the current spool volume configurations, and the new spool volume configuration.



On the *F CONFIG command, LOG=YES or LOG=NO or LOG=ERR

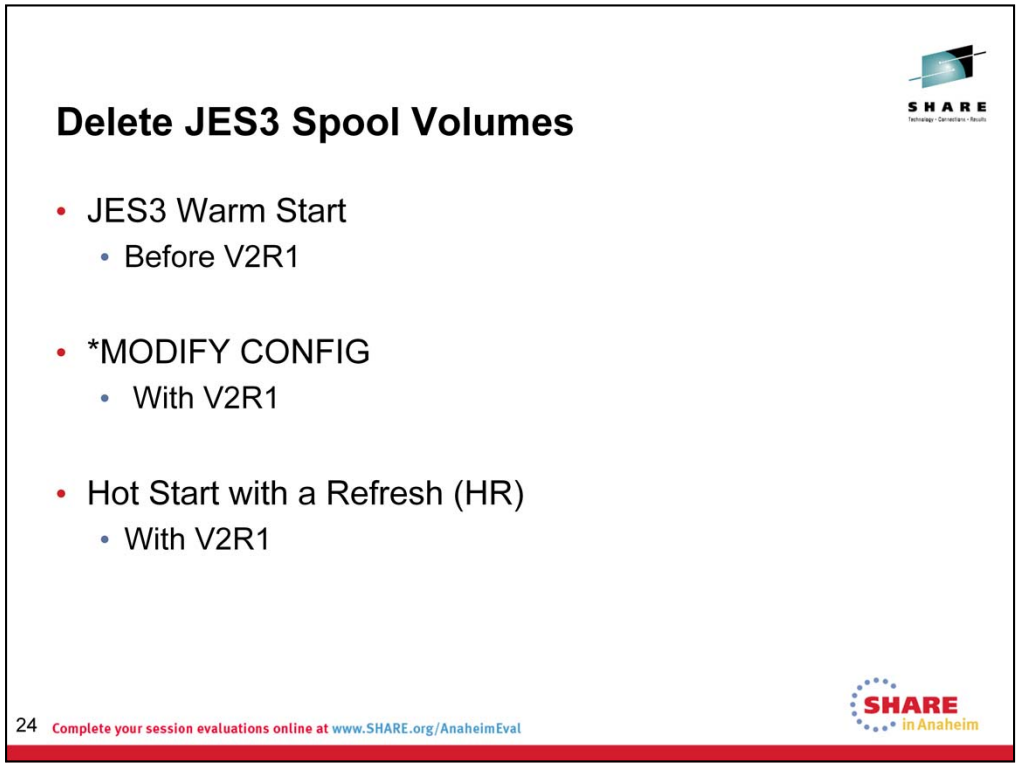
Specifies whether you want a log data set generated. The log data set contains the initialization statements and any error messages that are generated. The log data set is then spun off for printing at the end of *MODIFY,CONFIG processing.

YES Create the log data set.

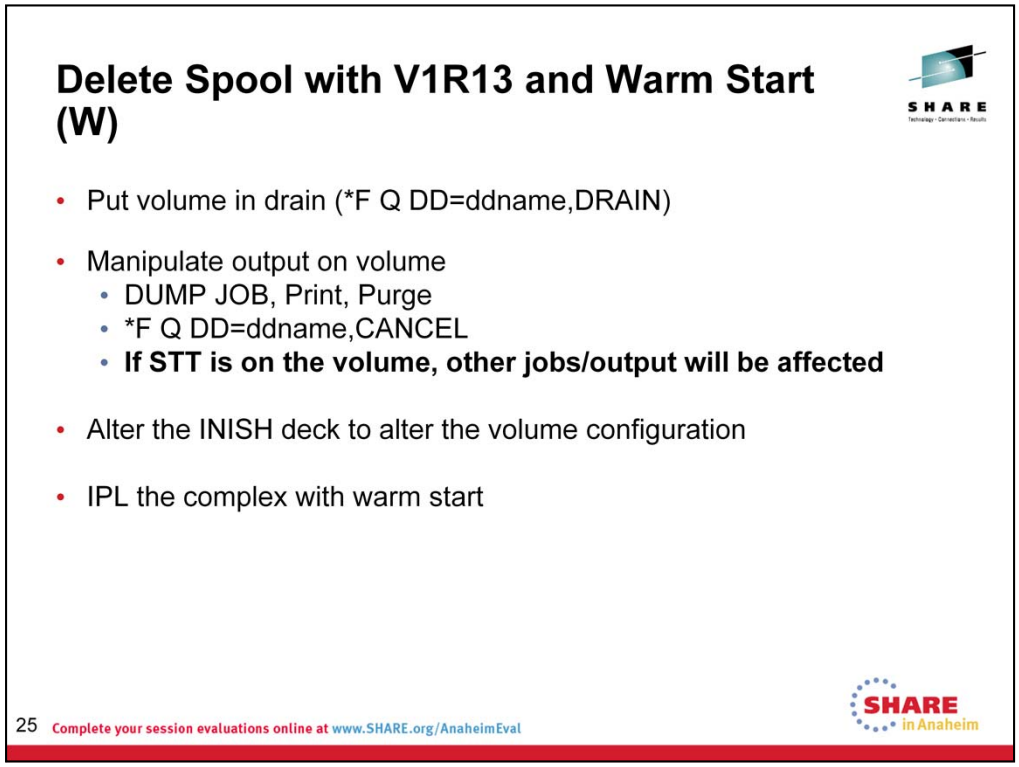
ERR Create the log data set only if an error occurs.

NO Do not create the log data set. All error messages will be displayed on the issuing console.

YES is the default.

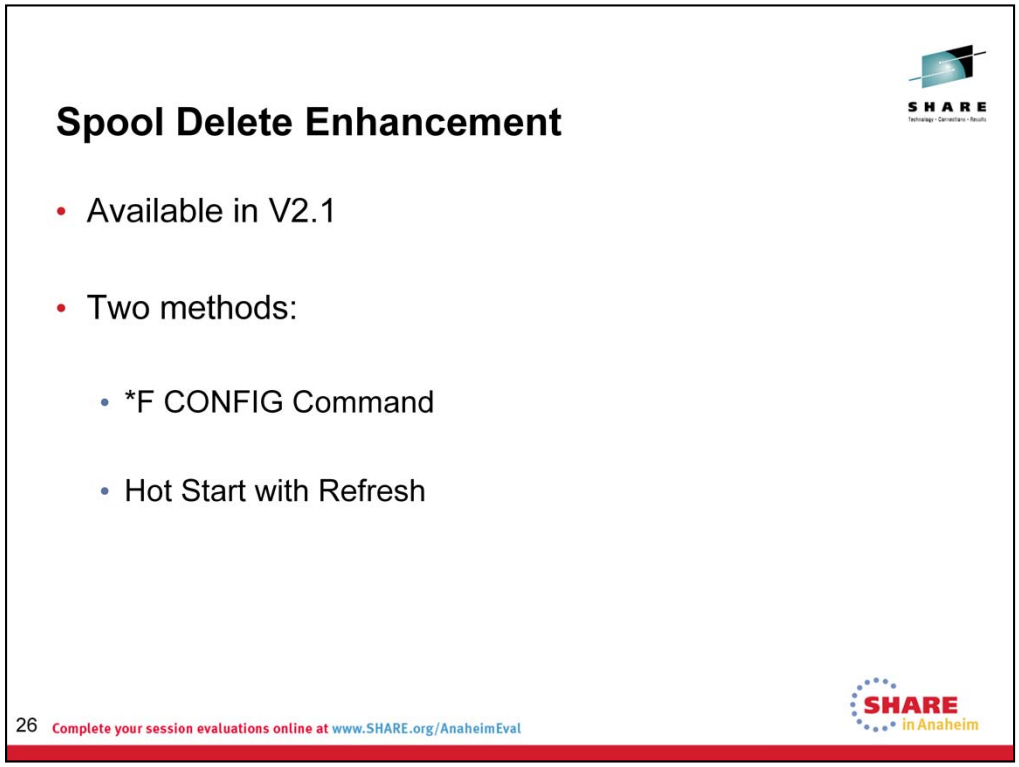


Now lets review the methods to delete a JES3 spool volume before V2.1 and then look at V2.1.

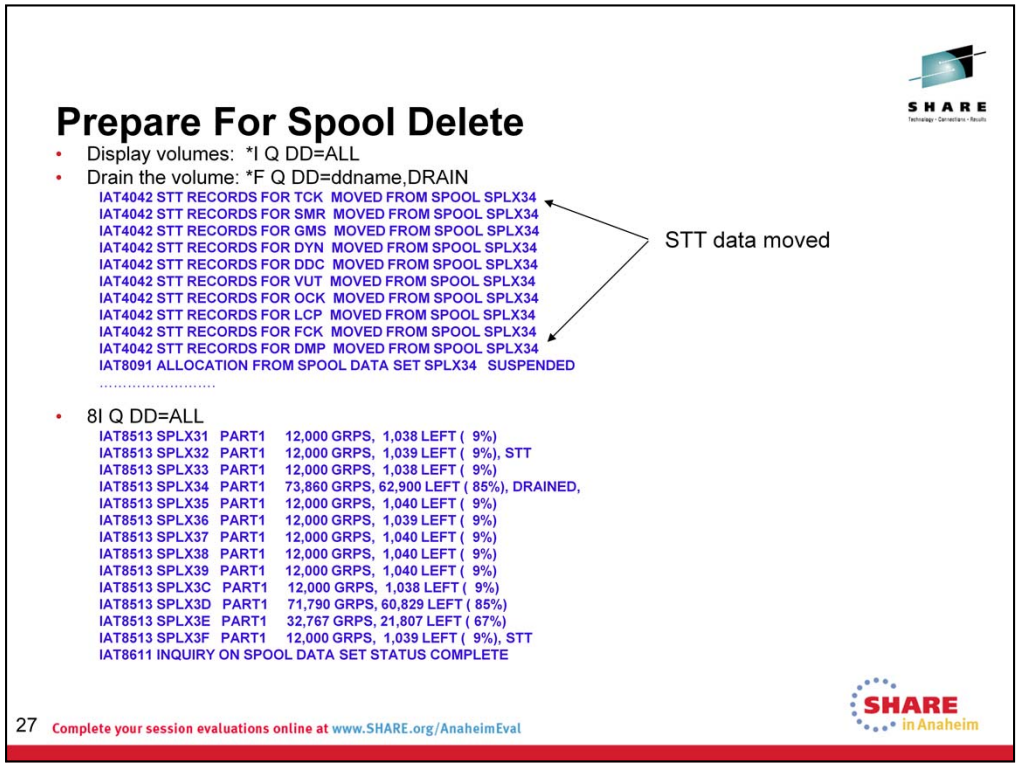


Place the spool volume in DRAIN to stop new work from allocating space on the volume. Remove all output from that spool volume by either using JES3 Dump Job, printing the output, or purging it. Finally, issue the CANCEL command for the spool. This command holds the spool abnd cancels all jobs that have data on the data set. If the Single Track Table (STT) is also on the volume, any job that is defined in it on that volume is also cancelled and purged.

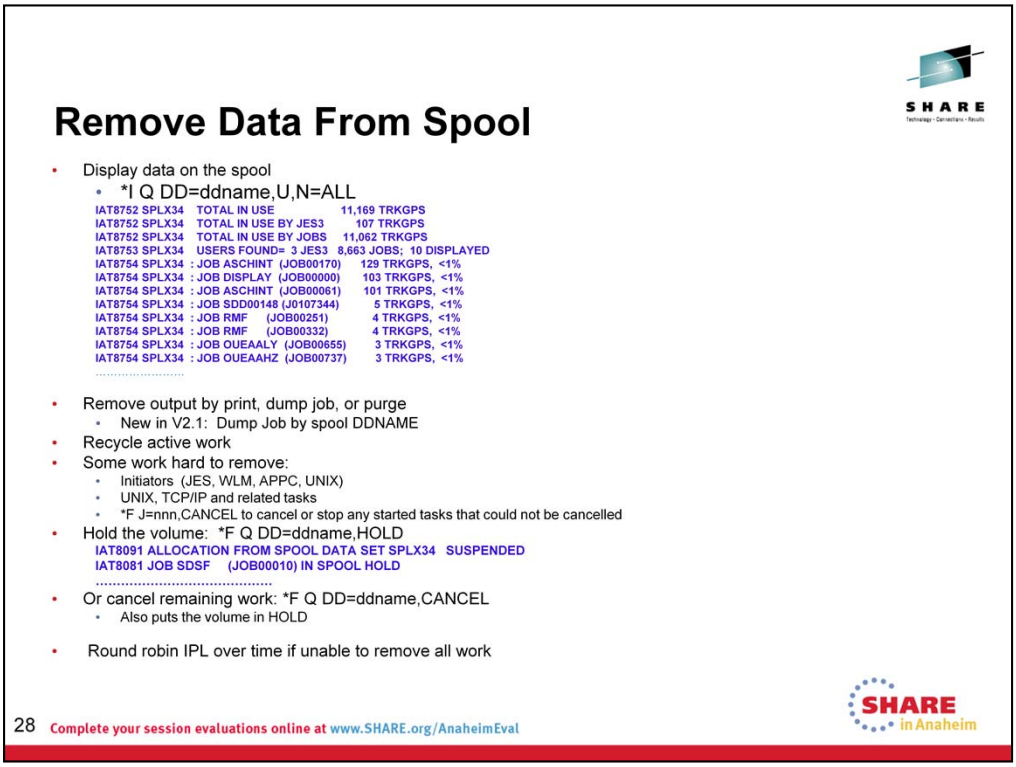
Create a copy of the JES3 initialization stream (INISH deck) without the spool volume definitions and warm start the complex.



A cleaner method of deleting the spool volume is available in V2.1 JES3. It can be done either of two ways: via a command or with a JES3 Hot Start with Refresh (HR).



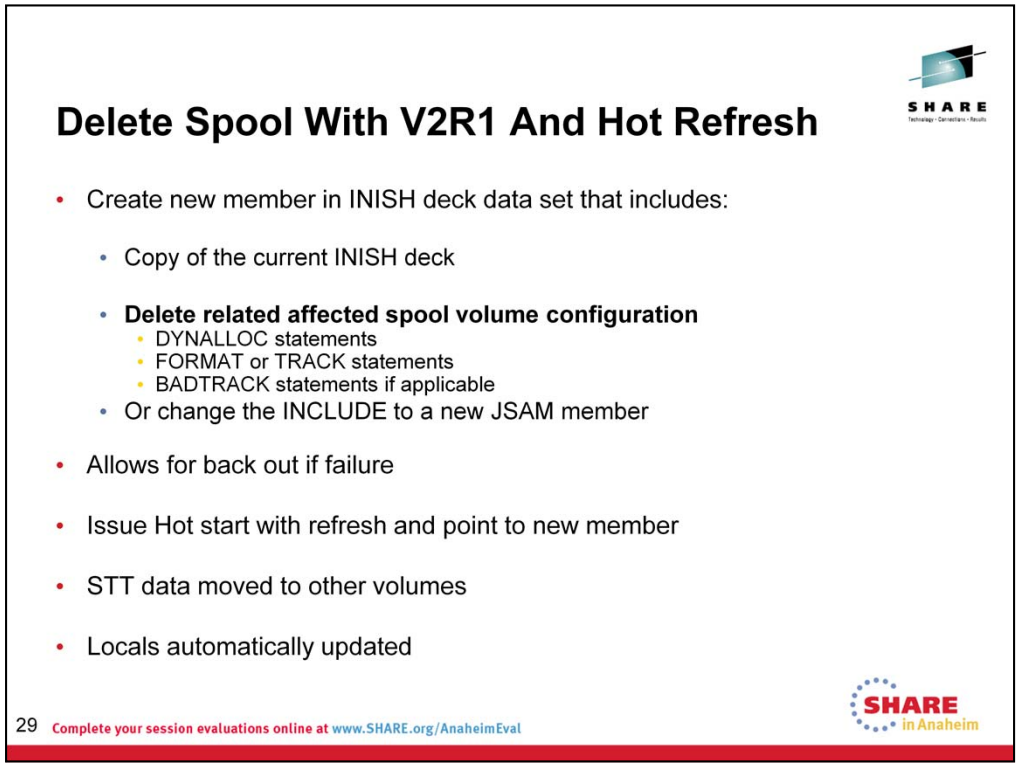
Validate your spool configuration and then place the spool volume in DRAIN. If the STT is on this volume, JES3 will move the data from it and place it on other volumes. Then place the volume in HOLD.



In V2.1, there is an enhancement that allows for a display of all the jobs on any specific spool volume. It makes no differentiation to whether the jobs is active or if it is only output.

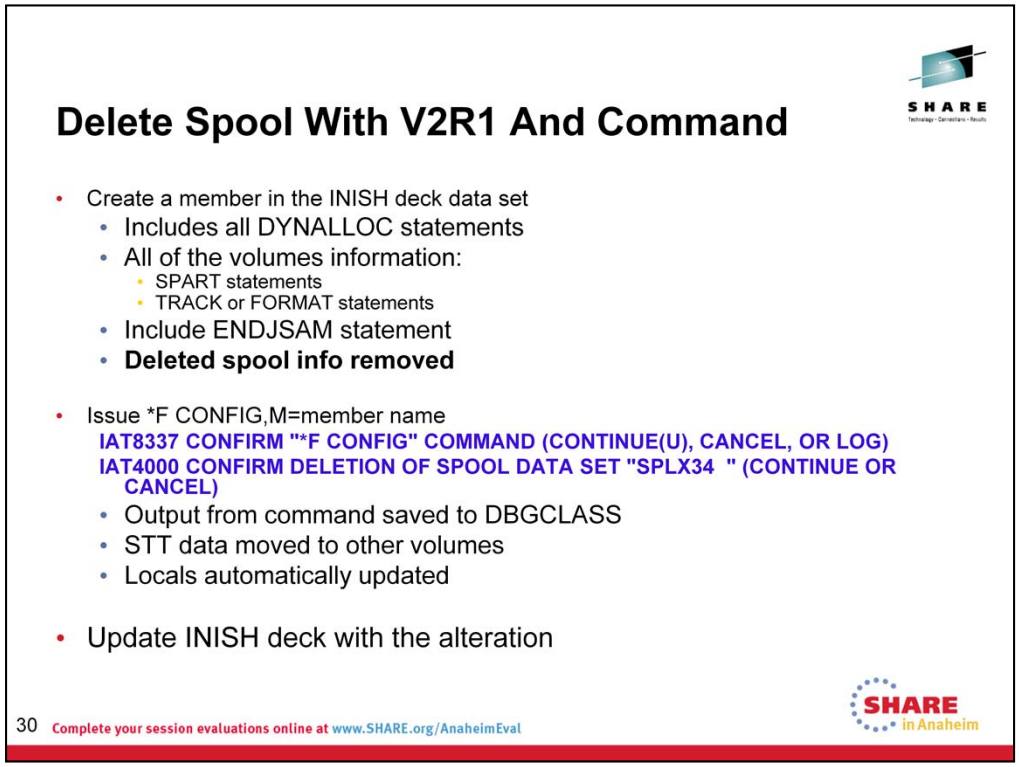
Remove the output by either doing a JES3 Dump Job, printing the output, or purging it. Recycle any active work that is on the volume. Then issue the CANCEL command to cancel any remaining work on the spool volume.

Some work is difficult to remove or unfeasible due to business needs so it one method to remove the final work is to do a round robin IPL of every system over a period of time to remove the active work. Then, the output from those jobs can be managed.



Once the spool volume is cleared of data, it can then be deleted. One method is the JES3 Hot start with Refresh. Create a copy of the current INISH deck member with a new suffix into the same data set and then delete the affected spool volume configuration. If JES3 does not like the configuration, you can always hot start back onto the original member.

Recycle the JES3 global with a hot start with refresh. There are some verification messages at initialization and the locals will automatically be refreshed with the change.



The other method for deletion of the spool volume is via a command that points to a new member in the INISH deck which includes just the spool volume configuration changes. Be sure to include all of the DYNALLOC statements and the ENDJSAM statement. The output from the command defaults to a spin-off data set under job zero. The JES3 Locals are automatically updated with the change.

Questions?