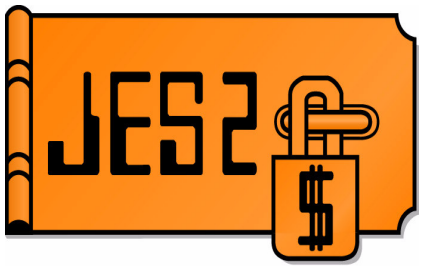
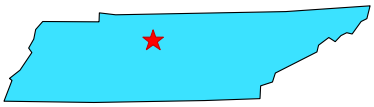


JES2 Product Update

SHARE 98, Session 2655



Monday, March 4, 2002

Chip Wood
JES2 Design/Development/Service
Poughkeepsie, NY

chipwood@us.ibm.com

JES2 z/OS 1.2

- Greater than 64K jobs support
  - **Session 2656, Wed. 11:00**
- Dynamic PROCLIB support
- INCLUDE initialization statement
- Long running jobs JESLOG support
- Large spool volume support
- Miscellaneous enhancements

JES2 z/OS 1.2 installation

- From **JES2 OS/390 R3** or earlier
  - Migrate to more recent spool-compatible release first (preferably **R8)** to avoid **COLD** start
- From **JES2 OS/390 R4** or **R5**
  - ➡ Note that **R5** and earlier releases are not supported on **z/OS 1.2** BCP (*enforced!!!*)
  - **$ACTIVATE** required to avoid **COLD** start
  - No MAS coexistence (all-member-warm start)
- MAS coexistence from **OS/390 R7-R10**
  - APAR **OW47328** needed on downlevel member
  - **$ACTIVATE** required on **R7**-**R8**

z/OS 1.2 installation

- 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
- IBM recommends running in z2 mode for a while before implementing new limits

>64K jobs support

- Internal limits increased
  - **JOBDEF JOBNUM** up to **200,000**
  - **JOBDEF RANGE** up to **999,999**
  - **OUTDEF JOENUM** up to **500,000**
  - **CKPTSPACE BERTNUM** up to **500,000**
  - **SPOOLDEF TGSPACE=MAX** up to **16,580,355**
- JOBID changes
  - JOB*nnnnn* or J*nnnnnnn*
- 4-byte job number fields
- JQEs/JOEs now chained by index (not offset)
- Widespread internal changes
- **Session 2656, Wed. 11:00**

Dynamic PROCLIB

- **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
  - SHARE requirement *SS-JES2-98.203*
- **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

PROCLIB statement

- 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
- New operator commands
  - **$ADD PROCLIB(*xxxxxxxx*)**
  - **$T PROCLIB(*xxxxxxxx*)**
  - **$DEL PROCLIB(*xxxxxxxx*)**
  - **$D PROCLIB(*xxxxxxxx*)**

PROCLIB example

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

```
//PROC01  DD   DSN=USER.PROCLIB1,VOL=SER=J2COM1,UNIT=3390
//        DD   DSN=USER.PROCLIB2,VOL=SER=J2COM1,UNIT=3390
//        DD   DSN=SYS1.PROCLIB
```

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

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

  - Modify using $T PROCLIB(PROC01)

Modifying dynamic proclibs

- 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!

Modifying static proclibs

- To change concatenation for static PROC01
  - Method 1:
    - **$ADD PROCLIB(PROC01),DD(1)=...,DD(2)=...**
    - Dynamic definition overrides static definition
    - **$T PROCLIB(PROC01)** to update as necessary
    - **$DEL PROCLIB(PROC01)** to revert to static definition
  - Method 2:
    - **$ADD PROCLIB(TEMP01),DD(1)=...**
    - **$T PROCLIB(TEMP01),DD(2)=...**
    - Test and update TEMP01 as required
    - **$T PROCLIB(TEMP01),NAME=PROC01**

$D PROCLIB example

- **$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)
```

  - May also display old concatenations with non-zero use counts, if $T command has been issued

INCLUDE statement

- **Problem:** Changing JES2 init deck concatenation requires changing JES2 PROC
  - If update is incorrect, JES2 will not start
  - May be difficult to fix when JES2 is down
- **Solution:** New INCLUDE initialization statement
  - Reduces need to update JES2 PROC

INCLUDE statement

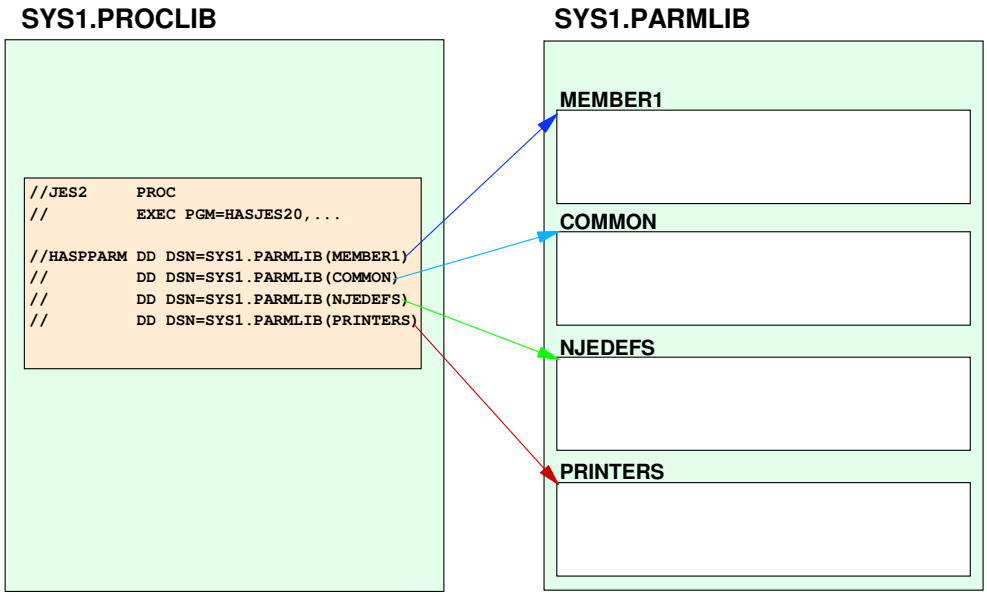
- New **INCLUDE** initialization statement

```
INCLUDE   DSNAME=dsn,
          VOLSER=volser,
          UNIT=unit
```

  - DSNAME=*dsn* may include a member name
  - VOLSER and UNIT are optional (if cataloged)
- **D INCLUDE**
  - Displays current INCLUDEd data set
  - If not in INCLUDE, displays current data set in HASPPARM concatenation
  - Useful to locate source of error when unexpectedly placed in CONSOLE mode due to error in initialization deck

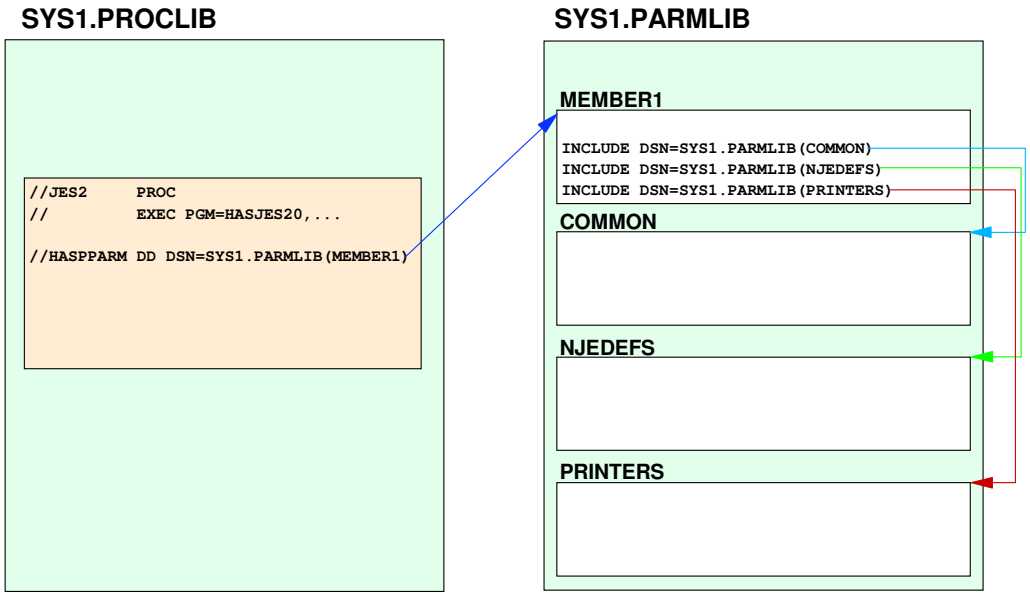
Include statement example

- Old way:



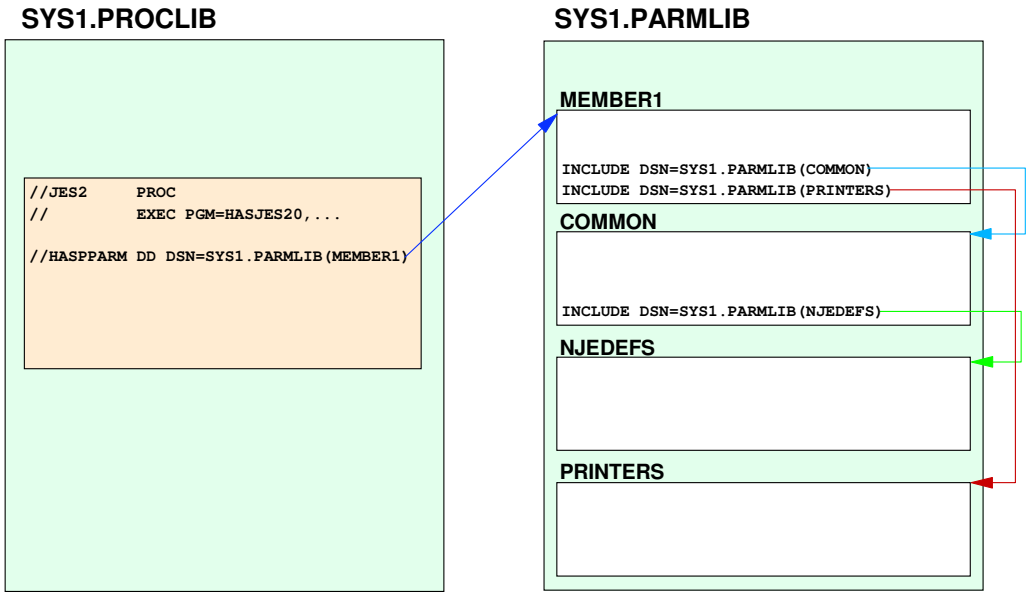
Include statement example

- New way:



Include statement example

- New way:



Combining INCLUDE and PROCLIB

- Simplify JES2 PROC
  - EXEC, one HASPPARM DD
  - Define PROCLIBS via PROCLIB statement
  - INCLUDE additional HASPPARM datasets
- In emergency, ***start JES2 without a PROC!***

```
S IEESYSAS,PROG=HASJES20,JOBNAME=JES2
```

  - Assumes **HASJES20** in LINKLIST (no STEPLIB)
  - When **HASPPARM** open fails, reply to **$HASP469** message with
    - **INCLUDE** statement(s) for correct init deck(s)
    - **PROCLIB** statements (if not in init decks)

Long running jobs JESLOG support

- **Problem:** **JESLOG** (JES2 joblog and system messages) data sets use spool space which cannot be freed until the job ends execution
  - For some jobs, this implies an IPL
  - Space is used even if a dummy MSGCLASS is specified
  - SHARE requirement *SO-JES3-89.360*
- **Solution:** Allow the **JESLOG** datasets to be spin data sets

JESLOG Externals

- **JESLOG=** parameter added in several places to specify what is to be done with JESLOG data sets:
  - JCL
    - **// JOB ,JESLOG=(...)**
  - JES2 initialization statements
    - **JOBCLASS(x) JESLOG=(...)**
    - **REQJOBID JESLOG=(...)**
  - JES2 commands
    - **$T JOBCLASS(x),JESLOG=(...)**
    - **$T REQJOBID,JESLOG=(...)**

JESLOG values

- JESLOG= may specify any of the following:
  - **JESLOG=SUPPRESS** - no JESLOG produced
  - **JESLOG=SPIN** - JESLOG is spin eligible
  - **JESLOG=NOSPIN** - JESLOG is not spin eligible
  - **JESLOG=(SPIN,*n*)** - JESLOG automatically spun after n lines in either data set
    - *n* is 500-999, 1**K**-999**K** or 1**M**-999**M**
  - **JESLOG=(SPIN,*hh*:*mm*)** - JESLOG spun at time of day
    - Time is 0:00 to 23:59
  - **JESLOG=(SPIN,+*hh*:*mm*)** - JESLOG spun at interval
    - Interval is +0:10 to +99:59

JESLOG values

- Special rules for hh:mm (time) and +hh:mm (interval)
  - JESLOG is not spun until the first message after the given time or interval
  - JCL rules require the time or interval to be enclosed in quotes when specified on the job card
    - **//JOBNAME JOB ,JESLOG=(SPIN,'+2:00')**
- **$T JOB,SPIN** command
  - If JESLOG is spin eligible, immediately spins data sets regardless of any limits or intervals that were specified

$HASP138 message

- Appended to each spun JESLOG data set (both of them) when data sets are spun
- Includes reason for spin
  - Operator command

```
$HASP138 JESLOG SPIN REQUESTED BY OPERATOR COMMAND
```

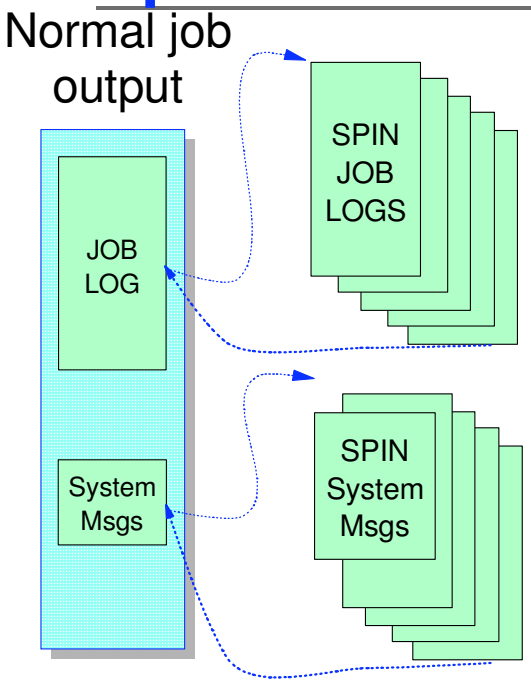
  - Line count

```
$HASP138 JESLOG SPIN HAS OCCURRED BECAUSE OF LINE COUNT
```

  - Time/Interval

```
$HASP138 JESLOG SPIN HAS OCCURRED BECAUSE OF TIME INTERVAL
         OR TIME OF DAY
```

Spool data structures



- Spin data sets only have messages from execution
- Non-Spin data sets have messages from input, conversion, and job stats
- Can only suppress messages in SPIN data sets
- Both data sets spin simultaneously when either reaches limit

Browsing JESLOG data sets

- Spool browse supports special data set names as of **OS/390 R10** JES2
- To access base (non-spin portion), use data set name:
  - *user.jobname.jobid*.D0000002.JESMSGLG
- To access individual spin datasets, use data set name:
  - *user.jobname.jobid*.D*nnnnnnn*.JESMSGLG
- To access concatenated job log (**z/OS 1.2** only), use data set name:
  - *user.jobname.jobid*.JESMSGLG
  - Only works if all data sets are available

Large spool volume support

- **Problem:** JES2 SPOOL volume processing limits data sets to tracks below 64K
  - Spool addresses are MTTR where TT is absolute track address (from start of volume)
  - Volumes with >64K tracks (**3390-9**) can only be used if the spool data set is forced within the first 64K tracks
- **Solution:** Allow relative track address (from start of data set) in MTTR
  - Spool data set can be up to 64K tracks anywhere on volume
  - SSI call to read spool buffers
  - Also in OS/390 R10 via APAR **OW49317**

Large volume support externals

- **SPOOLDEF RELADDR=**
  - **NEVER** - Fail start if data set crosses 64K absolute track boundary
  - **ALWAYS** - Always uses relative track addresses
  - **ASNEEDED** - Uses relative track addresses only if the dataset crosses 64K absolute track boundary
- Applies only when volume is starting
  - Must be supported on all active members at the time
- All members must support relative addressing before it will be used
  - Down-level system re-entering MAS will fail because volume start/end tracks in $DAS do not match
    - **$HASP401 EXTENT ERROR ON *volser***

SPOOL Read SSI

- New SSI interface to read SPOOL data
  - Application need not know addressing scheme
  - No need to allocate or open SPOOL volume
  - Pass MTTR as a token and get SPOOL record
- Interface is new function on SSI 71
  - Interface CB is **IAZSPLIO**
  - IAZSSJI updated with new function codes
- Can read either data records or signature records

SPOOL Read SSI

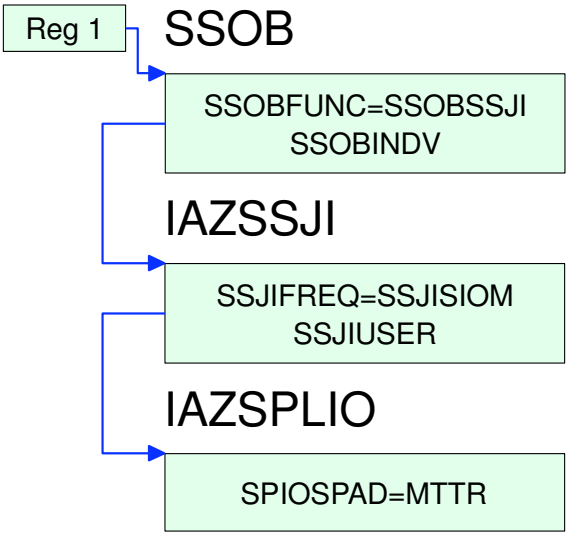
- **IAZSPLIO** data area input fields
  - **SPIOSPAD** - 8 byte spool address to read (MTTR in 1st 4 bytes)
  - **SPIOCTYP** - CB type to read (must be one of CHK, HDB, IOT, JCT, NHSB, OCT, SIG, or SWBI)
  - **SPIOJNAM** - Job name to verify
  - **SPIOJID** - Job id to verify (only the number is verified)
  - **SPIOJKEY** - Job key to verify
  - **SPIODSKY** - For CBs that support them, data set key to verify

SPOOL Read SSI

- **IAZSPLIO** data area output fields
  - **SPIOOUTA** - address of output buffer
  - **SPIOOLEN** - length of output buffer
- Output fields only set if return code is zero
- **SPIOSTRP** is a storage token that should be zeroed on first call and then not modified

SPOOL Read SSI

- SSI data areas 1st call:

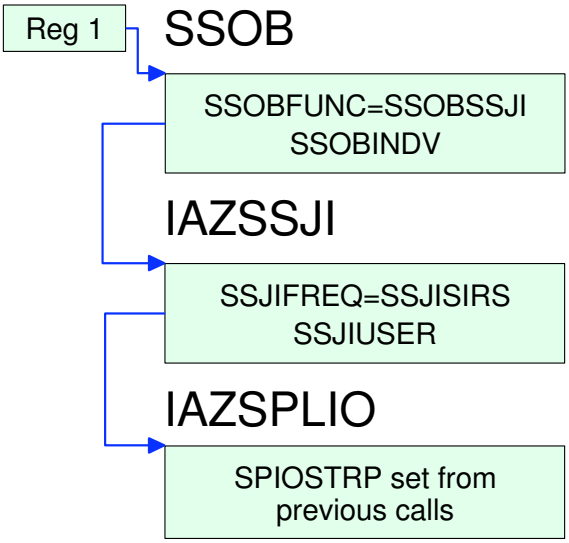


Register 1 points to SSOB
SSOBINDV points to SSJI
SSJIUSER points to IAZSPLIO

First call passes in SSJIFREQ set to SSJISIOM. First 4 bytes of SPIOSPAD is set to the MTTR to read. Any additional verification is set in IAZSPLIO. The call returns a return code, buffer address (points to spool data no prefix) and length

SPOOL Read SSI

- SSI data areas, final call:



Register 1 points to SSOB
SSOBINDV points to SSJI
SSJIUSER points to IAZSPLIO

Final call passes SSJIFREQ set to SSJISIRS and SPIOSTRP set from previous call. This requests that any storage obtained to be returned.

Convert Device Id SSI

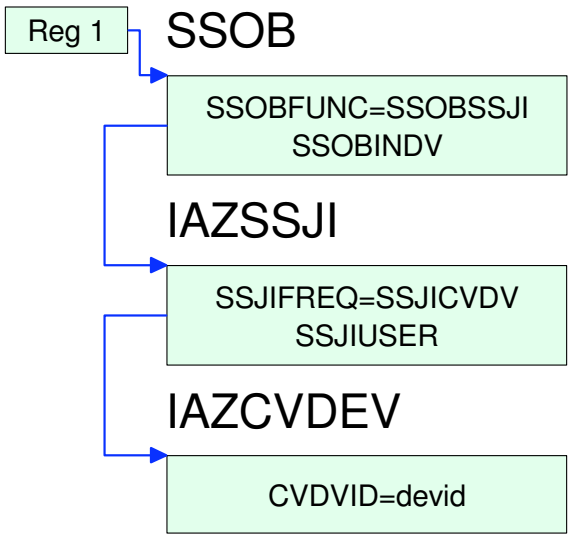
- New SSI interface to convert device ids from binary to EBCDIC
  - 3-byte Binary device ids appear in JES2 control blocks (JOEs, JQEs, DCTs)
  - EBCDIC device name may be needed for message
- Interface is new function on SSI 71
  - Interface CB is **IAZCVDEV**
  - IAZSSJI updated with new function codes
- Device ids are now assigned to lines and logons

Convert Device Id SSI

- **IAZCVDEV** data area input fields
  - **CVDVID** - 3 byte binary device id
- **IAZCVDEV** data area output fields
  - **CVDVNAME** - 18 byte EBCDIC device name

Convert Device Id SSI

- SSI data areas:



Register 1 points to SSOB

SSOBINDV points to SSJI

SSJIUSER points to IAZCVDEV

CVDVID is set to the 3-byte devid from the JQE, JOE, or DCT.

On output, CVDVNAME contains the full device name

Dynamic NJE subdevices

- **NJEDEF LINENUM=** value can now be defined greater than the number of lines in the init deck
  - Allows number of transmitter/receiver "packets" to be overdefined in case of future $ADD LINE command
- **LINE*nnn* JTNUM**,**JRNUM**,**STNUM**, and **SRNUM** values can now be modified via command
  - **$ADD LINEnnn,JTNUM=n,JRNUM=,...**
    - A dedicated "packet" can be created when the line is added
  - **$T LINEnnn,JTNUM=n,JRNUM=n,...**
    - Dedicated "packets" can be created or removed
    - The number of any subdevice type can be changed

$DJQ,SPOOL

- **Problem:** TG count in the JQE is a halfword
  - Any job using more than 32K track groups looks the same
    - **TGS=***** on **$HASP890**
    - **PERCENT=**.****** on **$HASP890**
  - **$DJQ,SPOOL=TGS** filter or **SPOOL=PERCENT** filter give strange results
- **Solution:** Claim a third byte in the JQX
  - Requires **$ACTIVATE,LEVEL=Z2**
  - Composite 3-byte field in JQA: **JQPTGNBR**
  - Maximum TG count is now **8,388,607**
- **Session 2656, Wed. 11:00**

Termination changes

- Automatic reply to termination WTORs
- After 5 minutes, highlighted "nag" message issued every 30 seconds

```
$HASP065 AWAITING RESPONSE TO $HASPxxx MESSAGE, AUTO-REPLY
         IN nnn SECONDS
```

- After 10 minutes, default action taken based on message
  - **$HASP070** - "RECOVER" - recovers and takes a dump
  - **$HASP089** - "N" - does not retry failed dump
  - **$HASP098** - "END,DUMP" - new action, terminates with dump

Termination changes

- "Ended" PCEs may be detached under some circumstances
  - Remote subdevices - Ended PCE detached when remote is disconnected
  - NJE Line subdevices - Ended PCE detached when line is disconnected
- New $HASP068 text for NJE/RJE devices:

```
$HASP068 PARTIAL RECOVERY SUCCESSFUL - R1.PR1 PCE ENDED
         JES2 CONTINUES WITHOUT RJE PRINTER
         2 OF 3 RMT.PRT PCES REMAIN ON REMOTE1
```

- Allows remote or NJE line to be restarted after subdevice abend
  - $HASP068 message deleted

$D PCE

- Improved filtering on **$D PCE** command
  - **DETAILS** subparameter
    - Can filter on DETAILS level keywords to see individual PCEs matching criteria, rather than just types
    - **NAME** - PCE name
    - **CURJOB**/**JOBID** - job active on PCE
    - **ACTIVE** - active count for PCE
    - **I/O** - I/O count for PCE
    - **ENDED** - whether PCE is ended or not
    - **EXIT** - which exit the PCE is $WAITing in
    - **TRACE** - whether PCE is tracing or not
  - APAR **OW52163** fixes base $SCAN problem

$D PCE examples

- Without DETAILS filter

```
          $dpce,active>0
$HASP653 PCE(PRT)    COUNT=(12,12,0),ACTIVE=2,TRACE=NO
$HASP653 PCE(MLLM)   COUNT=(1,1,0),ACTIVE=10,TRACE=NO
```

- With DETAILS filter (new)

```
          $dpce,details=(active>0)
          $HASP653 PCE(PRT)
$HASP653 PCE(PRT)    NAME=PRT3,WAIT=FSS,XECB,INHIBIT=NO,
$HASP653             MOD=HASPFSSP,SEQ=03367000,TIME=2001.166,
$HASP653             19:44:33,ACTIVE=1,I/O=0,
$HASP653             NAME=PRT2,WAIT=IO,MOD=HASPPRPU,
$HASP653             SEQ=39285000,TIME=2001.166,19:42:04,
$HASP653             CURJOB=STC00002,ACTIVE=1,I/O=0
          $HASP653 PCE(MLLM)
$HASP653 PCE(MLLM)   NAME=MLLM,WAIT=WORK,MOD=HASPRTAM,
$HASP653             SEQ=03970000,TIME=2001.166,19:50:46,
$HASP653             ACTIVE=10,I/O=0
```

PERFDATA

- **$D PERFDATA(CKPTSTAT)**
  - Displays specific information about CKPT I/O

```
          $dperfdata(ckptstat)
          $HASP660 $DPERFDATA(CKPTSTAT)
$HASP660 CKPT PERFORMANCE STATISTICS - INTERVAL=13:31.725791,
$HASP660 AVGHOLD=0.425011,AVGDORM=4.920611,TOT$CKPT=54540,
$HASP660 WRITE-4K=17,WRITE-CB=1606,OPT$CKPT=33444,OPT4K=0,
$HASP660 IO=R1,COUNT=147,AVGTIME=0.014144,
$HASP660 IO=R2,COUNT=0,AVGTIME=0.000000,TOTAL4K=0,TOTALCB=93,
$HASP660 IO=PW,COUNT=147,AVGTIME=0.007197,TOTAL4K=82,TOTALCB=0,
$HASP660 IO=IW,COUNT=151,AVGTIME=0.006798,TOTAL4K=0,TOTALCB=573,
$HASP660 IO=FW,COUNT=148,AVGTIME=0.006470,TOTAL4K=17,TOTALCB=1033
```

$T CKPTDEF,MODE

- **$T CKPTDEF,MODE=DUAL/DUPLEX**
  - Allows checkpoint mode to be changed without an all-member warm start
  - Simplifies migration to (and from) checkpoint on CF
    - **MODE=DUPLEX** required for CF checkpoint
    - **MODE=DUAL** most efficient for DASD
  - Code in compatibility APAR (**OW47328**) allows down level member to recognize change
    - **OW52689** fixes 0C4 ABEND in this processing

APPLCOPY

- Support for application copy of checkpoint (**APPLCOPY**) has been discontinued
  - Impractical for new large checkpoint sizes
  - **$HASP003** message if **CKPTDEF APPLCOPY=PRIVATE** or **APPLCOPY=COMMON** is specified
  - Use checkpoint versions (SSI 71) instead

HASX05C

- JES2 sample command translation exit **HASX05C** has been moved to SHASSAMP
  - Not automatically loaded or enabled
  - ➡ **Functionality of exit has not changed**
  - Will continue to ship in SHASSAMP for the foreseeable future
- See DOC APAR OW51031 for more information

JES2 z/OS 1.2

- Greater than 64K jobs support
  - **Session 2656, Wed. 11:00**
- Dynamic PROCLIB support
- INCLUDE initialization statement
- Long running jobs JESLOG support
- Large spool volume support
- Miscellaneous enhancements

➡ APPLCOPY has been deleted

➡ HASX05C is now shipped as a sample exit