z/OS Version 1 Release 2

JES2 Version 1 Release 2

JES2 Enhancements z/OS 1.2

- Greater than 64K job numbers
- Long running jobs support
- Dynamic proclib support
- Initialization statement members
- Large spool support
- Migration concerns

Greater than 64K Job Numbers

- Various limits being increased
- CKPT size limits need increasing
- New limits break 3 byte offset pointers
  - JQE chains, JOE chains, JOE to JQE chains
- Job numbers (binary) increase to 4 bytes
- JQE extensions moved out of master record
- JOBIDs change based on upper limit on range
  - < 100,000 then format unchanged (JOBnnnnn)
  - >= 100,000 then format is J0nnnnnn

Limits Increased when in z2 Mode

JOBDEF	JOBNUM limit 200,000
JOBDEF	RANGE limit 999,999
OUTDEF	JOENUM limit 500,000
CKPTSPACE	BERTNUM limit 500,000
SPOOLDEF	TGSPACE=MAX limit 16,580,355

- These values can be decreased - Only from z2 mode
  - JOBNUM - JOENUM - BERTNUM
- JOBIDs are changed depending on limit value
  - STC12345 becomes S0123456

New $ACTIVATE Level - z2

- New $ACTIVATE level (z2 or R4)
  - Two modes of operation
    - Compatibility mode (Release 4 mode) - LEVEL=R4
    - New exploitatioin mode (z2 mode) - LEVEL=Z2
    - Can switch to either mode via a command
- When in z2 mode
  - Binary job numbers are 4 bytes
  - JOE and JQE chaining is by 3 byte index
  - Checkpoint data set format changes
  - JOBID is J0nnnnnn if range over 99,999

Checkpoint mode commands

- Command to display current mode

```
$DACTIVATE
$HASP895 $DACTIVATE 513
$HASP895 JES2 CHECKPOINT LEVEL IS NOW OS/390 RELEASE 4
$HASP895 A TOTAL OF 1087 4K RECORDS ARE REQUIRED FOR $ACTIVATE.
$HASP895 ALL INUSE=YES DATA SETS ARE AVAILABLE AND LARGE ENOUGH.
$HASP895 $ACTIVATE WILL SUCCEED IF ISSUED FROM THIS MEMBER
```

- $ACTIVATE command without subparameter

```
$ACTIVATE
$HASP003 RC=(12),IVATE  - MISSING REQUIRED OPERAND LEVEL
$HASP895 JES2 CHECKPOINT LEVEL IS NOW OS/390 RELEASE 4
```

- Command to change current mode

```
$ACTIVATE,LEVEL=Z2
$HASP895 z/OS 1.2 LEVEL IS NOW ACTIVE
$HASP895 JES2 CHECKPOINT LEVEL IS NOW z/OS 1.2
```

$Activate Considerations

- Increase limits after V1.2 is stable
- If problems arise with new limits
  - Decrease the limits with operator commands
- JES2 parameter UNACT can control start processing
  - PARM=(COLD,UNACT)
    - Cold starts in R4 mode
  - PARM=(WARM,UNACT)
    - Switches to R4 mode
  - Note: $ACTIVATE is the preferred way to switch

64K Support for Max JQEs

- Changing maximum number of jobs allowed
  - $T JOBDEF,JOBNUM=200000
  - Checkpoint size is validated

```
$T JOBDEF,JOBNUM=200000
$HASP296 MEMBER SC59  -- CKPT1 SYS1.HASPCKPT ON OP1TS2 - SPACE 592
         INSUFFICIENT -- 944 TRACKS NEEDED
$HASP003 RC=(74), 593
$HASP003 RC=(74),JOBDEF  - CURRENT CHECKPOINT DATA SETS ARE TOO
$HASP003         SMALL
```

Job Number Range

- Transition period if job number range increased above 99,999 via $T command
  - Spool and running jobs will have old format
  - Operator commands have new format
- When job number range changes:
  - *T JOBDEF,RANGE=(1-999999)
    - JOBnnnnn becomes J0nnnnnn
    - STCnnnnn becomes S0nnnnnn
    - TSUnnnnn becomes T0nnnnnn

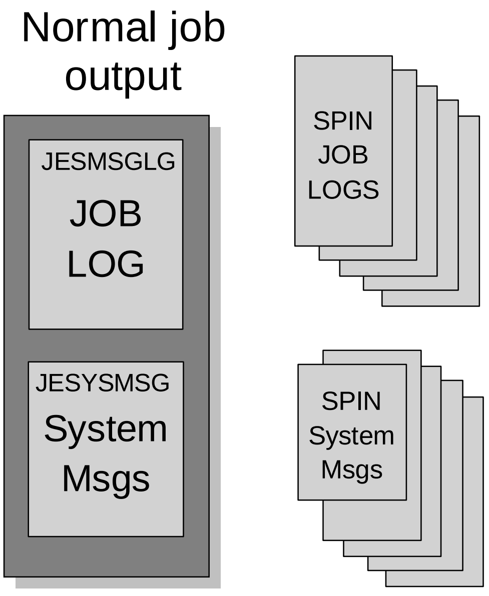
Jobs from NJE

- Job number assigned to NJE job depends on:
  - JOBDEF   REASSIGN=YES
    - Can assign a number outside of range
    - Assign original number if available - If number > 99999
    - New JOBID format used

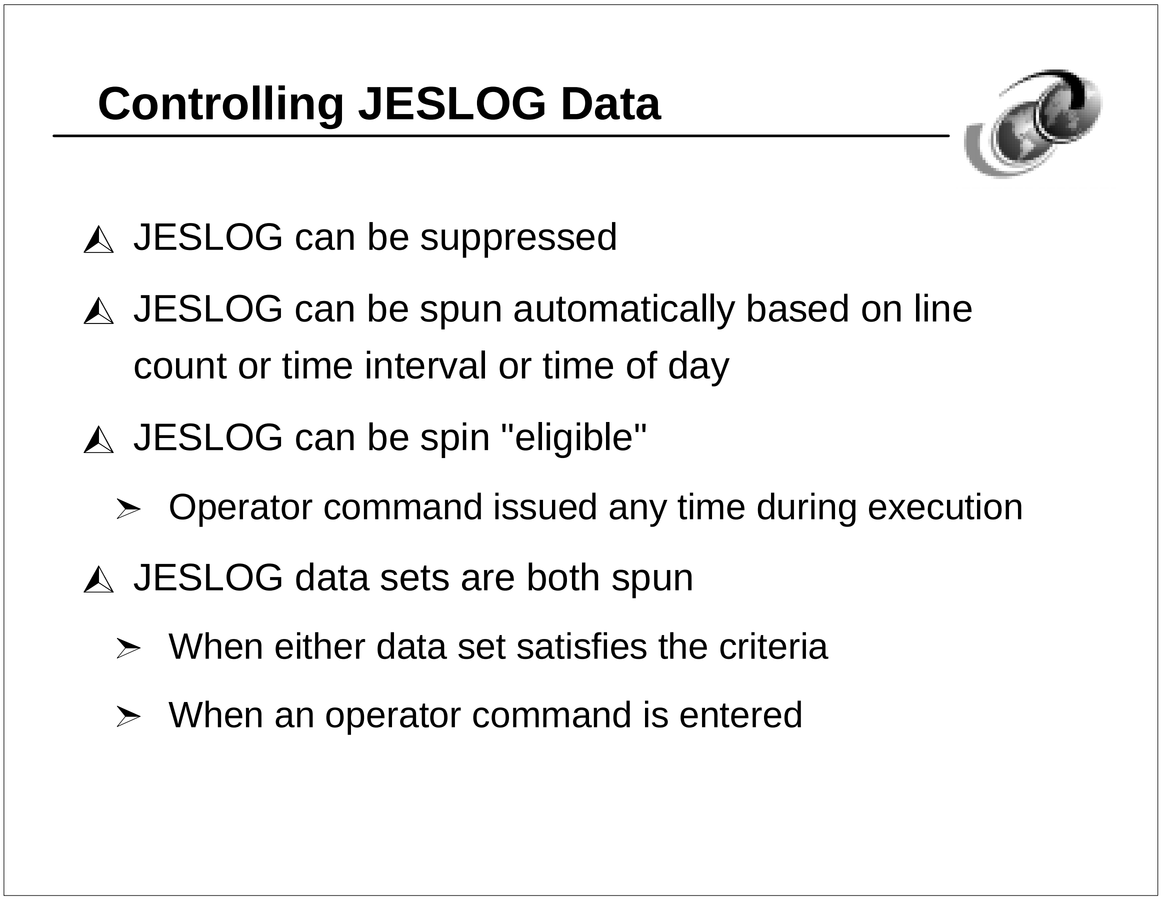
Long Running Jobs Support

- The JESLOG (JESMSGLG and JESYSMSG) data sets for long running jobs:
  - Use spool space which cannot be released until the job ends execution
  - Ending execution usually implies an IPL
- The JESLOG data sets use space even if:
  - A dummy MSGCLASS
  - NOLOG is specified on /*JOBPARM

JESLOG Spin Data Sets

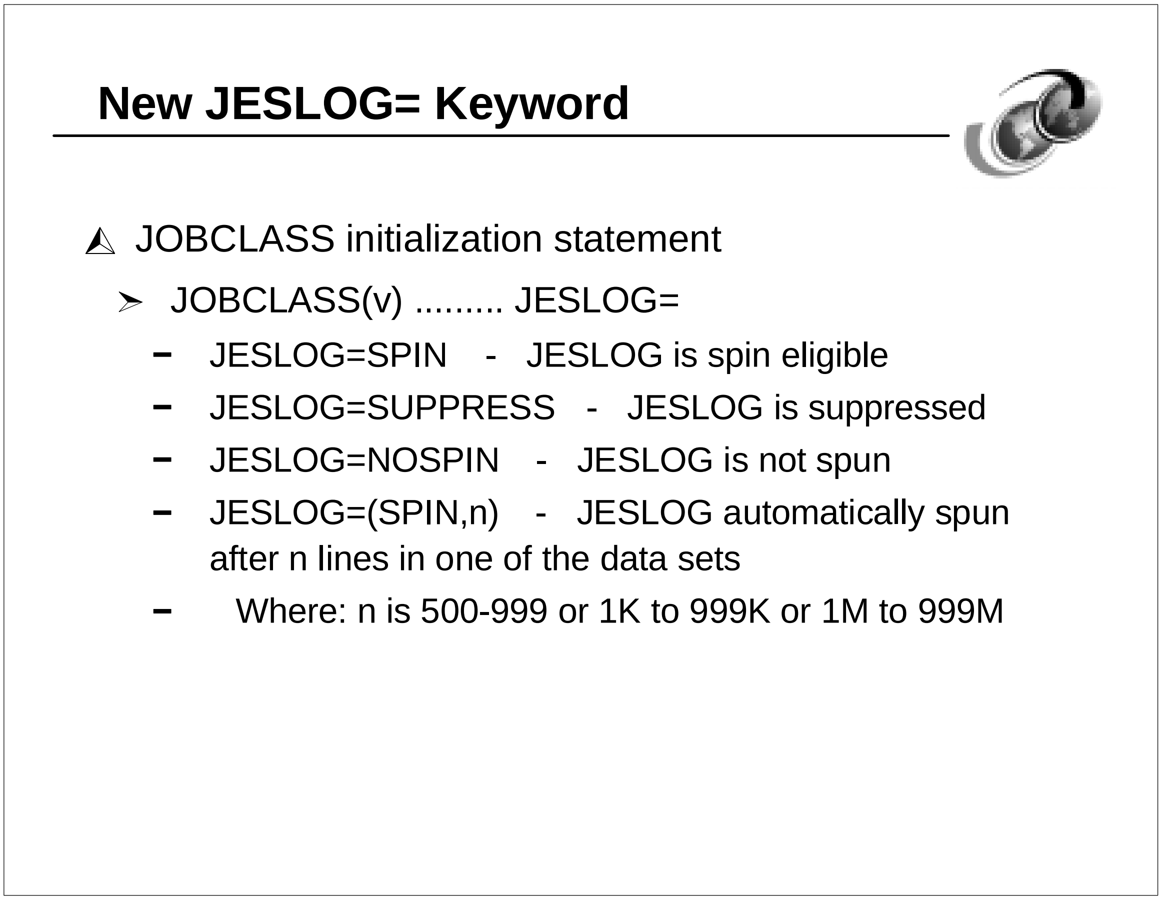


- Spin data sets only have messages created during execution
- Non-Spin data sets have messages from input, conversion, and job stats
- Can only suppress messages in SPIN data sets



Controlling JESLOG Data

- JESLOG can be suppressed
- JESLOG can be spun automatically based on line count or time interval or time of day
- JESLOG can be spin "eligible"
  - Operator command issued any time during execution
- JESLOG data sets are both spun
  - When either data set satisfies the criteria
  - When an operator command is entered



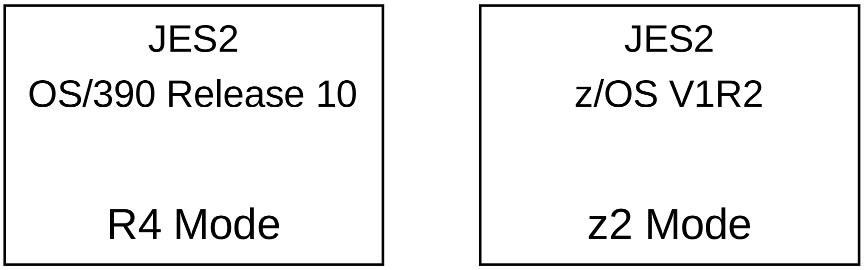
New JESLOG= Keyword

- JOBCLASS initialization statement
  - JOBCLASS(v) .... JESLOG=
    - JESLOG=SPIN - JESLOG is spin eligible
    - JESLOG=SUPPRESS - JESLOG is suppressed
    - JESLOG=NOSPIN - JESLOG is not spun
    - JESLOG=(SPIN,n) - JESLOG automatically spun after n lines in one of the data sets
    - Where: n is 500-999 or 1K to 999K or 1M to 999M

New JESLOG= Keyword

- JCL job statement
  - JESLOG= same syntax as for JOBCLASS
    - Except if time specified
    - Enclose time of day or elapsed specification in apostrophes (single quotes accepted)
  - //JOBNAME JOB ... ,JESLOG=(SPIN,'+5:13')

Mixed MAS Support



1) J123 executes here   J123 converted here

Does not perform JESLOG spin

2) J901 converted here

Can never perform JESLOG spin

Dynamic PROCLIB Support

- **Problem:** PROCLIBs defined in the JES2 start proc require a JES2 restart to change
  - Change may require ALL MAS members to be restarted
  - Error in JES2 PROC may prevent restart

Dynamic PROCLIB Support

- **Solution:** Allow dynamic allocation of PROCLIBs
  - **PROCLIB(***xxxx***)** initialization statement
  - **$ADD PROCLIB(***xxxxxxxx***)** command
  - **$T PROCLIB(***xxxxxxxx***)** command
  - **$DEL PROCLIB(***xxxxxxxx***)** command
  - **$D PROCLIB(***xxxxxxxx***)** command

Dynamic PROCLIB Support

- New PROCLIB initialization statement

```
PROCLIB(xxxxxxxx)  DD(n)=(DSNAME=dsn,
                          VOLSER=volser,
                          UNIT=unit),
                          UNCONDITIONAL
```

  - Up to 255 DDs per PROCLIB
  - VOLSER and UNIT are optional (if cataloged)
  - UNCONDITIONAL - create even if allocations fail

Defining PROCLIBs

- Old way (Static PROCLIB)
  - In JES2 PROC:

```
//PROC01  DD   DSN=USER.PROCLIB1,VOL=SER=J2PROC,UNIT=3390
//        DD   DSN=USER.PROCLIB2,VOL=SER=J2PROC,UNIT=3390
//        DD   DSN=SYS1.PROCLIB
```

- New way (Dynamic PROCLIB)
  - In JES2 initialization stream:

```
PROCLIB(PROC01) DD(1)=(DSN=USER.PROCLIB1,VOLSER=J2PROC,UNIT=3390
                DD(2)=(DSN=USER.PROCLIB2,VOLSER=J2PROC,UNIT=3390
                DD(3)=(DSN=SYS1.PROCLIB)
```

$DPROCLIB Command

- $D PROCLIB(PROC01)

```
$HASP319 PROCLIB(PROC01)
$HASP319 PROCLIB(PROC01)  DD(1)=(DSNAME=USER.PROCLIB1,
$HASP319                  VOLSER=J2COM1,UNIT=3390),
$HASP319                  DD(2)=(DSNAME=USER.PROCLIB2,
$HASP319                  VOLSER=J2COM1,UNIT=3390),
$HASP319                  DD(3)=(SYS1.PROCLIB)
```

- $D PROCLIB(PROC01),DEBUG

```
$HASP319 PROCLIB(PROC01)
$HASP319 PROCLIB(PROC01)  USECOUNT=0,DDNAME=SYS00006,
$HASP319                  CREATED=2001.149,20:42:22.36,
$HASP319                  DD(1)=(DSNAME=USER.PROCLIB1,
$HASP319                  VOLSER=J2COM1,UNIT=3390),
$HASP319                  DD(2)=(DSNAME=USER.PROCLIB2,
$HASP319                  VOLSER=J2COM1,UNIT=3390),
$HASP319                  DD(3)=(SYS1.PROCLIB)
```

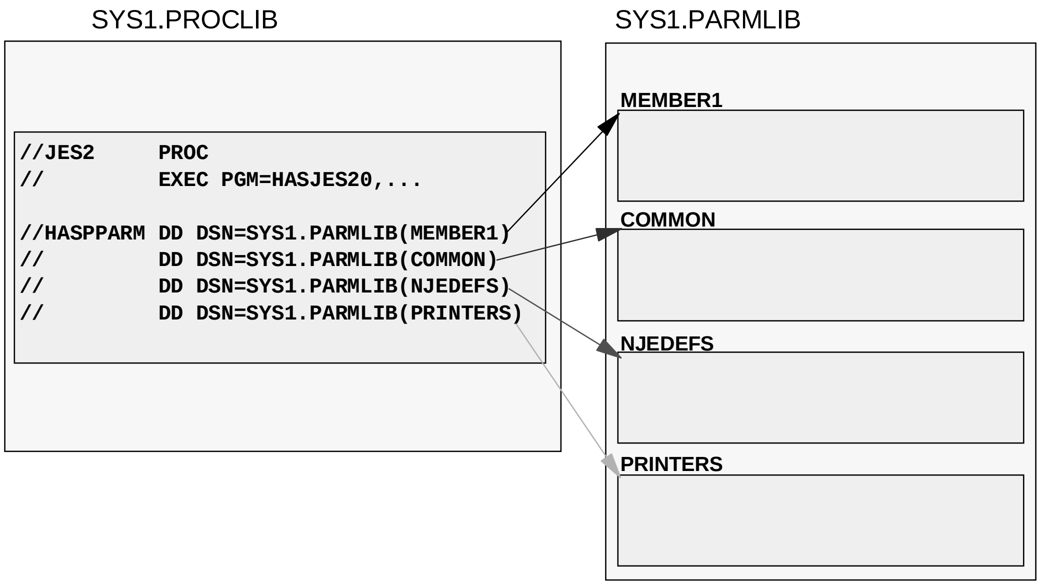
PROCLIB Considerations

- To change concatenation for dynamic PROC01
  - Method 1:
    - **$T PROCLIB(PROC01),DD(1)=...,DD(2)=...**
    - Could require several commands due to command length limitations
    - Advantage: Simplest way if few datasets in concatenation
  - Method 2:
    - **$ADD PROCLIB(TEMP01),DD(1)=...**
    - **$T PROCLIB(TEMP01),DD(2)=...**
    - Test and update TEMP01 as required
    - **$T PROCLIB(TEMP01),NAME=PROC01**
    - Advantage: ATOMIC, Allows testing!

INCLUDE Initialization Statement

- Concatenation of JES2 initialization members requires an update of the JES2 procedure
  - If update is incorrect - JES2 does not start
- New INCLUDE statement
  - Includes other members
  - Reduces need to update procedure
  - Can use symbolics with member to include

Old Method for Initialization

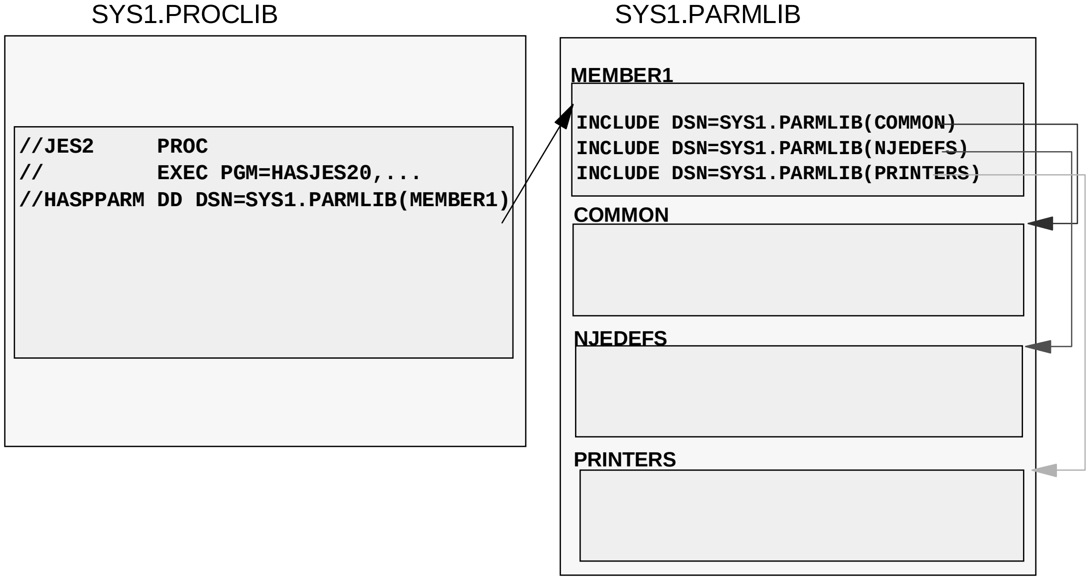


```
//JES2     PROC
//         EXEC PGM=HASJES20,...

//HASPPARM DD DSN=SYS1.PARMLIB(MEMBER1)
//         DD DSN=SYS1.PARMLIB(COMMON)
//         DD DSN=SYS1.PARMLIB(NJEDEFS)
//         DD DSN=SYS1.PARMLIB(PRINTERS)
```

New Way using INCLUDE Statements

Simplify JES2 procedure



SYS1.PROCLIB

```
//JES2     PROC
//         EXEC PGM=HASJES20,...
//HASPPARM DD DSN=SYS1.PARMLIB(MEMBER1)
```

SYS1.PARMLIB

MEMBER1

```
INCLUDE DSN=SYS1.PARMLIB(COMMON)
INCLUDE DSN=SYS1.PARMLIB(NJEDEFS)
INCLUDE DSN=SYS1.PARMLIB(PRINTERS)
```

COMMON

NJEDEFS

PRINTERS

Emergency Start of JES2

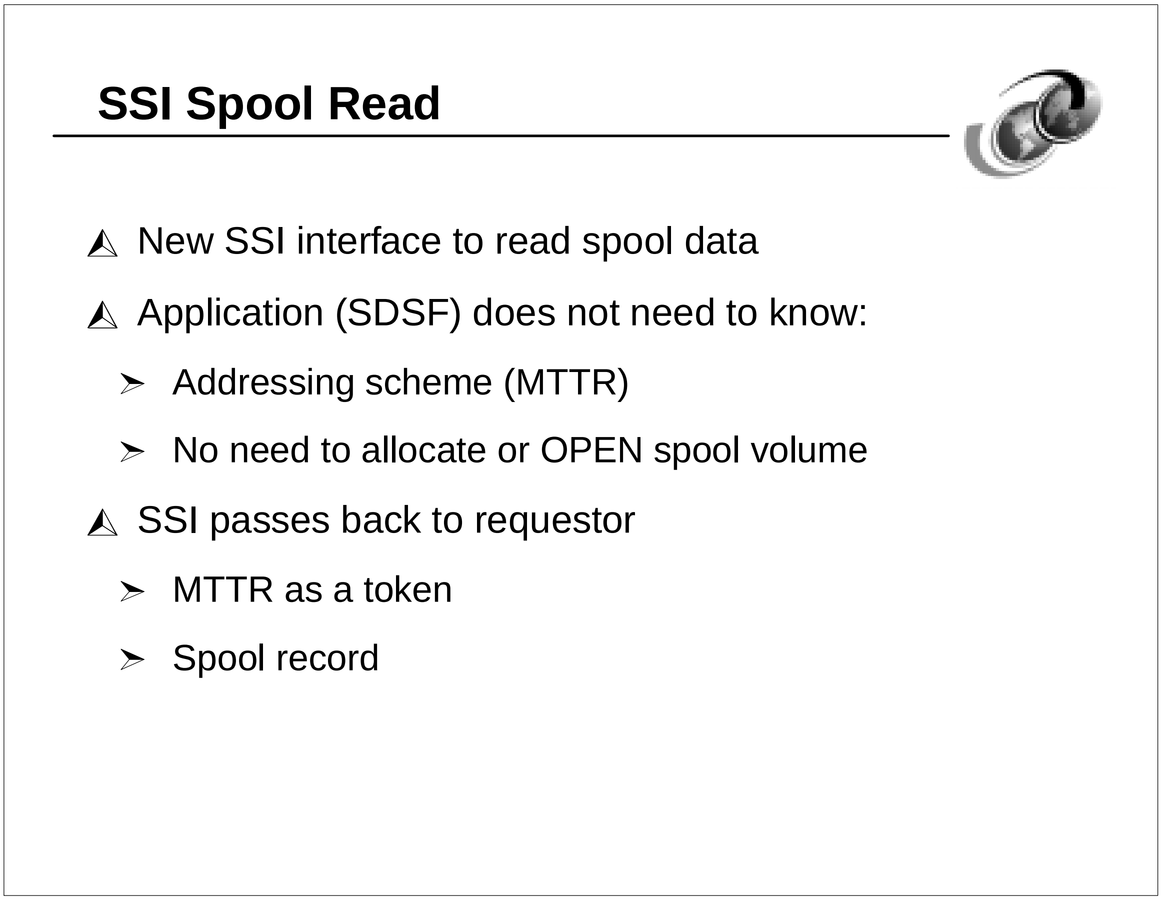
- Simplify JES2 PROC
  - EXEC, STEPLIB DD, one HASPPARM DD
  - Define PROCLIBS via PROCLIB statement
  - INCLUDE additional DD statements
- In emergency, ***start JES2 without a PROC!***
  - S IEESYSAS,PROG=HASJES20,JOBNAME=JES2
  - Assumes **HASJES20** in LINKLIST (no STEPLIB)
  - When **HASPPARM** allocation fails, reply to **$HASP469** message with
    - **INCLUDE** statement(s) for correct init deck(s)
    - **PROCLIB** statements (if not in init decks)

Large Spool Support

- Support for up to 64K track per data set
  - Anywhere on the volume
  - Uses relative tracks in the MTTR
- New spool read SSI
- New device id conversion SSI

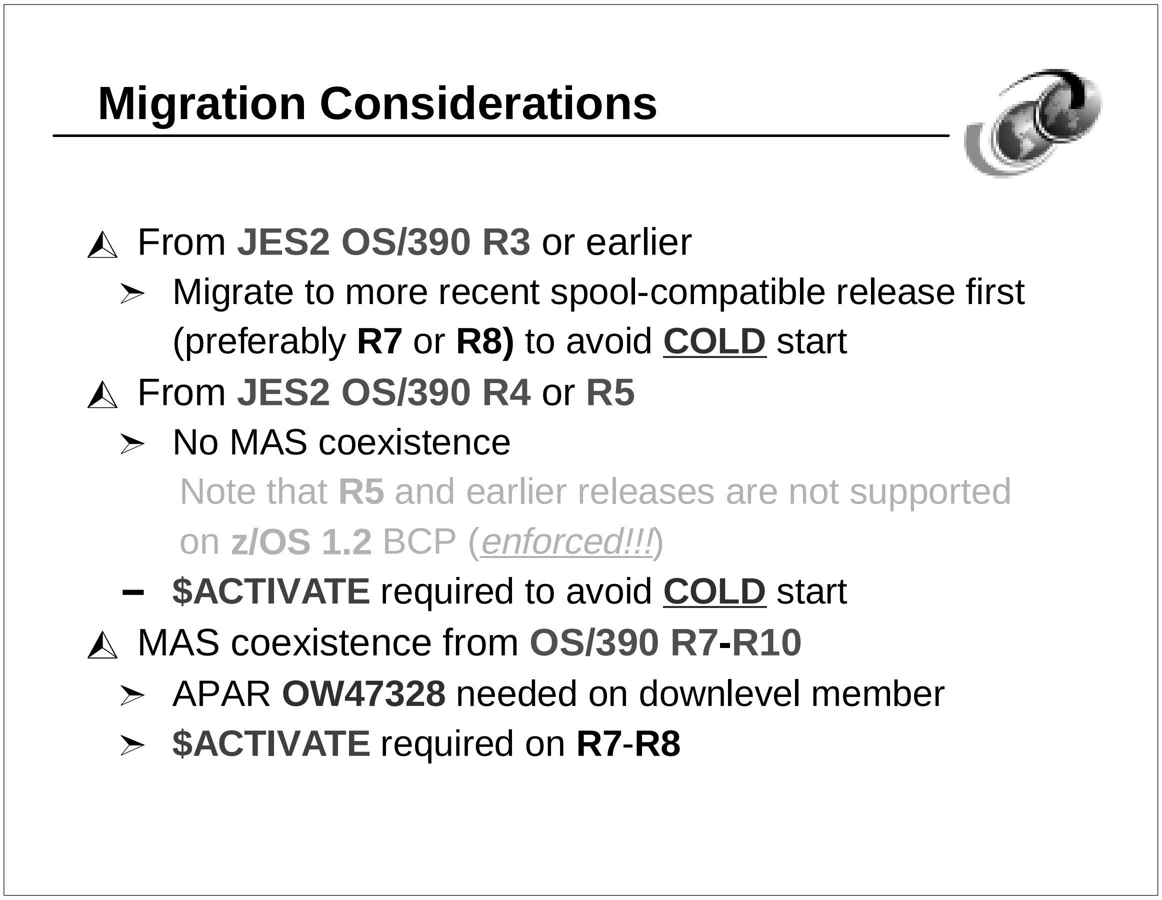
SPOOLDEF Initialization Statement

- SPOOLDEF RELADDR=NEVER|ALWAYS|ASNEEDED
  - NEVER - never use relative addressing
  - ALWAYS - use relative addressing for all volumes
  - ASNEEDED - use relative addessing if 64K is crossed
- Applies when volume is starting
- All members must support relative addressing
  - Support rolled down to R10 (end of 2001)



SSI Spool Read

- New SSI interface to read spool data
- Application (SDSF) does not need to know:
  - Addressing scheme (MTTR)
  - No need to allocate or OPEN spool volume
- SSI passes back to requestor
  - MTTR as a token
  - Spool record



Migration Considerations

- From **JES2 OS/390 R3** or earlier
  - Migrate to more recent spool-compatible release first (preferably **R7** or **R8)** to avoid **COLD** start
- From **JES2 OS/390 R4** or **R5**
  - No MAS coexistence

    Note that **R5** and earlier releases are not supported on **z/OS 1.2** BCP (*enforced!!!*)
  - **$ACTIVATE** required to avoid **COLD** start
- MAS coexistence from **OS/390 R7-R10**
  - APAR **OW47328** needed on downlevel member
  - **$ACTIVATE** required on **R7**-**R8**

Installing JES2 V1R2

- New **$ACTIVATE** level (z2 mode)
  - Needed for some >64K jobs functions
  - Can switch from R4 to z2 mode or z2 to R4 mode via operator command
    - **$ACTIVATE,LEVEL=Z2**
    - **$ACTIVATE,LEVEL=R4**
    - **LEVEL=** is a required keyword
  - Can also switch from z2 to R4 mode via PARM=UNACT start option
  - **$D ACTIVATE** displays current $ACTIVATE level