



JES2 z/OS 1.7: Large Spool, NJE over TCP/IP, and Other Features

SHARE 105, Session 2657
Tuesday, August 23, 2005

Permission is granted to SHARE Inc. to publish
this presentation in the SHARE proceedings. IBM
retains its right to distribute copies of this
presentation to whomever it chooses.

Chip Wood / Tom Wasik
JES2 Design/Development/Service
Poughkeepsie, NY
chipwood@us.ibm.com
wasik@us.ibm.com

JES2 z/OS 1.7



- **New changes in z/OS 1.7**
 - NJE over TCP/IP
 - Large (>64K track) SPOOL Data Sets
 - Reader/NJE exit changes
 - Long SYSIN support (32K LRECL)
 - Table pair enhancements
 - \$SCAN from non-main task environments
 - SSI for JES2 monitor information
 - SAPI and extended status enhancements
 - Checkpoint recovery (DAS corruption)
 - Other goodies (requirements!!!!)
 - Drop of support for release 4 mode

JES2 z/OS 1.7 is the largest and most function-rich release of the product in many years. This session will provide an overview of the function added in the release. Many of the functions will be discussed in much more detail in sessions throughout the week.

Large (>64K track) SPOOL Data Sets



- **JES2 z/OS 1.7 supports SPOOL data set of up to 1,048,575 (was 64K)**
- **Change the format of a SPOOL address**
 - Pre-z7: MTTR (4 bytes)
 - z7: MTTtr (still 4 bytes)
 - ◆ 8 bits of M, 20 bits of T, and 4 bits of R
- **Implies some changes to format of allocation IOT**
 - TGAE goes from MTT (3 bytes) to MTTTT (5 bytes)
 - MTTTT also referred to as MQT in code (Q="quad")
- **Other data areas also affected**
 - \$QSE and \$HCT sniffer work fields
 - DAS TGM offset and size fields, track limits
- **Cannot be used if >15 records per track**
 - buffer sizes less than 2943 on a 3390
 - **Action: increase spool buffer size**
 - ◆ Requires **COLD** start (so schedule well in advance!)

The problem that is being addressed by this line item is the current limit on the size of a SPOOL data set. Before JES2 could address this limit, DFSMS

needed to write support code that allowed a non-extended (non-SMS managed) data set that was greater than 64K tracks. Once that was done, JES2 needed support code to be able to address that much space in a single data set.

JES2 uses 4 byte MTTRs to address records on SPOOL. Using this scheme, we can address up to 64K tracks with 255 records per track. But JES2 formats the tracks with much less than 255 records per track. On a 3390 with the recommended buffer size of 3992 bytes, JES2 used 12 records per track. This implies that we can use some of the bits from the "R" value to supplement the TT value. By borrowing 4 bits, we can get 20 bits or 1M tracks. If the buffer size is too small, such that there are more than 15 records per track, this scheme cannot be used. That is considered a permanent restriction.

The SPOOL read SSI makes these changes transparent to any application that does not look at the contents of the MTTR.

Large (>64K track) SPOOL Data Sets (cont...)



- **New external, SPOOLDEF LARGEDS=FAIL|ALLOWED|ALWAYS**
 - Applies when a volume is started
 - FAIL causes start to fail if data set is >64K tracks
 - ALLOWED will permit the start and use new MTTtr if >64K tracks
 - ALWAYS will permit all starts and always use new MTTtr
 - ♦ ALWAYS intended for testing
- **\$T to switch to any values (FAIL to ALLOWED or vice versa)**
- **Once LARGEDS=ALLOWED or ALWAYS you CANNOT start z5 or lower**
 - SPOOL data areas incompatible with older releases
 - Restricted even if LARGEDS is set back to FAIL

- **JES2 will no longer support SPOOLDEF RELADDR=**
 - In particular, RELADDR=NEVER is no longer supported
- **All new volumes will use relative track addressing**
- **Will continue to support absolute track addressing for volumes started on previous releases**

Because of the nature of the changes, down level releases cannot support these new larger SPOOL volumes (in part because DFSMS on the down level MVS do not support the larger data sets). As a result, a new external was needed to “activate” the support. The LARGEDS= parameter was added to SPOOLDEF for this line item. LARGEDS has 3 values, FAIL, ALLOWED and ALWAYS. FAIL does not allow any large data set to be used. It also allows down level JES2 member to co-exist with this level of JES2. ALLOWED will activate the LARGEDS support (update checkpoint and change the format for new SPOOL control blocks) and allow large data set to be started. ALWAYS is similar to ALLOWED except you do not need a large volume to test the new format for MTTTR. As such, ALWAYS is intended as a testing tool.

Once LARGEDS is set to ALWAYS or ALLOWED, down level JES2 members can NEVER again enter the MAS (even if LARGEDS is set to FAIL).

Commands and Init Statements



▪ Support activated by SPOOLDEF LARGEDS=

– LARGEDS=FAIL

- ◆ All volumes must be less than 64K tracks
- ◆ Compatible with older releases of JES2
- ◆ MTTtrs and MQTs never used

– LARGEDS=ALLOWED

- ◆ IOT, TGAEs and other data areas converted to MQTs
- ◆ Large volumes can be started
- ◆ MTTtrs used for large volumes only
- ◆ Older release can NEVER be warm started again

– LARGEDS=ALWAYS

- ◆ Same as ALLOWED but MTTtrs are used for all new volumes

Commands and Init Statements



- **\$T command and init statement is MAS scope**
- **Init statement honored on Cold starts**
 - Warm starts use previous value
- **Setting LARGEDS=ALLOWED or LARGEDS=ALWAYS like mini-activate**
 - Checkpoint expanded for larger data areas
- **Once set to ALLOWED or ALWAYS, cannot warm start down level release even if \$T'ed back to NEVER.**
 - Down level JES2 cannot deal with SPOOL data area changes
 - Must offload and cold start to allow down levels to join MAS
- **LARGEDS=ALWAYS intended for testing MTTtrs even if volume is not large**

The LARGEDS parameter has MAS scope and is honored in the init deck on a COLD start and can be modified via a \$T command. Once LARGEDS is set to ALWAYS or ALLOWED, JES2 starts creating SPOOL control blocks that down level members do not know how to process. Because of this, once LARGEDS is set to ALWAYS or ALLOWED, down level JES2s cannot ever join this MAS (unless a cold start is done).

Commands and Init Statements (Cont...)



- **\$D SPL command shows if MTTtrs being used**

- SPOOL2 is using MTTtrs (notice TRKRANGE)

```

$dspool(spool2),unitdata
$HASP893 VOLUME(SPOOL2)
$HASP893 VOLUME(SPOOL2)  UNITDATA=(EXTENT=01,TRKRANGE=(0001,
$HASP893                  003C),BASETRAK=0000000E,RECMAX=12,
$HASP893                  TRKPERCYL=15)
    
```

- SPOOL3 is using MTTtrs (notice TRKRANGE)

```

$dspool(spool3),unitdata
$HASP893 VOLUME(SPOOL3)
$HASP893 VOLUME(SPOOL3)  UNITDATA=(EXTENT=02,TRKRANGE=(00000001,
$HASP893                  0000003C),BASETRAK=0000000E,RECMAX=12,
$HASP893                  TRKPERCYL=15)
    
```

There is no direct external to determine whether a spool volume was started using the new format. However, there is an indirect way. The **TRKRANGE=** parameter on the **\$DSPL,UNITDATA** command will display as a two byte value (4 hex digits) if using the old format, and as a four byte value (8 hex digits) if using the new format.

Getting to Large SPOOL volumes



1. On a test system, test applications that access SPOOL by setting **LARGEDS=ALWAYS** and starting a SPOOL volume
2. Migrate to z/OS 1.7 JES2 on all MAS members
3. Wait for z/OS 1.7 JES2 to stabilize (no need to fall back)
4. **\$T SPOOLDEF** to **LARGEDS=ALLOWED**
5. Stabilize with the new format of data areas
6. Start a large SPOOL volume
 - Consider using SPOOL affinity to limit jobs using new SPOOL
7. Once stabilized, drain old SPOOL volumes and migrate to new larger SPOOL data sets
 - Clear SPOOL affinities if used for testing earlier

REMEMBER, once LARGEDS=ALLOWED is set, pre-z/OS 1.7 JES2s cannot be warm started even if set to LARGEDS=NEVER

This is the preferred migration path to large SPOOL data sets. It minimizes the risk to the system and provides a reasonable backout plan.

Exit/application Implications



- **Some applications may attempt to read/write SPOOL records**
- **Reading is supported using the SPOOL READ SSI**
 - Applications need to treat 4 byte SPOOL address as a token
 - Should not examine individual fields
- **Direct writing to SPOOL is not supported**
- **Applications that convert MTTR to BBCCHHR will need to be updated to support new MTTtr SPOOL address format**
 - Example of conversion can be found in routine \$EXCP in HASPNUC

- **Action:** use SPOOL read SSI to access SPOOL data directly

- **Fields in the \$DAS, \$IOT, and \$TGB (BLOB) are updated to 5 or 6 byte MQTs or MQTRs**
 - Examine data areas for any changes

If there are exits or applications that read or write the spool directly, they may be impacted by the changes in this support. In particular, an application that converts the MTTR to BBCCHHR for use in CCWs needs to be updated. Code in HASPNUC can help you convert any code that you may have that does this.

The SPOOL read SSI makes these changes transparent to any application that does not look at the contents of the MTTR.

Migration/Coexistence Considerations



- **z/OS 1.7 compatibility APAR is OA08145**
 - Supported on z/OS 1.4 (HJE7707) – z/OS 1.5 (HJE7708)
- **Support deleted for R4 checkpoint mode**
 - Must \$ACTIVATE,LEVEL=z2 prior to migration
- **Support deleted for SPOOLDEF RELADDR=**
 - All volumes started by z/OS 1.7 will use relative track addressing
 - Volumes using absolute track addressing still supported
 - ◆ No plans to drop support for absolute track addressing
- **Once LARGEDS support active pre-z/OS 1.7 JES2s will fail**

**\$HASP720 WARM START DENIED - LARGE SPOOL DATA SET FORMAT HAS BEEN
ACTIVATED IN THE MAS. THIS RELEASE DOES NOT SUPPORT LARGE SPOOL DATA
SET FORMAT.**

NJE over TCP/IP



- **Currently JES2 supports NJE over SNA and BSC networks**
- **TCP/IP is the standard for networking today**
- **VM (RSCS), AS/400, and VSE/POWER all have supported NJE over TCP/IP for years**
- **Requirements for JES2 support from many sources**
 - SHARE
 - zBLC
 - Customer Requests
- **JES2 z/OS 1.7 supports established TCP/NJE protocol**
 - Modified BSC CTC protocol
 - Enabled by APAR **OA12364**, available **1Q2006**
- **Enhancements to protocol also included for**
 - IPv6 connections
 - Enhanced security
 - Large LRECL SYSIN data streams
- **Overall better RAS characteristic have been designed in**
 - Fewer outages than current NJE
 - Better performance

Support for NJE over native TCP/IP has been one of the most requested JES2 enhancements over the last several years.

NJE over TCP/IP (Cont...)



- **Protocol to be used is that originally developed for VM**

- Assigned port is 175, port name is VMNET (non-SSL)
- Assigned port is 2252, port name is NJENET-SSL (SSL/TLS)
- Originally described as BITNET II protocol
- Documentation is in NJE formats and protocols

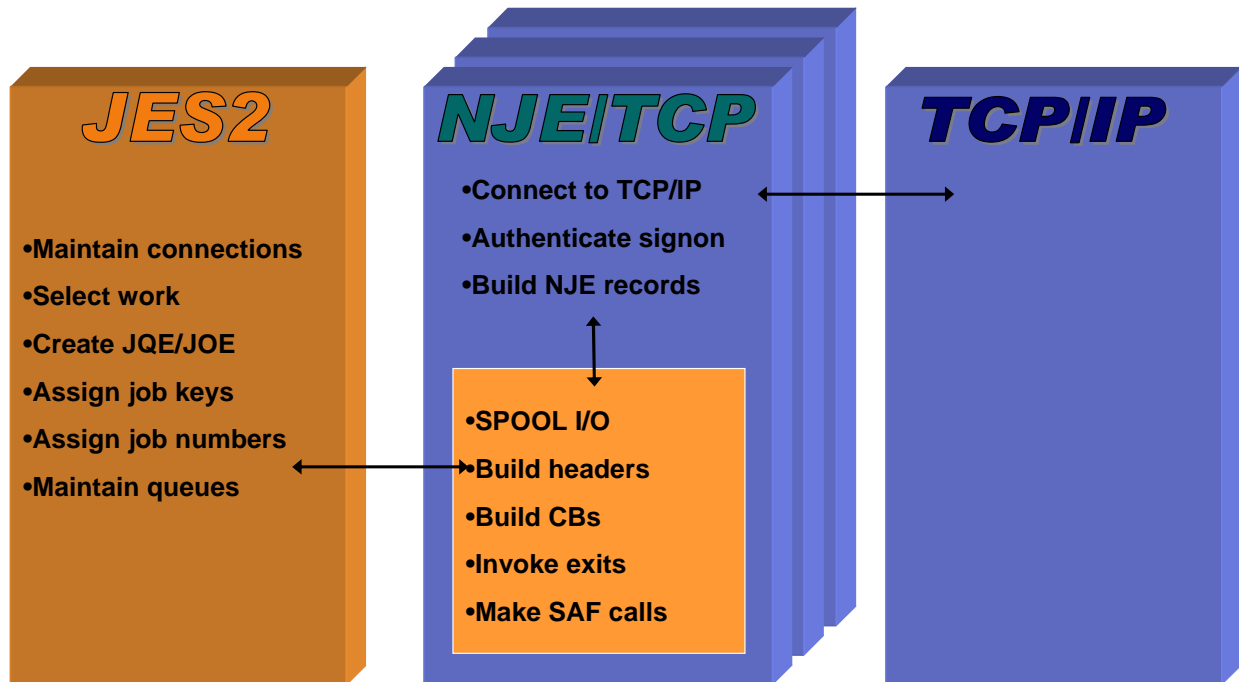
http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/has2a600/12.0

- Essentially BSC CTCA protocol with TCP/IP wrappers
 - ♦ Support for IPv6
 - ♦ Support for SSL/TLS
- Binary IP addresses will be optional in data records
 - ♦ Better support for NAT
- Stronger authentication will also be available (using SAF/RACF)
 - ♦ Controlled by **NODEnnn SIGNON=SECURE|COMPAT**

JES2 implemented the established NJE over TCP/IP protocol defined by VM (RSCS). However, since the VM implementation, a number of developments in TCP/IP required some updates to the existing protocol. These are in the area of IPv6 (larger IP addresses) and enhanced security. These required changes to the basic sign-on protocols. In addition, there is a desire to support SYSIN data streams that have an LRECL of up to 32K. This will also be added to the protocol.

As you will see, this is a redesign of how JES2 does NJE (for TCP/IP). The goal of the redesign is improved RAS and better performance (both for JES2 and for the NJE data transfers).

NJE over TCP/IP (Cont...)



One of the primary concerns customers have had with JES2 address space outages is that when the JES2 address space ABENDs, all NJE connections are lost. Even though the JES2 address space can be restarted, the NJE connections must also be reinitialized. Many customers have automation to do this, but only when the system is IPLed. So, the JES2 address space outage turns into a system outage.

In order to address this, the NJE/TCP support is moved to a separate address space, where the connection can remain active when the JES2 address space is unavailable. This approach has multiple benefits:

Availability – Outages of the JES2 address space do not affect the availability of the NJE connection. Conversely, problems in the NJE address space do not result in a JES2 outage as they would have in the past.

Performance – Most of the work associated with the NJE connection (I/O, building headers and trailers, etc.) is being done outside the JES2 address space, rather than under the JES2 main task. This frees up cycles in the already overtaxed JES2 main task to do other things.

The actual communication with TCP/IP is being done via a new common JES2/JES3 component, IAZNJTCP.

NJE over TCP/IP (Cont...)



- Within JES2, the externals will parallel what was defined for SNA

SNA	NJE/TCP	Description
LOGON	NETSERV	Represents a NJE/TCP address space
APPL	SOCKET	Maps an NJE node name to a TCP/IP address (either explicit or a name)
LINE	LINE	Logical connection

- Commands to start and stop networking/devices similar to SNA
 - \$SLINE, \$SNETSRV, \$SN, \$ELINE, \$PLINE, \$ENETSRV, \$PNETSRV, etc.
- NETSERVs can bind to all defined IP addresses or a specific address

The externals for JES2 NJE over TCP/IP are modeled after the SNA externals, with a NETSRV corresponding to a LOGON and a SOCKET corresponding to an APPL. Logical lines specifying UNIT=TCP are used.

NJE over TCP/IP Externals



- **NETSRV** statement

```
NETSRV(nnn) SOCKET=, STACK=,  
TRACEIO=( JES=NO, COMMON=NO, VERBOSE=NO)
```

- **SOCKET=** - points to SOCKET(xxxxxxxx) statement defining the local node's IP address and port
- **STACK=** - indicates a specific TCP/IP stack to use. Default is all stacks.
- **START=** - indicates whether NETSRV should start automatically
- **TRACE =**
 - ♦ **JES=YES/NO** – traces communication within JES2 (between JES2 and NETSRV address space), and from JES2 to the common code
 - ♦ **COMMON=YES/NO** – traces events and communication within the common component (such as TCP/IP API calls)
 - ♦ **VERBOSE=YES/NO** – issues additional messages to console/SYSLOG for diagnostics on live system
 - ♦ **TRACE=YES/NO** – affects all 3 settings

The NETSRV statement (which may also be specified as NETSERV or NSV) defines the characteristics of the network server.

NJE over TCP/IP Externals



▪ SOCKET statement

```
SOCKET(WSC)          IPADDR=9.117.2.233,PORT=175,
                     SECURE=NO,LINE=0,NODE=2,REST=0,
                     NETSRV=0
```

- **STATUS=** - the status of the socket (ACTIVE/INACTIVE) – display only
- **IPADDR=** - the IP address associated with the socket definition
 - ◆ Can specify host name (chipwood.pok.ibm.com), IPv4 or IPv6 address
 - ◆ *LOCAL indicates all IP addresses associated with this system
- **PORT=** - the port associated with the socket definition
 - ◆ Default is 175 if SECURE=NO, 2252 if SECURE=YES
- **SECURE=YES/NO** – whether to use SSL/TLS
- **NODE=** - the associated node
- **LINE=** - a dedicated line
- **NETSRV=** - the NETSRV to use when establishing a connection to this IP address/port
- **REST=** - the resistance associated with the socket
- **SOCKID=** - The socket id assigned by IAZNJTCP (display only)

The SOCKET statement defines the IP address, port, and other attributes associated with one end of a TCP/IP connection. These definitions are used in 2 places:

The SOCKET= parameter on the NETSRV statement identifies the IP address and port associated with the local node (i.e. this end of the connection)

The \$SN,S= command identifies the socket representing the IP address and port we wish to connect to (i.e. the other end of the connection)

NJE over TCP/IP Externals



- **LINE** statement
 - Maximum line number increased from 32767 to 65535
 - **UNIT=TCP** – defines line as TCP/IP line
 - **TRACE =**
 - ◆ **JES=YES/NO** – traces communication within JES2 (between JES2 and NETSRV address space), and from JES2 to the common code
 - ◆ **COMMON=YES/NO** – traces events and communication within the common component (such as TCP/IP API calls)
 - ◆ **VERBOSE=YES/NO** – issues additional messages to console/SYSLOG for diagnostics on live system

The line statement represents a logical line used for networking. Lines used for NJE over TCP work like lines used for NJE over SNA. The major difference is the **UNIT=** parameter is set to TCP.

NJE over TCP/IP Externals



■ Commands

- **\$ADD NETSRVnnn** – creates a new NETSRV device (1-999)
- **\$T NETSRVnnn** – change attributes of NETSRV
- **\$S NETSRVnnn** – Starts NETSRV address space
 - ◆ Address space name is *jesxSnnn*
 - ◆ **NETSRVnnn SOCKET=** - defines IP addr/port of this node
- **\$P NETSRVnnn** – Drains NETSRV address space
- **\$E NETSRVnnn** – Resets connections in NETSRV address space
- **\$D NETSRVnnn** – Display attributes of NETSRV
 - ◆ **ASID** – displays the ASID of the NETSRV, if active
 - ◆ **NAME** – displays address space name
 - ◆ **SESSIONS** – displays list of all SOCKETs connected to this NETSRV

NJE over TCP/IP externals

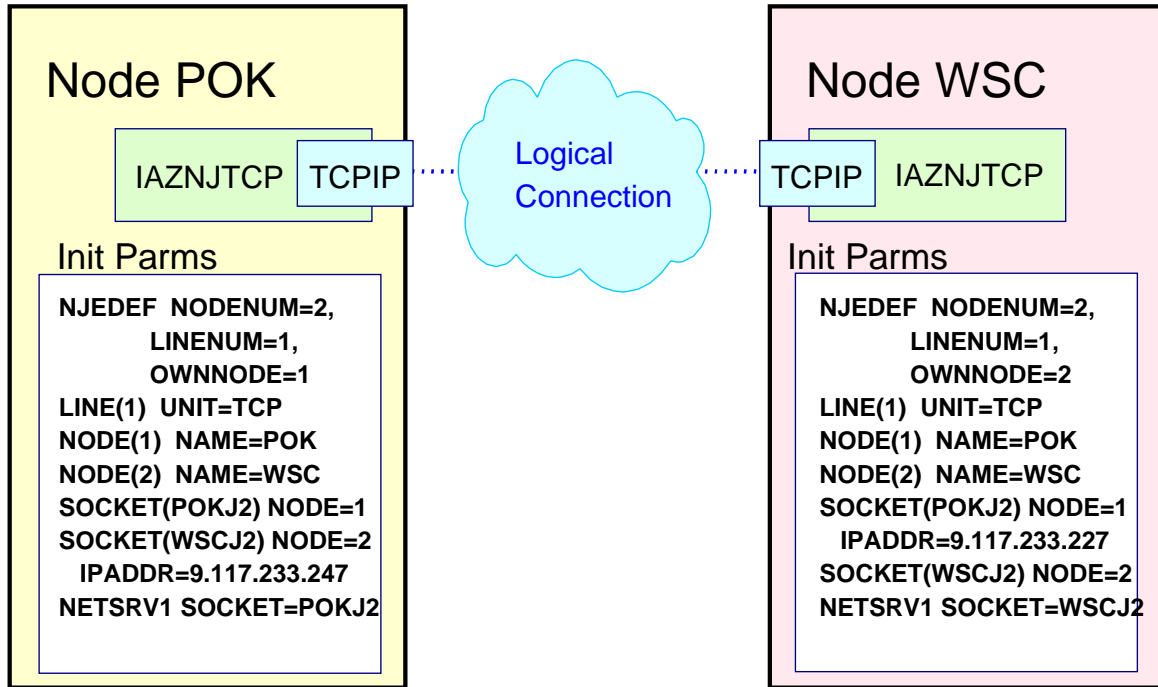


▪ More Commands

- **\$ADD SOCKET(*socket*)** – Creates a new socket definition
- **\$T SOCKET(*socket*)** – modify attributes of socket
- **\$D SOCKET(*socket*)** – display attributes of socket

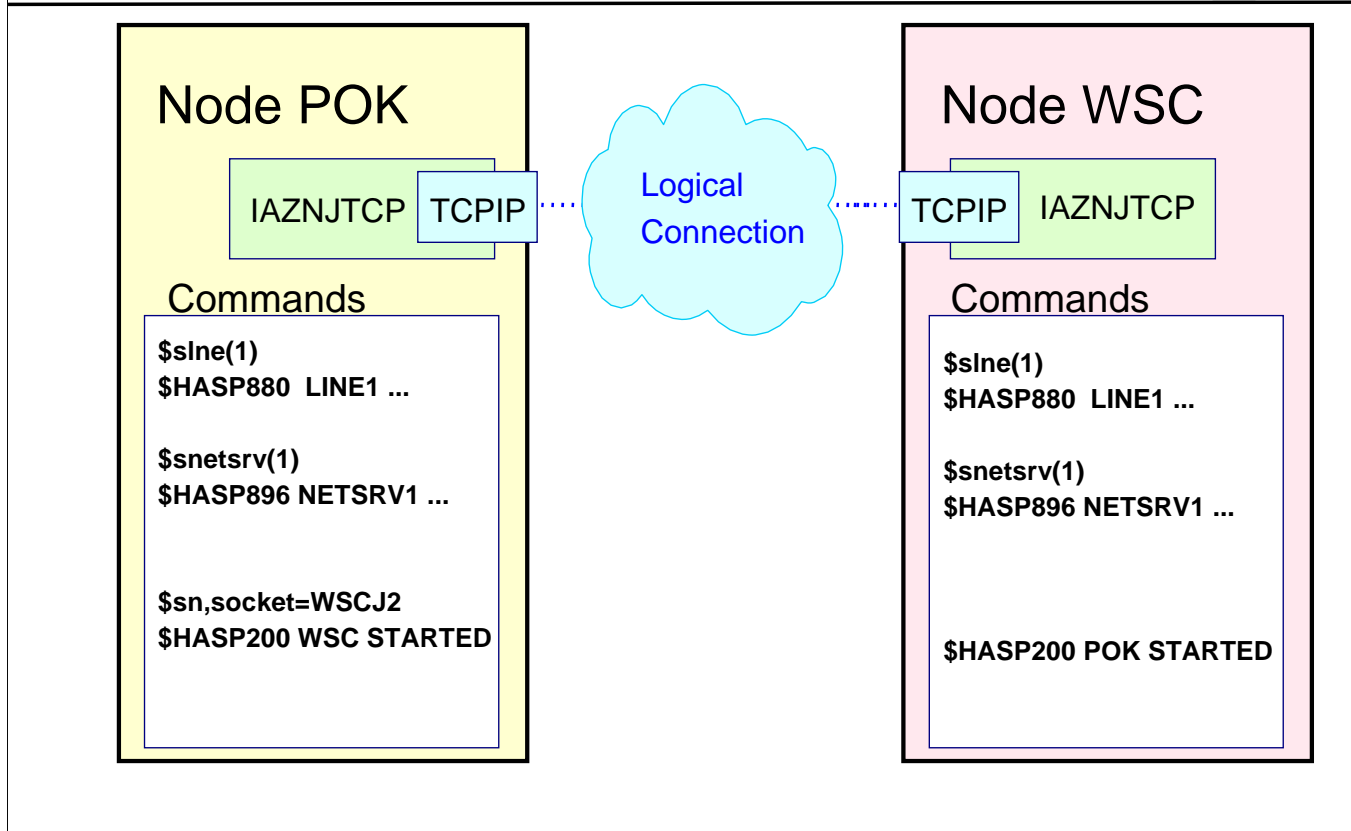
- **\$SN,S=*socket*** – Start networking with specified socket
 - ♦ **\$SN,N=*nodename*** – if **NODE(*nodename*)** **LINE=** points to TCP/IP line

JES2 NJE Initialization Statements (TCP/IP)



Here is a simple two node setup which will connect the two nodes using TCP/IP. Node POK is at IP address 9.117.233.227 and Node WSC is at address 9.117.233.247. Since PORT= was not specified on any of the SOCKET statements, the default NJE/TCP port (175) will be used.

JES2 NJE Commands (TCP/IP)



Here are the operator commands that are required to start the connection.

The NETSRV address space



- **`$$ NETSRVnnn`**
 - Results in ASCRE for network server address space
 - Address space name is **`jesxSnnn`**
 - ♦ **`jesx`** – subsystem name
 - ♦ **`nnn`** – NETSRV number
 - ♦ Example – starting NETSRV1 under subsystem JES2 results in address space name of JES2S001
 - Runs SUB=MSTR
 - IAZNJTCP code drives JES routines to process records, build headers, etc.
 - ♦ Control blocks and routine addresses anchored from IAZYTCT control block

The NETSRV address space



- **\$P NETSRVx**
 - Queues request to NETSRV address space to shut down cleanly
- **\$E NETSRVx**
 - Queues request to NETSRV address space to terminate all connections
- **NETSRV address spaces may be cancelled or forced if unresponsive to commands**
 - Not preferred technique, but available if necessary
 - **P JES2S001**
 - **CANCEL JES2S001**
 - **FORCE JES2S001**

Controlling NJE lines



- **The same controls exist for controlling TCP/IP lines as have existed for SNA and BSC lines in prior releases**
 - **`$$ LINE(nnn)`**
 - **`$P LINE(nnn)`**
 - **`$E LINE(nnn)`**
 - **`$$ Lnnn.JTx, $$ Lnnn.STx, $S Lnnn.JRx, $S Lnnn.SRx`**
 - **`$P Lnnn.JTx, $P Lnnn.STx, $P Lnnn.JRx, $P Lnnn.SRx`**
 - **`$E Lnnn.JTx, $E Lnnn.STx, $E Lnnn.JRx, $E Lnnn.SRx`**
 - **`$CLnnn.JTx, $C Lnnn.STx, $C Lnnn.JRx, $C Lnnn.SRx`**
 - **`NODEnnn TRANSMIT=, RECEIVE=, HOLD=`**
 - **`LINEnnn JTNUM=, STNUM=, JRNUM=, SRNUM=`**
 - etc.

Operationally, all of the traditional methods for controlling NJE lines, transmitters, and receivers exist for TCP/IP lines as well as for SNA and BSC. This means operators (and system programmers!) do not need to learn a whole new suite of commands and init statements.

Additional NJE Commands



- **\$DLINE(*nnnn*),NODES**
 - Displays list of all nodes reachable via the line
- **\$DL*nnn*.JT*n*,NODES**
 - Displays list of all nodes reachable via the job transmitter
- **\$DL*nnn*.ST*n*,NODES**
 - Displays list of all nodes reachable via the SYSOUT transmitter

- **\$DLINE(*),NODE=*name***
 - Displays all lines by which node is reachable
- **\$DL(*).JT(*),NODE=*name***
 - Displays all job transmitters by which node is reachable
- **\$DL(*).ST(*),NODE=*name***
 - Displays all SYSOUT transmitters by which node is reachable

The NODEs subparameter is added to the LINE, L.JT, and L.ST display commands. This can be used in conjunction with the \$DNODE, \$DPATH, and \$D CONNECT parameters to determine information about paths to a node, or to determine specifically why a particular transmitter is not selecting work for a specific node. The NODES parameter must be requested specifically to display as many configurations would result in a very large display.

Additional NJE-related parameters



- **NODEnnnn DIRECT=YES/NO**
 - Indicates whether paths considered to this node must be directly adjacent connections
 - ◆ **DIRECT=YES** – only adjacent paths will be considered for this node
 - ◆ **DIRECT=NO** (default) – all paths will be considered for this node
 - MAS-wide scope
 - May be changed via **\$T NODE** command

- **\$D NODEnnnn,ADJACENT**
 - Displays whether an adjacent connection to this node currently exists
 - Can be used as a filter to display all adjacent nodes

- **\$D NODEnnnn,OWNNODE**
 - Displays whether this is the local node
 - Can be used as a filter to readily display the local node definitions
 - ◆ **\$D NODE,OWNNODE=YES**

NJE NODE Definition Changes



- **NJE node definitions are now MAS scope**
 - Changes on one member reflected on others
 - Data is stored in the JES2 checkpoint
 - Applies to NAME=, SUBNET=, PATHMGR=, ENDNODE=, PRIVATE=, and DIRECT=
 - Init deck changes accepted when other members active
 - ♦ Warning message if incompatible, option to continue or terminate
- **Change node names when node is active (no restart)**
 - **\$TNODE(*nodename*),NAME=*newname***
 - ♦ Indicates node is renamed, not redefined
 - Applies to local node name as well (was all-member warm start)
- **Change maximum node number via command**
 - **\$T NJEDEF,NODENUM=** (was all member warm start)
 - Value can only be increased

This support required that NJE definitions persist over a restart of JES2. To ensure this, the NJE NODE statements are now stored in the JES2 checkpoint. This also results in the node definitions having a MAS scope. Changes on one member will be automatically propagated to all members of the MAS. This applies only to the listed operands on the NODE statement. Since many customers use init deck changes and restarts to update their network definitions, a warm start will cause the global node definitions to be updated. If the new definitions are not compatible with the current network, a warning message is issued and a WTOR is issued to ask the operator if we should continue.

You can change the name of the local node as well as other active nodes in the network using the \$TNODE command. The \$TNODE command subscript always could have been either the node number or the node name. In the past, it did not matter which form you used, both worked the same. However, when modifying the name of an active node (one that is reachable in the network or the local node) the 2 forms behave differently. If you specify the node number, JES2 will perform the standard tests to see if the node is active before allowing a node name change. However, if the node name is specified, the active checks are bypassed and the name can be changed.

The maximum number of nodes can now also be modified by an operator command (increase only).

NJE Security Changes



- **Support for SSL/TLS to be included in NJE over TCP/IP**
 - Application Transparent TLS support in Z/OS 1.7 (AT-TLS) used
 - ◆All definitions for SSL/TLS are in TCP/IP policy definitions, not JES definitions
 - ◆Future standards for TLS will automatically be supported
 - Only JES2 control is **SECURE=YES/NO** on **SOCKET** statement

- **Secure form of NJE signon now supported**
 - Exchanges DES-encrypted passwords in I/J signon records
 - Controlled by **SIGNON=SECURE|COMPAT** on **NODE** statement
 - Uses **APPCLU** class in RACF/SAF
 - ◆Entity is **NJE.node1.node2**
 - ◆Uses **SESSKEY** associated with profile for encryption
 - Can be used by SNA or BSC nodes as well as TCP/IP

- **NODE(n) PASSWORD=** is not used when **SIGNON=SECURE**

In the TCP/IP world, security becomes much more important than in the more protected environments of SNA and BSC. For NJE/TCP connections, JES2 exploits the application Transparent TLS support introduced in z/OS 1.7 by TCP/IP. The setup for this is controlled totally by TCP/IP and SAF/RACF. The only JES2 specification is the **SECURE=YES/NO** on the **SOCKET** statement.

TLS and SSL go a long way, but additional validation may be required. TLS and SSL, for example, help ensure that the partner on the other end of the connection is who he claims to be in a TCP/IP sense. However, we still need to validate that he is who he says he is in an NJE connectivity sense. Also, if TLS/SSL is not available, you may not want to send nodal passwords into TCP/IP-land in clear text.

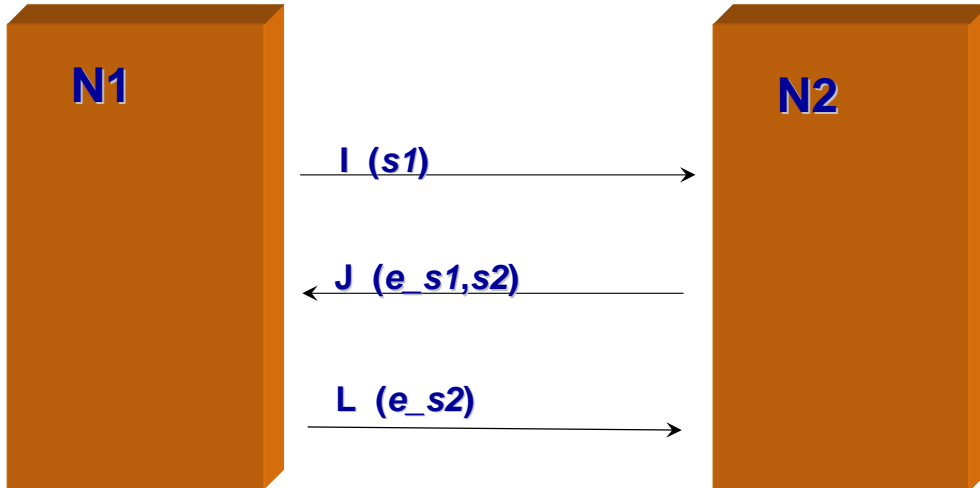
To improve this security, we now allow for DES-encrypted passwords to be sent in NJE connection records in place of the current clear text values.

NJE Security



```
RDEFINE APPCLU NJE.N1.N2  
SESSION (SESSKEY (key))  
UACC (NONE)  
SETROPTS CLASSACT (APPCLU)
```

```
RDEFINE APPCLU NJE.N2.N1  
SESSION (SESSKEY (key))  
UACC (NONE)  
SETROPTS CLASSACT (APPCLU)
```



This figure depicts the definitions needed to establish a secure sign-on. The N1 and N2 in the profile names are the NJE node names of the 2 nodes connecting. The key that is specified must match on both nodes. Only if both sides have a matching key can the sign-on complete.

Notify Processing



- **\$HASP548/\$HASP549 no longer issued from SYSOUT receiver**
 - Too early anyway
 - No way to force for locally created output
- **\$HASP548/\$HASP549 now issued during OUTPUT processing**
 - When job is actually available to TSO RECEIVE
 - Can now force message out for locally created output
 - Can use exit 13 in down-level releases to simulate this behavior
- **\$EXIT 13 functionality replaced by:**
 - **NJEDEF MAILMSG=YES/NO** (since SP 4.3.0!!!)
 - \$EXIT 47 (SYSOUT receiver DSH reception)
 - ◆ Process data set header fields in SYSOUT receiver
 - \$EXIT 40 (JOE creation)
 - ◆ Flags in \$XPL to influence whether messages issued
 - ◆ Flag in Pddb to influence job eligibility for message
 - ◆ Can use exit 13 in down-level releases to move notify message to \$EXIT 40 time
 - HASX13A with OA08145
 - ◆ Exit can provide optional text for \$HASP548 message if output deleted to replace "INVALID USERID"

In this release, JES2 is addressing a long standing complaint about the message issued when SYSOUT for a TSO user is received. Prior to this release, the message was issued early in processing the SYSOUT data set that was being received. If the SYSOUT data set was large and the TSO user did a receive after seeing the message, it is possible that the data set may not yet be available for processing.

The notify processing was moved from SYSOUT reception processing to OUTPUT/SPIN processing. This ensures that the message is not received before the output is ready for processing. However, exit 13 no longer made sense in this environment. As a result, and because it is unlucky, we deleted exit 13. The function of the exit can be replaced by the existing external NJEDEF MAILMSG and new function added in exit 40.

Tracing and Diagnosis



- **New \$TRACE ids added**
 - **34** – Traces control records between JES2 main task code and JES2 code in the NETSRV address space
 - **35** – Traces data records (NCC and NMR) between JES2 main task code and JES2 code in the NETSRV address space
 - **36** – Traces control records between JES2 NETSRV code and IAZNJTCP
 - **37** – Traces data records (NCC and NMR) between JES2 NETSRV code and IAZNJTCP
 - **38** – Traces data passed between IAZNJTCP and TCP/IP
 - **39** – Traces API calls out of IAZNJTCP
- **Based on TRACE= parameters on LINEnnn and NETSRVn statements**
- **All new traces (34-39) include detailed IPCS formatting support from in-storage trace tables**

- **\$EXIT trace (13) traces the XPL on input to and output from the exit effector**

- **Select \$SAVE/\$RETURN traces (1/2,18/19) display routine input parameters or routine output.**

- **Time stamp granularity in trace is now 1/1000 second instead of 1/100.**

Tracing and Diagnosis



▪ **New IPCS formatters added**

- Existing NJE/RJE formatters consolidated into single option from main JES2 panel
- New NETSRV panel accessible from NJE/RJE panel
- Formats relevant control blocks based on
 - ◆ NETSRV device name (from panel)
 - ◆ NETSRV ASID (from panel)
 - ◆ Default ASID from dump
- Can filter based on line, subdevice, or specific control block type
- Can invoke common component formatters from panel

Reader/NJE exit changes



- TCP/IP NJE processing occurs outside the JES2 main task
- INTRDR processing also moving outside JES2 main task
- Main task exits no longer get control for TCP/NJE and INTRDR jobs

- New exits were added corresponding to current exits
- Additional main task exits have been defined
 - When jobs are added to the job queue

- Exits 36 and 37 will still be called but from a different address space
- Exit 8 will be called in some cases instead of exit 7

- Changes are **INCOMPATIBLE** with previous releases of JES2!!!!

- [Session 2656, JES2 z/OS 1.7 Migration, Wednesday, 1:30](#)

The changes to NJE to implement TCP/IP will make it impossible to call the traditional HASPRDR exits in the JES2 main task. Similarly, changes to internal reader processing will also make it impossible to call the traditional HASPRDR exits in that environment. To address this, a new set of input processing exits has been defined. These exits will run in the user environment in the NETSERV address space. In addition, a new exit, exit 51, was defined in the main task when jobs change phase. Data can be passed to exit 51 from other exits. Exit 51 can be used as the ultimate end of input exit in the main task for all input sources.

In the case of exits 36 and 37, the exits will still be called, but they are called from a different address space.

For control block I/O, since the I/O is being done outside the main task, exit 8 instead of exit 7 will be called.

Other goodies



- **Job list command enhancements**
 - OS/390 R4 only implemented filters on commands equivalent to existing commands
 - Many customer requests for specific functionality
 - ♦ Display only jobs AWAITING execution (/Q=XEQ does not)
 - ♦ Purge/display output based on various criteria
 - Number of JOES
 - Amount of total output produced
 - Job completion code
 - Almost all keywords now filterable on all commands
 - Many new filters and display parameters added
 - ♦ **BUSY=, CRTIME=** on job-level commands (**\$xJ**)
 - BUSY= - system on which job is busy
 - System name or number (0-255!)
 - CRTIME= - job creation time
 - ♦ **CC=** on output-level commands (**\$xOJ**)
 - Job completion code
 - ♦ **OUTGRPS, RECORDS, PAGES** on **\$LJ**

Several customer requests have been seen over the last few years asking for specific filters on the job list commands. Although any one specific filter was not requested frequently, it was clear that extending the filtering capabilities to all parameters would satisfy a large class of requests.

Job-level Commands



Command Filter	\$A	\$C	\$E	\$H	\$P	\$T	\$D
CLASS= (C=)	NEW!	NEW!	NEW!	NEW!	NEW!	*	*
PRIORITY= (P=)	NEW!			NEW!	NEW!	*	*
SYSAFF= (S=)	NEW!	NEW!	NEW!	NEW!	NEW!	*	*
HOLD= (H=)		NEW!			NEW!	*	*
SECLABEL=	NEW!	NEW!	NEW!	NEW!	NEW!	NEW!	*
SECLABEL_AFF=	NEW!	NEW!	NEW!	NEW!	NEW!	NEW!	*
SCHENV=	NEW!	NEW!	NEW!	NEW!	NEW!	*	*
SCHENV_AFF=	NEW!	NEW!	NEW!	NEW!	NEW!	NEW!	*
USERID=	NEW!	NEW!	NEW!	NEW!	NEW!	*	*
CARDS=	NEW!	NEW!		NEW!	NEW!	*	*
DELAY=		NEW!			NEW!	*	*
CC=		NEW!			NEW!	*	*
OFFS=	NEW!			NEW!	NEW!	*	*
BUSY=	NEW!	NEW!		NEW!	NEW!	NEW!	NEW!

* - Available prior to z/OS 1.7

Here are all of the job-level filters that were added.

Output-level Commands



Filter \ Command	\$C O	\$P O	\$O	\$T O	\$D O
BURST=	NEW!	NEW!	NEW!	*	*
FCB=	NEW!	NEW!	NEW!	*	*
FLASH=	NEW!	NEW!	NEW!	*	*
FORMS=	NEW!	NEW!	NEW!	*	*
OUTDISP=	NEW!	NEW!		*	*
HOLDRC=	NEW!	NEW!		*	*
PRIORITY=	NEW!	NEW!	NEW!	*	*
PRMODE=	NEW!	NEW!	NEW!	*	*
UCS=	NEW!	NEW!	NEW!	*	*
WRITER=	NEW!	NEW!	NEW!	*	*
RECORDS=	NEW!	NEW!	*	*	*
CC=	NEW!	NEW!	NEW!	NEW!	NEW!
SECLABEL=	NEW!	NEW!	NEW!	*	*
USERID=	NEW!	NEW!	NEW!	*	*

* - Available prior to z/OS 1.7

Here are all of the output-level filters that were added.

\$LJ command



- **\$LJ**

- Displays number of output groups by class and hold status

```
JOB00013 $HASP891 OUTPUT(MULTOUT)
$HASP891 OUTPUT(MULTOUT)    READY=(A=1,S=60,T=1,U=60,V=1,W=60,
$HASP891                      X=1,Y=60,Z=1),HELD=NONE
```

- **\$LJ, LONG**

- Additionally displays total record and page counts for all output groups

```
JOB00013 $HASP891 OUTPUT(MULTOUT)
$HASP891 OUTPUT(MULTOUT)    READY=(A=1,S=60,T=1,U=60,V=1,W=60,
$HASP891                      X=1,Y=60,Z=1),HELD=NONE,
$HASP891                      RECORDS=(READY=88251,HELD=0),
$HASP891                      PAGES=(READY=0,HELD=0)
```

- **\$LJ, OUTGRPS**

- Displays/Filters based on total number of output groups for job

- **\$lj(*),outgrps>100**

- **\$LJ, RECORDS, \$LJ, PAGES**

- Display/Filter based on total/ready/held record or page count for job

- **\$lj(*),records=ready>1000**

- **\$lj(*),records=total>1000**

The \$LJ command still displays the number of output groups associated with the job by class and hold status. Additional optional summary information may also now be displayed using the LONG, RECORDS, PAGES, and OUTGRPS keywords. In addition, OUTGRPS, RECORDS, and PAGES can be used as filters to isolate what jobs are using or will use significant amounts of resource.

Table pair enhancements



- **Problem: Table pairs often require fields in \$UCT data area**
 - Only one \$UCT per installation
 - Only one \$EXIT 0 per installation
 - Difficult for multiple vendors to share \$UCT
 - Up to installation to merge multiple vendor exit 0 routines or \$UCTs

- **Solution: Create alternatives for UCT fields**
 - ADCON or VCON
 - Named token pointer to data area
 - ◆ Can be TASK, HOME, PRIMARY, SYSTEM level
 - ◆ Also SUBSYS level (SYSTEM level, last 4 chars of name set to subsystem name)
 - ◆ CB address assumed in first 4 bytes of token
 - New \$TOKENSR service to create, retrieve, or delete
 - Entry point name (for \$PCETAB/\$DTETAB entry points) - \$ENTRY in a LOADED module

- **In addition, use dynamic exit service to allow multiple exit 0s**
 - Exit name is HASP.\$EXIT0
 - Old exit 0 implementation remains
 - Parameter list is unchanged

A long standing problem with vendors has been the single \$UCT in JES2. Vendor products typically cannot use the UCT because it requires coordination with customers updating the single copy, and possibly with other vendors to ensure field names are unique. This can make it difficult for vendors to code JES2 tables (\$SCANTAB, \$PCETAB, etc.) because the only options for referencing data from those tables is via standard JES2 control blocks (HCT, PCE, etc.) or the UCT.

Table pair enhancements (cont...)



▪ Updated macros and operands

- \$DCTTAB
 - ♦ CHAIN=, CHAINTK=, COUNT=, COUNTTK= PCEPTR=, PCEPTRTK=,
- \$DTETAB
 - ♦ EPLOC=, HEAD=, HEADTK=
- \$PCETAB
 - ♦ CHAIN=, CHAINTK=, COUNTS=, COUNTTK=, ENTRYPT=
- \$PCTAB
 - ♦ ENTRYPT=, RECOVPT=
- \$SCANTAB
 - ♦ CB=
- \$WSTAB
 - ♦ DEVCB=, MODSCB=

\$TOKENSR service



- **New \$TOKENSR service to create, delete, or fetch a name/token**
 - Invokes MVS IEANTRT service under the covers
- **Support for all token types supported by MVS**
 - SYSTEM, HOME, PRIMARY, TASK
- **Also supports type of SUBSYS**
 - SYSTEM token with subsystem name placed in last 4 bytes of name
- **Input is the token name, type, and 16-byte value to place in token (for CREATE) or to return the token (for RETRIEVE)**
 - JES2 tables assume first 4 bytes to be address of a control block
 - Address can be used as CB address on various macros (\$SCANTAB, etc.)

This service can be used by exits to managed named tokens for use by the various JES2 table pairs.

Exit 0



- **New MVS dynamic exit `HASP.$EXIT0`**
- **Invoked immediately after traditional exit 0 processing**

- **Specify in PROGxx member:**
 - `EXIT ADD EXITNAME(HASP.$EXIT0) MODNAME(routine) DSNAME(XXXXXXXXXX)`

- **Dynamic tables may be used in `HASP.$EXIT0` modules**

HOWEVER:

 - Refreshing dynamic exits may result in relocation of routines
 - Tables may move, storage reused, but JES2 is never informed

RECOMMENDATION:

 - Use `$MODLOAD` service from `HASP.$EXIT0` routines to load dynamic tables
 - Example in `HASX00B`

Another long standing problem with vendors has been the single EXIT 0 in JES2. Since EXIT 0 is invoked very early in JES2 initialization, installations had no control over what exits run at this time (except to code or link edit router exits). To address this limitation, JES2 is using the dynamic exit services to invoke multiple exit 0 routines. The parameters needed to set up these exits is listed. One concern is that of tables (`$SCANTABs` for instance) that may exist in one of these exit 0s. The problem is JES2 has no control over when these exits are refreshed. If an exit 0 is refreshed, you could end up with tables in JES2 pointed to the freed storage for the old exit. To prevent this, IBM recommends that any dynamic tables be `$MODLOAd`ed from the dynamic exit 0 into non-dynamic storage. JES2 ships a sample exit 0 that uses this technique.

Performance Improvements



- **SPOOL space allocation processing**
 - Eliminated Step-Must-Complete from SPOOL allocation ENQ
 - ENQ more granular (added IOT address to ENQ minor name)
 - IOT writes occur asynchronously
- **HAM buffering improved**
 - More records buffered in write processing (less I/Os needed)
 - Read buffering improved (especially for large LRECL data sets)
- **\$TRACE processing ENQ removed for most trace records**
 - Affects user environment tracing
 - PLO used to serialize trace table updates
 - \$SAVE/\$RETURN processing greatly improved
 - Main task writing of trace records also improved
- **Start halted/draining SPOOL processing**
 - No longer post jobs if starting a draining volume (was never needed)
 - Job posts for halting volumes offloaded to a separate PCE and is paced (periodic \$WAITS)

A number of performance improvements have been made in z/OS 1.7

•Prior to z/OS 1.7, in order to allocate space on SPOOL a step must complete ENQ was obtained early in the processing. The step must complete would status stop other tasks in the address space. In a multi-tasking address space, this could impact overall performance of the application. In z/OS 1.7, the step must complete was removed from the ENQ (thus not affecting any other tasks in the address space) and the minor name was updated to include the primary allocation IOT address. This allows multiple SPIN data set to allocate SPOOL space without interfering with each other. In addition, if the SPOOL space is being allocated by HAM processing, then the IOT will be written out asynchronously using HAM's data structures.

•HAM buffering has been improved. Each write to a SPOOL volume can now write up to 24 SPOOL blocks (old limit was 12). Read processing will get enough buffers to hold at least 2 of the largest LRECLs in a SYSOUT/SYSIN data set. This allows the read-ahead processing to proceed more efficiently

•JES2 \$TRACE processing in the user address space has eliminated the ENQ that serialized \$TRACE buffers (if the data being traced is small enough). Most user environment \$TRACES are improved especially \$SAVE/\$RETURN tracing. The ENQs were replaced with PLOs. Also the main task writing of trace output records will write multiple records without waiting. This clears out ECSA trace buffers more quickly.

•Prior to z/OS 1.7, the starting a draining/halting SPOOL volume would caused a post of all jobs that had space on the volume. This was not needed for draining volumes (since they were not inhibited by the draining volume). For halting volumes, the post processing for Jobs and SYSOUT has been moved to a separate PCE that will \$WAIT to prevent one member from holding the checkpoint for too long. Posting will occur on multiple MAS members as appropriate.

Input Processing Improvements



- **SYSIN data records with a length of up to 32K supported**
 - Can be sent over NJE using BSC, SNA TCP/IP if adjacent node can handle it
 - Truncated to 254 bytes if not supported by NJE node
- **Support for variable length SYSIN records**
 - If all records are the same length, RECFM set to F
 - If all records are not the same length, RECFM set to V
 - Maximum LRECL placed in PDDB/JFCB
 - Default if 0 or 1 records in data set is RECFM of F
- **Support for carriage control in SYSIN**
 - If SYSIN record has CC, then it is saved on SPOOL
 - RECFM set to A or M based on CC type

A number of improvements have been made to SYSIN data set processing (DD * or DD DATA). These improvements include:

•SYSIN data records can now be up to 32K in length. The previous limit was 254 bytes. The new records can be sent over NJE if the adjacent node (and if store and forward, then all nodes to the destination) support the long SYSIN records. If any node does not support long SYSIN records (NJE term is spanned SYSIN records), then the sender will truncate the record at 254 bytes.

•SYSIN data sets will be marked as RECFM V or F based on whether the records that are passed are the same size or not. The LRECL will be set to the largest record encountered. The default for a null SYSIN data set or a 1 record SYSIN data set is RECFM F.

•Carriage control is also now fully supported for SYSIN records. JES2 will SPOOL CC with the SYSIN record and update the RECFM with A or M based on the CC it detects.

With this support, JES2 can support many more data sets as SYSIN as well as send that data over NJE to other nodes.

Input Processing Improvements



- **Clearer messages for JES2 detected JCL/JECL errors**
 - Example assumes JOBDEF JCLERR=YES is set
- **Processing does not stop after 1st error**

```
//TESTERR JOB MSGLEVEL=(1,1),CLASS=?, JOB00015
//          MSGCLASS=ABC,
//          NOTIFY=IBMUSER23
***** ILLEGAL JOB CARD - VALUE OF CLASS= PARAMETER IS NOT VALID *****
***** ILLEGAL JOB CARD - VALUE OF MSGCLASS= EXCEEDS 1 CHARACTER *****
***** ILLEGAL JOB CARD - VALUE OF NOTIFY= PARAMETER IS NOT VALID *****
/*JOBPARM SYSAFF=FRED
***** INVALID /*JOBPARM CARD - VALUE OF SYSAFF= PARAMETER IS NOT VALID *****
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=*
//SYSUT1 DD DATA,DLM=@@
$HASP106 JOB DELETED BY JES2 OR CANCELLED BY OPERATOR BEFORE EXECUTION
```

Error messages from input processing have been updated to clearly indicate what went wrong. In previous releases, ILLEGAL JOB CARD was all the message would say. Now we identify the keyword in error and attempt to indicate what the problem is. Also, previous releases would stop processing on the first JES2 detected error. Now, JES2 will process the entire job stream in an attempt to locate all errors in the stream.

Input Processing Improvements



- **Input processing messages in JOBLOG of submitter**

- Though messages were in SYSLOG, submitter is probably more interested in messages
- Messages exist even if job is purged
- On reader message can track what jobs were submitted

```

J E S 2   J O B   L O G   --   S Y S T E M   I B M 1   --   N O D E   P O K

09.43.50 TSU00012 ---- MONDAY,    22 AUG 2005 ----
09.43.50 TSU00012 $HASP373 IBMUSER  STARTED
09.49.31 JOB00013 $HASP100 CMTX0252 ON INTRDR                      FROM TSU00012 IBMUSER
09.49.31 JOB00013 $HASP112 CMTX0252 -- INVALID /*JOBPARM CARD - VALUE OF SYSAFF= PARAMETER IS NOT VALID
09.49.32 JOB00013 IRR010I  USERID IBMUSER  IS ASSIGNED TO THIS JOB.
09.50.23 JOB00014 $HASP100 CMTX0252 ON INTRDR                      FROM TSU00012 IBMUSER
09.50.23 JOB00014 $HASP110 CMTX0252 -- ILLEGAL JOB CARD - VALUE OF CLASS= PARAMETER IS NOT VALID
09.50.23 JOB00014 $HASP110 CMTX0252 -- ILLEGAL JOB CARD - VALUE OF NOTIFY= PARAMETER IS NOT VALID
09.50.23 JOB00014 $HASP112 CMTX0252 -- INVALID /*JOBPARM CARD - VALUE OF SYSAFF= PARAMETER IS NOT VALID
09.50.23 JOB00014 IRR010I  USERID IBMUSER  IS ASSIGNED TO THIS JOB.
09.51.28 JOB00015 $HASP100 TESTERR  ON INTRDR                      FROM TSU00012 IBMUSER
09.51.28 JOB00015 $HASP110 TESTERR  -- ILLEGAL JOB CARD - VALUE OF CLASS= PARAMETER IS NOT VALID
09.51.28 JOB00015 $HASP110 TESTERR  -- ILLEGAL JOB CARD - VALUE OF MSGCLASS= EXCEEDS 1 CHARACTER
09.51.28 JOB00015 $HASP110 TESTERR  -- ILLEGAL JOB CARD - VALUE OF NOTIFY= PARAMETER IS NOT VALID
09.51.28 JOB00015 $HASP112 TESTERR  -- INVALID /*JOBPARM CARD - VALUE OF SYSAFF= PARAMETER IS NOT VALID
09.51.28 JOB00015 IRR010I  USERID IBMUSER  IS ASSIGNED TO THIS JOB.
    
```

Another change is that input processing message will appear in the JOBLOG of the submitting job. This is especially useful if the job has an error and does not produce any output. These are the cases where a user does a submit and then looks in SDSF and the job is just not there. It is also useful to see what jobs have been submitted by a job (or a TSO user). However, this will increase the JOBLOG size. You may want to consider JESLOG processing to suppress or spin the JOBLOG data sets.

Input Processing Improvements



- **Input parameter for number of internal readers eliminated**

- No limit on the number of internal readers that can be allocated
- Minimal ECSA for each internal reader

- **\$D RDI command updated**

```

$drdi
      $HASP003 RC=(52),
$HASP003 RC=(52),RDI - NO SELECTABLE ENTRIES FOUND MATCHING
$HASP003          SPECIFICATION

      $drdi
      $HASP603 RDI(INTRDR)
$HASP603 RDI(INTRDR)      STATUS=ALLOCATED,AUTH=(DEVICE=NO,
$HASP603                  JOB=YES,SYSTEM=NO),CLASS=A,HOLD=NO,
$HASP603                  MSGCLASS=,OWNER=(JOB00016,D15USERA,
$HASP603                  ASID=0020),PRTDEST=LOCAL,PUNDEST=LOCAL,
$HASP603                  SYSAFF=(ANY),TRACE=NO

      $drdi,long
      $HASP603 RDI(INTRDR)
$HASP603 RDI(INTRDR)      STATUS=ACTIVE,JOB=(JOB00017,IBMUSER3),
$HASP603                  AUTH=(DEVICE=NO,JOB=YES,SYSTEM=NO),
$HASP603                  CLASS=A,HOLD=NO,JOBCOUNT=1,
$HASP603                  JOBSTAT=(RECORDS=99082,CARD=SYSIN),
$HASP603                  MSGCLASS=A,OWNER=(JOB00016,D15USERA,
$HASP603                  ASID=0020),PRTDEST=LOCAL,PUNDEST=LOCAL,
$HASP603                  SYSAFF=(ANY),TRACE=YES
    
```

The limit on internal readers has been removed. There is no limit on the number of internal readers that can be allocated (created). This implies that there are no pre-allocated internal readers. If you fo a \$DRDI command, it is possible that there are no internal readers allocated and thus no entries are found. If your operations staff use this command, they may need to be briefed to prevent a late night call that the internal readers are missing.

Also, the \$DRDI command now supports a LONG qualifier. This displays information on the number of jobs that this internal reader has processed (JOBCOUNT=) and the information on the current job being processed (record count and current card being processed).

Note that JOB= is the job being submitted at the moment and OWNER= is the job that has allocated the internal reader.

Input Processing Improvements



- **\$T RDI command turns on tracing for individual Internal Reader**
 - Internal reader must already be allocated

```
$trdi (INTRDR),owner=jobid=JOB00016,trace=yes
      $HASP603 RDI (INTRDR)
$HASP603 RDI (INTRDR)      STATUS=ALLOCATED,AUTH=(DEVICE=NO,
$HASP603                      JOB=YES,SYSTEM=NO),CLASS=A,HOLD=NO,
$HASP603                      MSGCLASS=,OWNER=(JOB00016,D15USERA,
$HASP603                      ASID=0020),PRDEST=LOCAL,PUNDEST=LOCAL,
$HASP603                      SYSAFF=(ANY),TRACE=YES
```

There is now a \$TRDI command that only allows you to set TRACE= on an internal reader. This activates selective tracing for this particular internal reader. Unfortunately, the internal reader must be allocated before you can issue this command, Selective tracing currently only applies to trace IDs 11 and 12 (selective \$SAVE/\$RETURN traces). Note filters can be used to identify the internal reader.

RAS enhancements



- **BEA (wild branch) added to \$HASP088 message**
- **64 bit registers in HASP088 if high half of any register non-zero**

```

$HASP088 JES2 ABEND ANALYSIS 585
$HASP088 -----
$HASP088 FMID   = HJE7730      LOAD MODULE = HASJES20
$HASP088 SUBSYS = JES2 z/OS 1.8 DRZ801
$HASP088 DATE   = 2005.224      TIME    = 14.33.16
$HASP088 DESC = INVALID OP-CODE
$HASP088 MODULE  MODULE      OFFSET  SERVICE  ROUTINE      EXIT
$HASP088 NAME    BASE        + OF CALL LEVEL   CALLED       ##
$HASP088 -----
$HASP088 *UNKNOWN 00000002 + 000000 UNKNOWN *ABEND S0C1  7
$HASP088 HASPNUC 00007000 + 007E44 NONE     DIAGXITM    7
$HASP088 HASPNUC 00007000 + 00875C NONE     $JESEFF
$HASP088 HASPNUC 00007000 + 00812A NONE     CBMWRTN
$HASP088 HASPCNVT 07F37000 + 000190 NONE     $CBIOM
$HASP088 PSW     = 071C2000 80000002 ILC = 2 IC = 01
$HASP088 BEA = 00000000 08607324
$HASP088 ASID   = 0018 (HOME) 0018 (PRIM) 0018 (SCND)
$HASP088 PCE    = CNVT      (080CD2A0)
$HASP088 R0     = 00000000 086072F6      00000000 00000000
$HASP088 R2     = 00000000 08607322      00000000 0783C354
$HASP088 R4     = 00000000 080CD2A0      00000000 08243000
$HASP088 R6     = 00000000 001440F0      00000000 08071528
$HASP088 R8     = 00000000 87F37190      00000000 7FFFFBAD
$HASP088 R10    = 00000000 08071528      00000000 00007000
$HASP088 R12    = 00000000 08606F38      00000000 080CD2A0
$HASP088 R14    = 08071528 88607264      00000000 00000000
$HASP088 -----
    
```

The \$HASP088 message has been updated to include the new BEA address (if on a series z9 processor) and if it is available in the SDWA/RB. This can be used to determine how you got to the program check (last branch).

Also, the full 64 bit register is displayed if any of the registers have a non-zero high half. In addition the TEA address is also always 64 bit.

RAS Enhancements



- **SYSZJES2 ENQ obtained on SPOOL volumes**
 - Prevents multiple MASes in a SYSPLEX from mistakenly using the same SPOOL
 - SPOOL volume will not start (or be allocated at cold start)

```
$HASP443 vvvvvv NOT ALLOCATED  
DSNAME AND VOLUME ALREADY OWNED BY ANOTHER MAS, RC=09
```
- **CKPT Reserve now uses ISGENQ**
 - Sug APAR OA06514
 - SYNCHRES always yes
 - I/O errors on reserve cause Reconfiguration Dialog
- **Related ASID added to PCE**
 - If PCE ABENDs, related address space is added to dump
 - Ensures requestor information available
 - Used for SAPI, FSS, etc
- **JES2 main load module is split RMODE**
 - Few modules left below the line but most moved above
 - Required JES2 DLIB/SLIB be a PDSE

•A new ENQ (SYSZJES2 major name) is obtained whenever a SPOOL volume is allocated. Within a SYSPLEX, this prevents one MAS from cold starting on a SPOOL that is being used by another member of the MAS. If you attempt to do this, an updated HASP443 message is issued.

•The CKPT RESERVE was change to an ENQ to address a concern raised in SUG APAR OA06514. The problem in that APAR was that an I/O error on the reserve I/O as a result of SYNCHRES was resulting in JES2 coming down with an ABEND. Though the problem was addressed by GRS APAR OA08596, the JES2 code was updated to use ISGENQ. This allows GRS to pass JES2 a return code if there is a problem with the I/O to get the reserve. In this case, JES2 will enter CKPT reconfiguration.

•The PCE was updated to contain a related ASID field. If the PCE ABEND, then the dump that is taken will include the related ASID. This helps in diagnosing problems where the requestor address space is needed. A number of PCEs now set that field.

•The JES2 main load module (HASJES20) is now a split load module. By doing this, most of the JES2 code can be moved above the 16M line providing up to 1 M of below the line private in the JES2 address space. This requires that the DLIB and SLIB for JES2 are now PDSEs.

RAS Enhancements



- Additional data can be added to \$SAVE/\$RETURN traces
- Used in Internal reader processing to trace records put
 - Another trace that could possibly have sensitive data

```

--- JES2 z/OS 1.8 EVENT TRACE LOG --- NODE POK          MEMBER IBM1 DATE 05.234 ---
09.59.30.420 ID = 11 $SAVE      ASID 0020              CIRDRPUT  R14-R1 = 80930318 077CD3B0 00006E40 7F540280 --
7BC0 6161C9C2 D4E4E2C5 D9F340D1 D6C24040          D4E2C7D3 C5E5C5D3 7E4DF16B F15D6BD4  *//IBMUSER3 JOB  MSGLEVEL=(1,1),M*
7BE0 E2C7C3D3 C1E2E27E C16BC3D3 C1E2E27E          E96B4040 40404040 40404040 40404040  *SGCLASS=A,CLASS=Z,
7C00 40404040 40404040 40404040 40404040          *
09.59.30.440 ID = 11 $SAVE      ASID 0020              RANLZCRD  R14-R1 = 00000000 00FCB108 00000000 7FFF0000 --
09.59.30.440 ID = 12 $RETURN    ASID 0020              RANLZCRD  R14-R1 = 877CD4E0 00000004 00007BC0 7F540288 --
09.59.30.440 ID = 12 $RETURN    ASID 0020              CIRDRPUT  R14-R1 = 80930318 00000000 00006E40 7F540280 --
09.59.30.440 ID = 11 $SAVE      ASID 0020              CIRDRPUT  R14-R1 = 80930318 077CD3B0 00006E40 7F540280 --
7C10 61614040 40404040 404040D5 D6E3C9C6          E87EC9C2 D4E4E2C5 D9404040 40404040  *//          NOTIFY=IBMUSER
7C30 40404040 40404040 40404040 40404040          40404040 40404040 40404040 40404040  *
7C50 40404040 40404040 40404040 40404040          *
09.59.30.440 ID = 11 $SAVE      ASID 0020              RANLZCRD  R14-R1 = 00000000 00FCB108 00000000 7FFF0000 --
09.59.30.440 ID = 12 $RETURN    ASID 0020              RANLZCRD  R14-R1 = 877CD4E0 00000000 00007C10 7F540288 --
    
```

\$TRACE processing for \$SAVE and \$RETURN were updated to include the ability to trace parameter data (from the \$SAVE or \$RETURN). This was exploited by internal reader processing to allow tracing of each record that was written to an internal reader. Installations need to be aware that this could cause sensitive information to appear in the \$SAVE/\$RETURN traces and should protect this data appropriately.

Requirements addressed in z/OS 1.7



- **zBLC**
 - **PUSC1002-464** - JES2 and JES3 NJE support over native TCP/IP
- **SHARE**
 - **SOJES293002** – Delay the \$HASP549 message until NJT receive
 - **SOJES292004** – Provide complete control of NJE store-and-forward
 - **SSJES300353** – Enhance the JES SSI 80 extended status interface
 - **SSJES2038887** – Ability to alter priority of output processed by SAPI
 - **SSJES2038885** – JES2 NJE support over native TCP/IP
- **Other requirements**
 - **MR00069427** – Retain original SYSOUT creation date after offload/reload
 - **MR00069758** – NJE over TCP/IP
 - **MR00074624** - \$DJOBQ,/Q=XEQ
 - **MR00075651** - \$HASP540
 - **MR00076705** – Adding the BFSZ value in msg HASP200 to SMF record type 55
 - **MR050300486** – NJE does not have TCP/IP support
 - **MR0529014658** – JES2 table driven customization constraints for OEM vendors
 - **MR081600245** - \$HASP890 to include total record count for job
 - **MR0829022830** – JES2 job transmitter recovery

Requirements addressed in z/OS 1.7



▪ Other requirements

- **MR083002610** – JES2 should fully utilize SPOOL volumes with >64K tracks
- **MR1117035645** – Define a JES2 and JES3 job class in which the job log is purged based on return code
- **MR1128012551** – SDSF with WLM
- **MR122701390** – JES2 command \$CJ (or \$PJ) does not offer the DUP option in order to cancel duplicate jobs in the queue
- **MR0122036132** – Ability to increase NJEDEF NODENUM without all-member warm start
- **MR05300022428** - \$HASP549 records field is too short
- **MR053003510** - \$HASP375 may show insufficient or misleading information
- **MR0620025414** – Dynamic NODENUM change for JES2
- **MR0707035020** – JES2 spool support for greater than 64k tracks, ie, mod27
- **MR0708031946** – JES2 support for spool data sets larger than 64K tracks
- **MR072203183** - \$CO, \$DO, \$PO, \$TO should reject when ALL used with other filtering options
- **MR0908026157** – Resource shortages on one node cause NJE transmitters to drain on other nodes
- **MR100203499** – JES2 does not have a command which shows the count of output groups a job has generated
- **MR1217046210** – JES2 SYMREC improvements