z/OS

JES2 Diagnosis

Version 2 Release 1

GA32-0993-00

z/OS

JES2 Diagnosis

Version 2 Release 1

GA32-0993-00

Note
Before using this information and the product it supports, read the information in "Notices" on page 129.

This edition applies to Version 2 Release 1 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

About this document

This document supports z/OS® (5650-ZOS). This document is specifically designed for installations running z/OS JES2.

This document provides information on the following tasks:

- Use external symptoms to identify problems in JES2
- Use tools including JES2 traces, the JES2 DEBUG facility, and the interactive problem control system (IPCS) to collect problem data
- Use problem data to locate a JES2 problem or to report the problem to IBM
- Correct the problem, when appropriate

Who Should Use This Document

This document is intended for JES2 system programmers or for anyone responsible for locating and correcting problems in JES2.

The reader should have a detailed knowledge of:

- Assembler language
- Dynamic system dump procedures
- Generalized Trace Facility (GTF) tracing
- Interactive Problem Control System (IPCS) usage
- JES2 commands
- MVS™ commands
- Obtaining stand-alone dump procedures
- Programming techniques
- Setting SLIP traps.

How to Use This Document

Problem analysis and correction requires a knowledge of JES2, MVS, and various diagnostic tools and techniques. *z/OS Problem Management* and *z/OS MVS IPCS User's Guide* contains detailed information about how to diagnose problems. This document does not duplicate that information.

Use this document to diagnose JES2 problems only.

Read and become familiar with:

- Chapter 2, "Collecting problem data," on page 21
- Table 1 on page 25, Trace Identifiers and Their Meanings
- Chapter 4, "Using the JES2 DEBUG facility," on page 91
- Chapter 5, "Using IPCS for diagnosis," on page 93.

These chapters describe the problem data you need to collect and some of the tools used to collect and analyze problem data.

When analyzing problem data, see Chapter 1, "Identifying the problem," on page 1 for a table that describes common symptoms and points to additional information about the problem to help resolve the error, or collect enough information about the error for the IBM Support Center.

Where to Find More Information

This document references the following publications using the shortened version of the document title. The following table lists the shortened titles, complete titles, and order numbers of the documents you might need while you are using this document.

Short Title Used in This Document	Title	Order Number
z/OS TSO/E REXX Reference	*z/OS TSO/E REXX Reference*	SA32-0972
z/OS MVS System Messages, Vol 5 (EDG-GFS)	*z/OS MVS System Messages, Vol 5 (EDG-GFS)*	SA38-0672
Principles of Operation	*z/Architecture Principles of Operation*	SA22-7832
SDSF	*System Display and Search Facility/MVS (SDSF/MVS) Guide and Reference*	SC23-0408
System Network Architecture Formats	*System Network Architecture Formats*	N/A

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or provide any other feedback that you have.

Use one of the following methods to send your comments:

1. Send an email to mhvrcfs@us.ibm.com.
2. Send an email from the "Contact us" web page for z/OS (http://www.ibm.com/systems/z/os/zos/webqs.html).
3. Mail the comments to the following address:
   IBM Corporation
   Attention: MHVRCFS Reader Comments
   Department H6MA, Building 707
   2455 South Road
   Poughkeepsie, NY 12601-5400
   US
4. Fax the comments to us, as follows:
   From the United States and Canada: 1+845+432-9405
   From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address.
- Your email address.
- Your telephone or fax number.
- The publication title and order number:
  z/OS V2R1.0 JES2 Diagnosis
  GA32-0993-00
- The topic and page number that is related to your comment.
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

Do not use the feedback methods that are listed for sending comments. Instead, take one of the following actions:

- Contact your IBM service representative.
- Call IBM technical support.
- Visit the IBM Support Portal at z/OS support page (http://www.ibm.com/systems/z/support/).

z/OS Version 2 Release 1 summary of changes

See the following publications for all enhancements to z/OS Version 2 Release 1 (V2R1):

- *z/OS Migration*
- *z/OS Planning for Installation*
- *z/OS Summary of Message and Interface Changes*
- *z/OS Introduction and Release Guide*

Chapter 1. Identifying the problem

JES2 issues messages for a variety of reasons. In some cases, JES2 may just be passing along information it received from another part of the system that encountered an error. Other times, what looks like an error may actually be confirmation of a normal occurrence. Consider the following questions before you collect any problem data:

- Is the function not doing what is expected? Sometimes what you expect to happen may not be what should happen. Before treating the situation as an error, verify that the current design is working as expected by reviewing *z/OS JES2 Introduction* or *z/OS MVS Initialization and Tuning Guide*
- Are there any other indications of the error? There could be messages from other components that not only confirm an error, but identify where the error occurred. Use the messages from other components to pinpoint the problem before reporting the problem to the IBM Support Center.
- Has this happened before? Examining your installation's documentation about previous JES2 problems could uncover a similar situation to help you solve the current problem.
- Has anything changed? Applying maintenance to your system might cause problems. If possible, after you collect any problem data, remove the changes and restart the system. If the problem does not recur, then determine what went wrong when you applied the maintenance. “Problems applying maintenance” on page 19 contains information about what you need to determine the cause of the error.
- Is JES2 running with the correct versions of the JES2 common storage modules, both from IBM and from the installation? *z/OS MVS Initialization and Tuning Guide* discusses the placement of the IBM® HASCxxxx load modules (CSA or LPA, above or below 16 megabytes of virtual storage); *z/OS JES2 Macros* discusses the placement of the installation modules that must be in common storage. If there are multiple versions or levels of JES2 at your installation, wrong levels of modules might be used (even though version checking is done during JES2 initialization). This can cause the system to perform differently than you expect.
- Were any modifications applied that may have caused the error? You may not be able to answer this particular question until after collecting all problem data and looking at it to determine the cause of failure. You will need a copy of any modifications made if you report the problem to the IBM Support Center.
- Is the problem occurring all the time or only in some instances? Do all commands receive the same error or only certain commands? Do all commands against devices receive the same error or only certain commands? Do all devices receive the same error?

If you suspect the problem is a JES2 problem, the following table directs you to the appropriate section for the problem symptom. Also, see Chapter 2, “Collecting problem data,” on page 21 for information about problem data useful for diagnosing JES2 problems.

Symptom	Problem Area
• JES2 ends abnormally • Message $HASP088 • Message $HASP095	"Abends" on page 3
• Checkpoint problems during JES2 initialization • Errors during JES2 initialization • JES2 ends abnormally during initialization • Message $HASP479 • Problem starting JES2	"Initialization problems" on page 6
• JES2 enters checkpoint reconfiguration dialog	"Checkpoint reconfiguration diagnostic procedures" on page 8
• JES2 does not respond to commands • JES2 waiting and jobs do not start	"Waits" on page 10
• Cannot purge an old output data set • Functional subsystem (FSS) address space abends • I/O error on printer, punch, or reader • Job on printer, but printer not processing • Job output marked "non-selectable" • Printed or punched output not what expected • Print Services Facility™ (PSF) printer status is "draining" instead of "inactive" • Wrong destination on output	"Output problems" on page 10
• Functional subsystem (FSS)-controlled printer is not printing • FSS or functional subsystem application (FSA) abends	"Print services facility (PSF) printer problems" on page 10
• JES2 looping • Processor busy during initialization	"Looping problems" on page 11
• JES2 disastrous error ($DISTERR) • Message $HASP096	"Disastrous errors" on page 11
• Resource shortages • Lack of $HASP100 message • Message $HASP050 in system log (SYSLOG) • Message $HASP304 for internal reader shortages • Spool shortages	"Resource shortages" on page 12

Symptom	Problem Area
• Bind image failure • Incorrect or garbled output to node or remote • I/O errors on nodes or remotes • Message $HASP094 • Message $HASP223 • Message $HASP528 • Message $HASP676 • Message $HASP679 • Message $HASP686 • Node flooded • Node or remote status “hung” or “draining” • Unable to establish connection with a node • Unable to transmit to a non-JES2 node	“Remote (RJE) and node (NJE) problems” on page 15
• Failure to create or delete data space • Message $HASP477	“Data space errors” on page 18
• Access violations and warnings for JES2 output work selection • JES2 Device Work Selection Screening • External Writer Work Selection Screening • SYSOUT Application Program Interface (SAPI) requests (SSI function code 79) • Access logging for: JES2 device work selection screening External Writer work selection screening SYSOUT Application Program Interface (SAPI) requests (SSI function code 79) • Message $HASP186	Chapter 4, “Using the JES2 DEBUG facility,” on page 91
• Jobs running slower than usual • Jobs taking a long time to enter the system from different sources (including local input devices) • Printers slowing down or pausing during output processing • Slow response to commands • Your customers complain of poor response time • Holding checkpoint for extended periods	“Performance problems” on page 18
• Failure while applying maintenance	“Problems applying maintenance” on page 19

See *z/OS JES2 Messages* to obtain information about each message.

Abends

For MVS-type abends (abends not preceded by a dollar sign), respond as indicated in *z/OS MVS System Codes*.

When JES2 detects an abend, or an abend occurs within JES2, JES2 issues a message and an error code. JES2 error codes are preceded by a dollar sign (for example, $Q04) and are part of the $HASP095 message. These error codes are documented in *z/OS JES2 Messages*. If JES2 issues message $HASP088, use this

message as the starting point to determine the cause of the problem. Figure 1 on page 5 shows a sample of the $HASP088 message. See "Description of message $HASP088" on page 5 for more information about the $HASP088 message.

Reasons for a JES2 abend

JES2 can abnormally end because:

- Changes were applied to the system.

  Adding maintenance could cause existing installation modifications to fail because of incompatibilities. Check the documentation for the maintenance you applied to ensure your modifications are still compatible. Also ensure that you installed all corequisite and prerequisite software.

  Commands that can help you locate the cause of this type of problem are:

  – $D OPTSDEF. This command displays the last start type requested and the actual start JES2 performed. Certain changes will not occur across different types of starts. Changes that did not occur could cause incompatibilities with any changes you have applied. See *z/OS MVS Initialization and Tuning Reference* or *z/OS MVS Initialization and Tuning Guide* for descriptions of the parameters that cannot be changed across different start types.

  – $D MODULE(jxxxxxxx). This command displays information about IBM and installation-defined modules. The display includes: the date and time the module was assembled, the storage address where the module is loaded, the JES2 processing environment(s) for the routines within the module, the containing load module, the module's $MODULE SPLEVEL= specification, and the IBM maintenance level or installation/vendor version value. This information can be compared to the routine name list and JES2 processing environment displayed by the $D EXIT(nnn) command, local information about the levels of installation or non-IBM code required or maintenance applied.

  – $D EXIT(nnn). This command displays the routines in use for an exit point within the load module. EXIT(nnn) routines are those routines named on the ROUTINE= parameter of the EXIT(nnn) initialization statement. Only routine names in installation-provided modules or in IBM-provided sample exit modules will be included.

    Use this information in conjunction with the $D MODULE and $D LOADMOD commands to confirm that modules and routines are in sync and at the correct level.

  – $D LOADMOD(jxxxxxxx). This command displays where in storage the specified load module JES2 is using resides. Problems could arise from JES2 using a copy of a module other than the copy you actually intended. See *z/OS MVS Initialization and Tuning Guide* for a description of how to direct JES2 to use a particular copy of a load module.

- Parameters were incorrect in the initialization stream or the operator or JES2 start procedure pointed to a member of SYS1.SHASPARM that does not exist.
- JES2 does not have the proper authority for needed resources.

  An incorrect profile or insufficient authority for a resource could cause JES2 to abend. See your security product's publications for details.
- An error occurred in an installation exit or table. Register 14 in the current save area in the dump contains the address of the code that called the failing service. This code typically indicates the cause of the problem. Remove the modification causing the failure, restart the system, and fix the modification.

For JES2 abends, the system log (SYSLOG) and a dump of the abending job are useful for diagnosis. System Management Facility (SMF) record 45 will also contain the completion code when JES2 abends. See *z/OS MVS System Management Facilities (SMF)* for the format of SMF record 45. If you suspect that an abending job caused JES2 to abend, make sure you obtain the dump of the job and JES2, the most recent assembler listing for JES2, and the source for any modifications before calling the IBM Support Center.

When collecting dumps, ensure that all dumps related to a problem are obtained. JES2 utilizes the remote feature of SDUMP to obtain dumps of multiple MAS members for certain errors. These dumps may at first appear to be duplicate but they are all needed to determine the actual cause of the error.

Whenever you reference a line in the source code, use sequence numbers rather than offsets. Modifications change the offsets and the offset may not help your service representative locate the failing instruction in the copy of the source code the representative has.

Description of message $HASP088

By examining message $HASP088, you may eliminate the need to examine a dump. Figure 1 illustrates message $HASP088.

```
Line  $HASP088 -----------  JES2 ABEND ANALYSIS  ------------
  1   $HASP088 FMID   = HJE6601    LOAD MODULE = HASJES20
  2   $HASP088 SUBSYS = JESA  OS 1.1.0
  3   $HASP088 DATE   = 1999.062        TIME   = 15.14.52
  4   $HASP088 DESC = PROTECTION EXCEPTION
  5   $HASP088  MODULE   MODULE     OFFSET  SERVICE  ROUTINE    EXIT
  6   $HASP088  NAME     BASE     + OF CALL LEVEL    CALLED     ##
  7   $HASP088  -------- --------   ------- -------  ---------- ----
  8   $HASP088  HASPNATS 00053000 + 000572  NONE    *ABEND S0C4
  9   $HASP088  HASPSCAN 00085000 + 001C56  NONE     PSTCONCT
 10   $HASP088  HASPSCAN 00085000 + 001282  NONE     PRTABENT
 11   $HASP088  HASPSCAN 00085000 + 0002EC  NONE     PRKEYWRD
 12   $HASP088  HASPCOMM 00025000 + 0058AC  OY29334  $SCAN       5
 13   $HASP088 FAILING INSTR WAS 58A0B3F0
 14   $HASP088 PSW  = 071C2000 80053572 ILC = 4  IC = 10
 15   $HASP088 ASID = 0010 (HOME) 0010 (PRIM) 0010 (SCND)
 16   $HASP088 PCE  = COMM     (02912008)
 17   $HASP088 R0   = 00000000  7FFDD780  7FFDD780  00000000
 18   $HASP088 R4   = 000B5CE0  00000000  00000001  0291220F
 19   $HASP088 R8   = 0009F8B8  0009F90C  02901008  40404040
 20   $HASP088 R12  = 00053548  02912008  000B5CE0  00053548
 21   $HASP088 ----------------------------------------------
```

Figure 1. Message $HASP088

After the identifying header for the message, the information in the message contains:

Line	Contents
1	The function modification identifier (FMID) and load module in which the abend occurred.
2	SUBSYS is the name you used when starting JES2 and OS is the version of that JES2.
3	The date and time of error.
4	A description of error.
5 - 7	Headers for module-related information.

8 - 12 (in this example)
Column 1 contains the module name. Column 2 contains the hexadecimal starting address of the module. Column 3 contains the offset of the call to the routine. Column 4 contains any service applied to the routine. Column 5 contains the name of the routine called.

13 The address of the failing instruction.

14 The contents of the program status word (PSW), the instruction length counter (ILC), and the interruption code (IC).

15 The address space identifiers (ASID) for the home, primary, and secondary address spaces at the time of failure. If there was only one address space, these values are the same.

16 The name and entry address of the $PCE that was in control at the time of failure.

17 - 20 The contents of registers 0 through 15. If any of these registers are access list entry table (ALET)-qualified, a Q appears next to the address.

This message can have different texts and lengths depending on the type of error and in which environment the error occurred. *z/OS JES2 Messages* shows the format and explanation of the message.

Initialization problems

Indications of initialization problems are:

- Initialization does not complete.
- JES2 ends and issues an abend code.
- JES2 does not start.
- The system loops during JES2 initialization.
- MVS enqueues on STCQUE.

For any error

Any error that occurs during initialization requires that you save the initialization log, LIST data sets, and the last message JES2 issued before the problem occurred. At this point in processing, there is little else that indicates what caused a problem.

JES2 stops or loops

If JES2 stops or loops during initialization, request a console dump by entering the following command from a console with master authority:

```
DUMP COMM=(JES2 HANG DURING INITIALIZATION)
```

The system will respond with the message:

```
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

Reply:

```
R id,JOBNAME=JES2,SDATA=(SERVERS,PSA,SQA,LSQA,RGN,TRT,LPA,CSA,GRSQ,SUM,XESDATA,COUPLE)
```

JES2 issues $HASP479

JES2 issues message $HASP479 when another member is probably holding the JES2 checkpoint lock or when

1. A cold start is in progress with a new checkpoint data set
2. An all-member warm start is in progress
3. An all-member warm start is being performed using the CKPTn option to restart JES.

Ensure that for the system identified, JES2 and MVS are inoperative before responding to the $HASP454 message. Entering the $D MEMBER command for all members of a multi-access spool identifies the member that has the checkpoint lock. You can then reset the jobs on the failed member to be available to other members by issuing the $E MEMBER command.

An abend occurs

For an initialization abend, obtain any logrec data set error records. You can use the interactive problem control system (IPCS) to view the LOGREC buffer in the dump, or use a reporting program such as Environmental Error Record Editing and Printing program (EREP) to view LOGREC records. If the abend occurs while JES2 is processing a job or output, restart JES2 with the REQ option. Starting with REQ differentiates the initialization process from actual JES2 processing and helps eliminate initialization processing as the cause of the problem. When you receive the $HASP400 message, try to hold ($HJ command) or purge ($PJ command) the job causing the abend before entering the $S command.

Initialization statement errors

If you are encountering initialization statement errors, you may have to restart JES2 with the LIST option to determine the failure. Initialization statement errors are most readily visible in the LIST output JES2 creates during initialization. To collect additional information about a problem, you should start JES2 traces 1, 2, 3, 6, 7, 11, 12, 13, 17, 18, 19, and 20 before starting JES2. However, you will not be able to start these traces if JES2 performs a hot start. For a loop, if you can determine where the loop is, set a SLIP trap to determine what processing occurred before JES2 started looping. After the first occurrence of the loop, you can determine where the loop is by examining the JES2 entries in the system trace table in the dump and using that information to set the SLIP trap. Start a GTF trace with the TIME=YES and SUB=MSTR parameters. See the following:

- *z/OS Problem Management* for more information about the system trace table and SLIP traps.
- *z/OS MVS Diagnosis: Tools and Service Aids* for more information about GTF and GTF traces.
- *z/OS MVS System Commands* for more information about the SLIP command.

JES2 does not start

If you can not start JES2, make sure the JES2 procedure points to the correct initialization stream. If MVS appears to be enqueued on STCQUE, you might try respecifying the MVS START JES2 command. Typically, the enqueue on STCQUE

occurs when MVS believes it is starting a task without JES available. The most common cause of this enqueue is the misspelling of the JES2 procedure on the START command or a command to start a secondary JES2 entered the system before the primary JES2 completed initialization processing. If it seems the initialization stream is correct, start another JES2 by changing your MVS initialization to point to the JES2 you want to use.

Checkpoint reconfiguration diagnostic procedures

The JES2 checkpoint reconfiguration is entered for a number of reasons. JES2 can initiate a reconfiguration to correct an I/O error, ask the operator to validate checkpoint data set forwarding information, or move the checkpoint off a volatile coupling facility structure. Also, you can initiate a reconfiguration dialog whenever needed with a `$T CKPTDEF,RECONFIG=YES` command. The dialog is then both a means to correct a problem situation and a way to maintain the coupling facility and DASD. If problems occur during a reconfiguration, JES2 provides a number of diagnostic messages to assist your understanding of the error condition.

JES2 checkpoint reconfiguration diagnostic messages

This topic discusses JES2 error, tracking, and processing delay messages issued during a checkpoint reconfiguration. These messages provide useful information to assist your diagnosis after a JES2 checkpoint reconfiguration problem occurs. Additionally, JES2 issues 'delay' messages, $HASP254 and $HASP257, to suggest actions you can take to complete a JES2 checkpoint reconfiguration if delays persist. See *z/OS JES2 Messages* for the full text of all the messages discussed here.

Error messages

The $HASP095 messages provide a series of $Kxx error codes that point to internal errors in JES2 and interface errors with other MAS members or with JESXCF. Following an abend you will need to collect available problem documentation; see "Information needed to debug a checkpoint reconfiguration error" on page 9.

Message
Meaning

$HASP095

$K25 abend requested by testing and diagnostic problem recreation

$K26 received a $CKX with an incompatible control block version because MAS members are at incompatible service levels

$K27 attempted to join a checkpoint reconfiguration already in progress

$K28 internal error within HASPCKDS or HASPCKRR

$K29 interface error with JESXCF

$K30 internal error within HASPCKRR or possibly an interface error with JESXCF

$K31 internal error detected in HASPCKDS

$K32 internal error detected in HASPCKDS

$K33 internal error detected in HASPCKDS

$K34 unexpected return code from JESXCF IXZXIXcc macro processing

Tracking messages

Several messages, $HASP233, $HASP236, $HASP255, and $HASP285 provide reconfiguration processing status. These are not diagnostic messages, but provide

the operator some indication of how the reconfiguration is progressing. None require a response. $HASP285 informs the operator that a reconfiguration is in progress for the reason stated in $HASP233. If issued, $HASP236 indicates a problem with the driving member, but JES2 recovers by selecting another driving member to replace the failed driving member that $HASP236 indicates and processing continues. These messages are issued to SYSLOG on every member to provide a complete set of messages for future reference.

Message
Meaning

$HASP233
reconfiguration reason

$HASP236
driving member (name) failed

$HASP255
checkpoint reconfiguration completed reason

$HASP285
checkpoint reconfiguration status

Processing delayed messages

Messages $HASP254 and $HASP257 indicate a delay in checkpoint reconfiguration processing. Either message can be informational only because the problem symptom might resolve itself or can offer information you need to diagnose and resolve a persistent problem. These messages and their explanations as presented in *z/OS JES2 Messages* suggest actions you can take to complete a JES2 checkpoint reconfiguration if delays persist.

Message
Meaning

$HASP254
reconfiguration delayed - awaiting member information

$HASP257
member delayed by another for reason

Information needed to debug a checkpoint reconfiguration error

It is important that you obtain information about every MAS member and MVS system in the XCF group. Perform the following:

- Collect the JES2-provided dump of the failing member (which includes a dump of the JESXCF address space and associated data space on this MVS system)
- Collect SYSLOGs of the systems on which the members of the MAS are
- Request a dump of all other MAS members and of the JESXCF address spaces and their data spaces on the MVS systems where the members are
- CTRACE JESXCF data for every member.

  See *z/OS MVS Diagnosis: Tools and Service Aids* for further information on collecting JESXCF CTRACE data.

When you have collected this data, contact your IBM service representative for corrective actions.

Waits

JES2 waits have the same symptoms as MVS waits.

Use SYSLOG to determine the cause of the wait. If you determine that JES2 is the cause of a wait state, ensure that you dump JES2 by entering the following command from a console with master authority:

```
DUMP COMM=(JES2 HANG DURING INITIALIZATION)
```

The system will respond with the message:

```
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

Reply:

```
R id,JOBNAME=JES2,SDATA=(SERVERS,PSA,SQA,LSQA,RGN,TRT,LPA,CSA,GRSQ,SUM,XESDATA,COUPLE)
```

Waits can be caused by resource shortages. See "Resource shortages" on page 12 for additional information.

Output problems

Incorrect output could be caused by an improper load of the forms control block (FCB) or the universal character set (UCS). Ensure that the name of the image is spelled correctly and the image:

- Exists in SYS1.IMAGELIB
- Is correct
- Is proper for the device you are loading.

For an output problem, collect the SYSLOG for the period after a printer started. If the problem is related to a printer, collect the JCL for the job that created the output and dumps of any Print Services Facility (PSF) address spaces that abended. For Data Facility Product (DFP) printers, the output of the DFP error recovery program (ERP) could indicate the problem.

If SYSLOG contains message $HASP151 and this message is preceded by an input output supervisor (IOS) error message, the IOS error contains the actual problem. Follow the sequence of events from the time the job that created the output ran until the output is marked non-selectable to pinpoint why the output is ineligible for processing. Also, if a command was entered that incorrectly changed the destination of output, that command appears in the SYSLOG.

Print services facility (PSF) printer problems

As soon as possible after an error involving an FSS-controlled printer, you must dump the address space of the FSS and JES2. At least one of these dumps must contain the common storage area (CSA). A summary (SUM) dump is not sufficient for PSF problems. Make sure your dump data sets are large enough to hold these dumps. You should also save the soft-copy of SYSLOG for use in later problem analysis. Examining SYSLOG could give you an indication of why an FSS abended

or the printer shows draining with a job still active. You might also find some indication of why JES2 marked a job as non-selectable.

In situations where the job status shows the job active on a printer but the printer is not processing the job, look for either of the following:

- A previous problem
- A canceled job that had problems on the printer. If this is the case, PSF treats the job as still active on the printer and that printer will be unavailable until the next warm start

If you must recreate the problem, set the GTF functional subsystem interface (FSI) trace and JES2 traces 14, 15, 16, 18, and 19. These traces can help pinpoint the problem.

Looping problems

When your system is in a loop, you may see:

- The processor appears busy, but jobs do not start or end.
- JES2 issues the same message repetitively.
- All I/O appears to stop.
- Messages indicating that a job(s) has exceeded limits, but the job does not abend.

Besides a dump, SYSLOG is necessary in order to identify the current processor control element (PCE). The system trace table in the dump is also useful in identifying the PCE.

If it appears a batch job may be causing the loop, the dump of that job is also necessary for problem identification.

Disastrous errors

JES2 issues message $HASP096 when a job encountered a disastrous error. The message identifies the job that encountered the disastrous error.

A “disastrous” error can be:

- a JES2 logic error
- a critical I/O error has occurred on SYS1.HASPACE data set (if the $HASP096 message is accompanied by a I/O error message).

When the situation occurs a $HASP096 message is issued and based on the installation defined RECVOPTS a system dump may be obtained.

Examine the sequence of events in SYSLOG that lead up to the error. Check your RECVOPTS(DISTERR) initialization statement to see if you should change the count defined in the initialization stream. Enter the $D RECVOPTS(DISTERR) command to see the count and interval specified in the initialization statement. Too low a count for RECVOPTS(DISTERR) causes JES2 to suppress dumps when the number of disastrous errors exceeds the count in a 24-hour period; you will lose helpful diagnostic information.

Also determine if any initial program loads (IPL) occurred between the time the job ran and before output for the job was processed. If you must recreate the error, you should set JES2 trace 3.

Resource shortages

JES2 indicates resource shortages through a variety of ways, including the $HASP050 and $HASP304 messages. These messages can be helpful in determining and correcting a resource shortage.

$HASP050 message for resource shortages

JES2 issues message $HASP050 to indicate shortage(s) of one or more of the following resource types:

- BERT - Block Extension reuse tables
- BSCB - Bisynchronous buffers
- BUFX - Extended logical buffers
- CKVR - Checkpoint versions
- CMBS - Console message buffers
- CMDS - Console message buffers used for JES2 commands
- ICES - VTAM® sessions
- LBUF - Logical buffers
- JNUM - Job numbers
- JQES - Job queue elements
- JOES - Job output elements
- NHBS - NJE header/trailer buffers
- SMFB - System management facility buffers
- TGS - SPOOL space/track groups
- TTAB - Trace tables
- VTMB - VTAM buffers

JES2 resource shortages might be temporary and occur during periods of heavy utilization or the shortages might be permanent and impact JES2 processing. Typically, a resource shortage is the result of not specifying a large enough value for a resource in your JES2 initialization statements.

In general, take one or more of the following actions as appropriate for the resource type:

- Increase the quantity of the resource on its corresponding JES2 initialization statement.
- Increase the quantity of the resource with a $T command.
- Decrease demand for the resource (such as purging old held output to relieve a shortage of JOEs).
- Monitor temporary or non-impact shortages for possible future action.

Some resources, such as those described later, require specific actions to determine why there is a resource shortage.

$HASP050 message for CMBs - console message buffers

When a JES2 Console Message Buffer (CMB) shortage occurs, JES2 issues:

```
$HASP050 JES2 RESOURCE SHORTAGE OF CMBS
```

An operator may have issued a command that generated a large volume of messages. This type of shortage is typically temporary. Consider increasing the number of CMBs (specified with CONDEF BUFNUM=) if you encounter a

temporary CMB shortage, even if the shortage occurs infrequently. You can determine your current CMB usage by issuing a $D CONDEF command. The $HASP830 message displays the number of CMBs defined (BUFNUM=) and the number of free CMBs (BUFFREE=).

Continuous, permanent growth in CMB utilization between JES2 restarts can mean a permanent shortage of CMBs. If you experience a permanent shortage of CMBs, the following procedures will help you determine how to resolve the problem.

Examine LOGREC Data: Examine LOGREC data in an EREP report or in a DUMP, using IPCS, to search for symptom records with the following text:

```
$WTO PARAMETER LIST ERROR
```

See "Diagnostic procedures for $WTO PARAMETER LIST ERROR symptom records" on page 113 for further information if you find $WTO PARAMETER LIST ERRORs.

Examine the $WTO parameter list for data that might cause JES2 to place an associated $CMB data area on a queue where the $CMB might remain indefinitely. Obtain a dump of the JES2 member producing the symptom records, if necessary, to continue analysis.

Analyze the Dump Using IPCS: CMBs may be found on a queue or in storage. By obtaining a dump of the JES2 address space, you can use IPCS to analyze the dump to search for CMBs.

- Confirm that a CMB shortage still existed at the time the dump was taken and that a sufficient number of CMBs were defined. To determine the number of CMBs defined, format the $HCT data area, where JES2 saves this information, and look at the following fields:
  - $NUMCMBS (half-word) the total number of static CMBs (specified by CONDEF BUFNUM=)
  - $CMBFREC (half-word) the number of free CMBs

  IBM suggests a minimum of CONDEF BUFNUM=1000. A larger value, up to 9999, can be specified if necessary.
- Search JES2 queues for static CMBs. The following $HCT fields are the queue headings for queues on which a CMB can be placed:

  $COMMQTP
  Queue (LIFO order) of commands from RJE, NJE, internal readers, or other MAS members.

  $BUSYQUE
  Queue (LIFO order) of messages or commands bound for the $HASPWTO subtask (the subtask issues most SVC 34's and 35's for JES2).

  $BUSYRQ
  Queue (CMB priority order) of messages, commands, or NMRs bound for RJE, NJE, or other MAS members.

  $COMMQUE
  Queue (FIFO order) of commands to be processed by this member. This queue is used by module HASPCOMM as a work queue and contains CMBs requeued from CCTCOMMQ, $COMMQTP, and also automatic commands that are ready to execute.

$CONWKQ
Work queue (CMB priority order) used by the $HASPWTO subtask. $HASPWTO re-chains CMBs from the $BUSYQUE to this queue for processing.

$DOMQUE
List (CMBDOMID order) of action messages waiting for eventual processing by a $DOM macro instruction.

$DOMQUEA
Queue (LIFO order) of action messages, issued by the $HASPWTO subtask, that are waiting to be requeued to the $DOMQUE by the $ASYNC processor.

- Use the JES2 IPCS panels to see if a number of CMBs have accumulated on any of the preceding queues. These queue names are on the panel selected from option 7 ("Select JES2 control blocks") of the "JES2 COMPONENT DATA ANALYSIS" panel. See Chapter 5, “Using IPCS for diagnosis,” on page 93.

  You can ignore GETMAINed CMBs (bit CMB2GETM is on in byte CMBFLAG2) because these CMBs are not counted for the HASP050 message.

  If you find that many CMBs have accumulated on one of the preceding queues, determine why they were not being processed. For the following queue types, the reasons may include:

  $COMMQTP
  An internal reader, a remote, or node might be flooding JES2 with commands, or the command processor might be stopped.

  $BUSYQUE
  The $HASPWTO subtask might be stopped or looping because of a problem in SVC 34 or 35 processing. Issue the IPCS command

  ```
  IP SUMM FORMAT JOBNAME(JES2)
  ```

  to analyze the RB structure for the $HASPWTO subtask. You might, for instance, find problems in MVS exits or in WTO SSI broadcast function routines.

  $BUSYRQ
  Look for problems with $MASCOMM, the remote console processor, RJE or NJE. Check whether the remote console processor, multi-leaving line processor, or other NJE or RMT PCE has ended (see message $HASP068) or is stopped. Examine the CMBs on the queue for commonality such as:
  - Command or message type CMBs
  - The destination node, member, remote, or user
  - The source node, member, remote, or user.

  $COMMQUE
  Check if the command processor PCE is stopped.

  $CONWKQ
  The $HASPWTO subtask might be stopped or looping because of a problem in SVC 34 or 35 processing. Issue the IPCS command

  ```
  IP SUMM FORMAT JOBNAME(JES2)
  ```

  to analyze the RB structure for the $HASPWTO subtask. You might, for instance, find problems in MVS exits or in WTO SSI broadcast function routines.

$DOMQUE
Look at the messages to determine where and how the messages might have been issued, or why the messages have not been deleted with a $DOM macro instruction.

$DOMQUEA
CMBs should not stay on this queue very long. See if the JES2 main task is stopped or if the asynchronous I/O processor PCE is stopped.

- Use IPCS BROWSE to search storage, using the FIND command:

  ```
  F 'CMB ' NOB
  ```

  If you find a number of CMBs that are not on any queue, check their contents for information to:
  - Determine if a CMB queue chain might have been broken.
  - Determine if CMBs are obtained with $GETCMB and are not subsequently queued by a $WTO macro instruction with CMB=YES or freed with a $FRECMB macro instruction.

$HASP050 message for JQEs - job queue elements

If the message indicates there are not enough job queue elements ($JQEs) or if jobs are not starting, enter a $D JOBDEF command to determine the available $JQEs. If there are not enough $JQEs, purge unneeded jobs and start, or change, the selection characteristics of your printers so the printers select available work. Processing output for jobs could release $JQEs back to the free queues.

$HASP050 message for TGs - spool space and track groups

If the message indicates that there is a spool shortage, enter the $D JOBQ,SPOOL=(%>nnn) command to determine the percentage of spool that jobs are using. Either offload or purge the jobs using the highest amount of spool to alleviate the problem. All the output for the job must be removed, not just parts of it, because the job still uses spool and JES2 typically does not recover spool space that a job uses until a job purges.

Remote (RJE) and node (NJE) problems

Before doing anything that involves the network, request a dump. This action ensures that any indications of the problem that exist in the system are not changed by any attempts at recovery. For this discussion, problems are grouped into:

- Problems common to RJE and NJE
- NJE problems.

Problems common to RJE and NJE

The following are some of the most common problems that occur in a networking environment:

Node or remote status is draining for several minutes

This condition is commonly called the "hang" situation. Nothing appears to be happening on the line defined for the connection. Try the following to isolate the cause:

- Check the attributes for the remote. The definitions on both sides of the session should be the same. If this is a systems network architecture (SNA) session, make sure the JES2 and Virtual Telecommunications Access Method (VTAM) definitions are compatible.

- Enter the $T LINE(n),LOG=Y command to turn on event tracing. This command displays if there actually is any activity. You should see a $HASP094 message for each transaction across the line. Lack of the $HASP094 message confirms the problem.
- Enter a $P LINE(n) and a $E LINE(n) command to cause the line to drain.
- Enter a $P LOGON(n) and a $E LOGON(n) command to cause the logon to drain.

If the problem persists, you must cause JES2 to abend, using the $P JES2,ABEND command and warm start JES2 to clear the problem. Before restarting the nodes and remotes, set JES2 trace 4 (BSC) or 5 (SNA) to trace the problem if it recurs.

Erroneous I/O errors

When you start a line, a few I/O errors are normal. These indicate an exchange of information between the two ends of the session to establish communication. In certain situations, time-out errors messages are to be expected. For time-out errors, enter a JES2 $S command to get the session working as expected, again. Depending on the type of remote, you may have to restart the other end of the session.

An emulator can also cause I/O errors if it does not correctly emulate **all** hardware functions.

Message $HASP094 contains the sense information you need to ascertain the cause of a networking problem. The command codes for networking are documented in *z/OS MVS System Messages, Vol 5 (EDG-GFS)*.

Bind image failure

Bind image failures are indicated by messages from VTAM.

You can examine the bind image by draining the line in error and enabling JES2 trace 5. The trace output contains the bind image. Inspect the bind parameters and correct them.

Unable to establish connection

The return feedback codes in message $HASP094 indicate what may be wrong and whether the failure is hardware, VTAM, JES2, or something else. You can start event logging by entering a $T LINE(n),LOG=Y command to determine what part of the connection sequence is failing. “NJE problems” describes other things to try in an NJE environment.

NJE problems

In addition to the problems common to RJE and NJE, the NJE environment creates other concerns.

Unable to establish connection

If JES2 does not make a number of connections, you can enter a $D NODE(*) command to list all the nodes and a $D CONNECT command to list their connections. Make sure you route the output of this command to the hard-copy log as it can be quite large and difficult to read on the console. Also, entering a $D PATH(*nodename*) command shows you the paths to one or more nodes as defined to JES2. See *z/OS JES2 Commands* for more information about the $D PATH(*nodename*) command. If the connection was delayed or not completed, message $HASP501 is issued, listing the member that did not respond.

If the node involved is a non-JES2 node, you need to ensure that the initialization statements that define the connection are correct. *z/OS MVS Initialization and Tuning Guide* discusses the considerations for connecting non-JES2 nodes.

Examine SYSLOG to see if JES2 issued any resource shortage messages. A shortage of teleprocessing (TP) buffers or lines can cause connection failures. Also, message $HASP676 indicates network-specific resource shortages. JES2 issues messages $HASP676 and $HASP679 on the system initiating the connection to indicate the cause of the connection failure. The node on the receiving end of the connection may receive message $HASP223, which indicates network-specific resource shortages on that node. In either case, the node with the shortage may require a warm start before the connection can complete.

You can increase the number of teleprocessing buffers available by using the $T TPDEF command. See *z/OS MVS Initialization and Tuning Guide* for more information about teleprocessing buffers.

Note: For the next section , output refers to both jobs and SYSOUT.

Output remains on your node

If output remains on your node, check the following:

- Ensure the node name on the statement that defines the output is correct. If you determine that the node is wrong, enter a $TO or a $R command against the output to reroute it and change the statement in error.
- Ensure the target node is connected.
- Ensure a SYSOUT transmitter is available to the node. You can view the definition of a node by entering a $D NODE(avvvvvvv) command:
  - If the display shows that the connection is made **VIA SPOOL**, reenter the command on the member of the multi-access spool, which is directly connected to the node.
  - If the status of the line shows the line is **UNCONNECTED**, start a line for that node.
  - If the status shows **ACTIVE**, enter a $DU,LINE(nnnn) command for the line displayed in the node display to determine the status of the transmitters.
  - Ensure that the SYSOUT transmitter's work selection characteristics (line and page limits) are set to a range that allows it to select the output. To display or change a transmitter's work selection characteristics, use the $D and $T L(nnnn).ST(n) commands.

If individual jobs are held, check the following:

1. Examine the SYSLOG for any $HASP528 messages.
2. Enter a $L or a $T O command for the held job. If this is SYSOUT received from another node and the HOLDRC=005, the number of nodes through which the SYSOUT has passed exceeded the value of MAXHOPS on the NJEDEF initialization statement.

These symptoms could indicate a loop in your network definitions. By using the $D PATH command to determine the paths to the target node for the output and $D NODE command to display the definition for the nodes in the path, you could locate an existing loop. If no loop exists, ensure that the value of MAXHOPS is large enough to send a job and receive the output from that job through the longest path in your network.

If your node is flooded, use VTAM and the network control program (NCP) with the virtual route extensions to route output past the node. You may also want to ensure that the destination node is operative, especially if your node is adjacent to the destination. Jobs and output destined for a node will travel as far as they can in the network until they can go no further. If the destination is inoperative, you will continue to receive output for that destination until the node becomes operative.

If your node receives incorrect output, note the node from which the output came. Contact the sender, explain the problem, and request a retransmission. The sending node is responsible for the validity of output transmitted through the network and that node must determine their problem.

Recreating NJE problems

Before recreating a sequence of events that caused an NJE problem, set JES2 traces 21 through 24. These records trace the connection records and signon sequence between nodes. The records could be helpful when trying to establish why sessions are not established between nodes or why output is not leaving a node for another destination.

Activate these trace identifiers by using the TRACE parameter on any of the following:

- NODE(nnnn) initialization statement
- $T NODE command
- $T NODE command
- $T LINE command

The TRACE parameter specifies whether JES2 is to trace NCC records to and from a direct network connection to a specified member of the node. The $D NODE,TRACE command shows you whether the traces are active.

Data space errors

JES2 issues message $HASP477 when it cannot create or delete a data space.

To help identify the error, you need the reason code from the $HASP477 message, if the reason code is 12 or greater. You must also have logrec data set available when you call the IBM Support Center.

Performance problems

Some symptoms of JES2 performance problems include:

- Jobs run more slowly than usual
- End-users complain of poor response
- Jobs take a long time to enter the system from all sources
- Printers slow down or pause
- The system responds slowly to commands.
- A member of the MAS holds the checkpoint for extended periods.

SYSLOG is extremely helpful when analyzing a performance problem. If the system appears sluggish, look through SYSLOG and see if the DEBUG facility was enabled. Other items you may want to investigate:

- Active traces. Deactivating any active traces could improve performance.

- Dumps. Performance is affected if the system is currently writing a dump.
- Resource shortages. See "Resource shortages" on page 12 for additional information on resource shortages.
- Checkpoint contention. If you suspect that the performance problem is checkpoint-related, set JES2 trace 17 and use the trace reduction program supplied in SYS1.SHASPARM to reduce the trace output before calling the IBM Support Center.
- Work selection criteria. Selecting certain items in your work selection criteria could be detrimental to your performance. See *z/OS MVS Initialization and Tuning Guide* for more information about work selection criteria.

Problems applying maintenance

Problems applying maintenance to the system generally are not caused by JES2. Your IBM systems engineer (SE) should be able to assist you with these problems. When you do encounter problems, it is important to save the output from Systems Modifications Program/Extended (SMP/E) to aid service in locating the problem.

You should also make sure that you have the correct macro libraries specified when you install a new release.

When updating JES2 source code, ensure that you re-assemble all of the JES2 source modules. Changes to the macro libraries such as SYS1.MACLIB and SYS1.MODGEN could cause problems because some JES2 modules may have been assembled with different macro libraries and are no longer compatible with the remaining source.

To assemble or re-link-edit JES2 modules:

- Using SMP/E, you can use member HASISMPA in SYS1.SHASSAMP.
- Without using SMP/E, you can use member HASIBLD in SYS1.SHASSAMP.

Chapter 2. Collecting problem data

If you suspect the problem you have is a JES2 problem, get as much information about the environment at the time of the error as you can **before doing any other task**. Any attempt to resolve the problem can cause a change in the environment. That change can alter information and possibly destroy what could have been a key indicator to the problem.

Basic information to collect

Whether you decide to diagnose the problem yourself or call the IBM Support Center for assistance, collect:

- A dump of JES2. If JES2 did not write a dump, request one by entering the following command from a console with master authority:

```
DUMP COMM=(JES2 DUMP 9-1-94)

NOTE:  The keyword value of COMM= that is in parenthesis
       is a descriptive title of your choosing.
```

  The system will respond with the message:

```
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

  Reply:

```
R id,JOBNAME=JES2,SDATA=(SERVERS,PSA,SQA,LSQA,RGN,TRT,LPA,CSA,GRSQ,SUM,XESDATA,COUPLE)
```

  To have JES2 automatically dump its storage if a problem occurs twice in a 24-hour period, set the COUNT parameter on the RECVOPTS initialization statement to 2. This dump includes the same information as the previous SDATA requests and could prevent you from having to recreate the problem. JES2 utilizes the remote feature of SDUMP to capture dumps of all members or an MAS for certain types of errors. These remote dumps will have the same symptoms as the original problem and thus may at first appear to be duplicates. However, they are not duplicates in these cases. You must ensure that all dumps that are created for an error are retained for problem analysis.
- A dump of any suspect jobs. This is useful when JES2 ends abnormally and it appears that a job running on your system may have caused JES2 to stop running.

  Enter the $D I,LONG command to display the address space identifier (ASID) of the initiator which the job is running. You can then specify ASID=nn in the SDATA when you request the dump to ensure the system dumps the address space which the failing job is running.
- A copy of the hard-copy log or the most current system log (SYSLOG) available. If you suspect that a job might be causing the problem, the SYSLOG should cover the period of the entire life of the suspect job. **Ensure you save the soft-copy version of the SYSLOG for the failure period until you have resolved the problem.**
- Logrec data set error records, if available. This is especially useful in hardware- and teleprocessing (TP)-related problems.

- Any installation modifications to JES2. Have the source for all exits, table pairs, and installation modifications to JES2.

Additional information you might need

Depending on the complexity of the problem, you may need the following additional information:

- Traces. Traces can complete the picture of the environment at the time of error. There are different types of traces you can run. See Table 1 on page 25 for more information about JES2 traces. Traces include:
  - JES2 traces. The $TRACE facility creates JES2 traces. Different situations can be traced with different trace identifiers. Table 1 on page 25 explains how to enable the traces, how the traces are related to specific functions, and the information in each trace.
  - CCWTRACE. These traces are useful for diagnosing I/O- or TP-related problems. For a hardware problem, CCWTRACE can supplement the information in logrec data set.
  - GTFTRACE. If a GTFTRACE is active for the area that gave you the problem, this trace can be helpful. The GTFTRACE is especially important for VTAM-related problems. If you must recreate the problem for any reason and you can trace the area with GTFTRACE, do so.
  - MTRACE. These traces are useful for diagnosing some problems, if SYSLOG is unavailable. **However, MTRACE should not be used as a substitute for a copy of SYSLOG.**
- LIST output of the initialization process. When the error occurs during JES2 initialization, restart JES2 with LIST or LOG to help you locate the cause of the problem. It is probably the easiest way to trace the initialization statements.
- Program event recording (PER) output. This is the information created from a SLIP trap. See *z/OS Problem Management* and *z/OS MVS System Commands* for information about using the MVS SLIP command to set SLIP traps.
- Environmental, Reporting, Editing, and Printing Program (EREP) output. Use EREP to format and print logrec data set error records. See *EREP User's Guide* for more information about EREP.
- Register contents when using the JES2 DEBUG facility. See Chapter 4, "Using the JES2 DEBUG facility," on page 91 for more information about the JES2 DEBUG facility.
- Output from IPCS. See Chapter 5, "Using IPCS for diagnosis," on page 93 for more information about IPCS.
- If you are experiencing JESXCF address space problems by system abend codes DC5 and EC5, JES2 $HASP501 or MVS IXZ0108E, see *z/OS MVS Programming: JES Common Coupling Services* for procedures on how to dump the JESXCF address space and all associated address spaces. That document also provides an example of how to end and restart the JESXCF address space.

Chapter 3. Establishing JES2 traces

In JES2, there are various processes that have trace points. When enabled, these points give information about JES2 registers, buffers, control blocks, and other pertinent items. During JES2 initialization, you direct JES2 to trace its activity by using the TRACE(n) and TRACEDEF initialization statements. Another way of establishing traces is by entering the $S TRACE(n), $P TRACE(n), and $T TRACEDEF commands. See *z/OS MVS Initialization and Tuning Reference* and *z/OS JES2 Commands* for information about the syntax of initialization and *z/OS JES2 Commands* for statements and commands related to tracing.

You can access trace data in two ways: in a dump of unformatted trace tables residing in the extended common storage area (ECSA) and in formatted system output.Formatting the trace information for system output is known as “logging”.

Setting up a JES2 trace environment with CTRACE

JES2 provides a component tracing (CTRACE) function, SYSjes2, that is started automatically during initialization. SYSjes2 contains three sublevel traces that run continuously and concurrently; they are a JQE service trace, a JOE service trace, and a JES2 dispatcher service trace. Included in the various service traces for SYSjes2 are traces of calls to the JES2 macros, such as $#BUSY and $QBUSY. See *z/OS MVS Diagnosis: Tools and Service Aids* for a detailed description of this component trace.

Setting up a JES2 trace environment at initialization

At JES2 initialization, you can establish a trace environment by specifying:

- The amount of storage to use for trace data. On the TRACEDEF statement: the TABLES parameter specifies the number of trace tables; the PAGES parameter specifies the number of 4-kilobyte pages per TRACE table. When a trace table fills, JES2 switches to the next available table to write the trace records. When the last available trace table fills, JES2 switches to the first table and overwrites the entries at the beginning of the table. This process is known as “wrapping”. Ensure that your trace tables spin off for output processing to avoid the data lost caused by wrapping.
- Whether JES2 should automatically start the TRACE facility. On the TRACEDEF statement specify either YES or NO for the ACTIVE parameter.
- Whether JES2 should log trace data in a trace log data set for later printing. On the TRACEDEF statement specify either YES or NO on the START subparameter of the LOG parameter
- The maximum number of lines the trace log data set can contain before being spun off for output processing. On the TRACEDEF statement specify the value of the SIZE subparameter of the LOG parameter.
- The SYSOUT class of the trace log data set. On the TRACEDEF statement specify the CLASS subparameter of the LOG parameter. Consider the security attributes for the trace data sets and how those attributes affect the printing of the trace data sets.
- Which trace identifiers are active. Use the (n) form of the subscript on the TRACE(n) statement to specify individual trace identifiers, the (n-m) form for a

range of identifiers, or the (n-*) form for a generic range to specify identifiers n to 255. Also, use the START={YES|NO} parameter on the TRACE(n) statement to start the specified identifier(s).

Setting up a JES2 trace environment using commands

You can use commands to begin tracing. For example, the following series of commands shows how you can start TRACE ID=4. It is a simple trace that is easy to run and will show the sequence of events.

Starting A BSC Trace

```
$TTRACEDEF,TABLES=20,LOG=(CLASS=H,SIZE=64000)
$TTRACEDEF,ACTIVE=Y,LOG=(START=YES)
$TLINEx,TR=Y
$STRACE(4)
```

After starting the trace, you then start the remote. At this point, any I/O activity on the specified line is being traced.

Stopping A BSC Trace

```
$TLINEx,TR=N
$TTRACEDEF,SPIN,ACTIVE=N
$TTRACEDEF,LOG=(START=NO)
```

The job, called $TRCLOG, in the class H output queue, has the trace data.

Note: $TRCLOG is the automatically generated job for trace data on the output queue.

Summary of trace identifiers

JES2 assigns each event eligible for tracing a numeric identifier used to reference the event. Each time an activated trace event occurs (such as processing of a $SAVE macro), JES2 writes information to the trace table. The standard trace table entry consists of the trace identifier, the time-of-day clock value, an 8-character symbol taken from the $TRACE macro, the job number or ASID (if available), and the processor control element (PCE) name and address (if available). The trace information is then produced in a formatted output. In addition, each trace event can record other unique information. For example, each traced $SAVE macro also records the contents of registers 14, 15, 0, and 1 the EBCDIC name of the routine that issued the $SAVE macro.

Tracing requires at least 3 trace tables of equal size, which must be in ECSA. Specify the number of trace tables on the TABLES parameter of the TRACEDEF initialization statement and the size of each trace table, in units of 4-kilobyte pages, on the PAGES parameter of the TRACEDEF statement. If a TRACEDEF initialization statement or operator command requests logging by specifying LOG=(START=YES), JES2 formats the contents of the filled trace table and writes the formatted entries to spool while writing the new entries to the next table. Formatting routines in the event trace log processor perform any required formatting for each trace identifier.

If a TRACEDEF initialization statement or operator command requests no logging by specifying LOG=(START=NO), JES2 wraps data from the active log to the oldest log.

Table 1 identifies the valid event identifiers and their meanings.

Table 1. Trace Identifiers and Their Meanings

Trace Identifier	Meaning
1	Trace JES2 $SAVE macro calls for the JES2 main task and FSS environments. This identifier traces all JES2 processors. See trace ID 11 and ID 18.
2	Trace JES2 $RETURN macro calls for the JES2 main task and FSS environments. This identifier traces all JES2 processors. See trace ID 12 and ID 19.
3	Trace JES2 disastrous errors ($DISTERR macro).
4	Trace channel-end completions for BSC lines. A line is eligible for tracing if the associated LINE(nnnn) initialization statement specifies TRaceio=YES.
5	Trace SNA events on VTAM lines and the JES2/VTAM interface. A line or interface (LGNn) is eligible for tracing if the associated LINE(nnnn) initialization statement specifies TRaceio=YES.
6	Trace each time JES2 is initialized or reinitialized.
7	Trace each time JES2 ends along with the ending code.
8,9,10	These identifiers all perform the same function. They are provided so that local debugging may be done by tracing processor flow independently from other trace identifiers. Three identifiers are provided for independent tracing function. Unless JES2 $TRACE macros specifying these identifiers have been placed into JES2 modules, no tracing occurs as a result of enabling these trace identifiers.
11	Trace JES2 $SAVE macro calls. Only processors having TR=Yes option defined (see the $T PCE command), and device-related processors that have the TR=Yes option defined (see the device initialization statements and $T commands) are eligible for tracing with this identifier. The device-related TRACE setting also applies to tasks running in support of those devices in the FSS environment. In the user environment, tasks running with JES2 SSI function routines for which the SSI(nnn) setting is TRACE=YES are eligible for tracing with this identifier. See trace ID 1 and ID 18.
12	Trace JES2 $RETURN macro calls are to be traced. In the user environment, tasks running with JES2 SSI function routines for which the SSI(nnn) setting is TRACE=YES are eligible for tracing with this identifier. See trace ID 2 and ID 19.
13	Trace JES2 exit points. Exits to be traced are determined by the EXIT(nnn) initialization statement and the $T EXIT(nnn) operator command.
14	Trace the FSIREQs (GETDS, RELDS, SEND) on behalf of functional subsystems only when the $DCT for the printer allows tracing. Use the $T PRT(nnnn),TR=Yes command to enable the trace.
15	Trace the FSIREQs (GETREC, FREEREC, CHKPT) on behalf of functional subsystems only when the $DCT for the printer allows tracing. Use the $T PRT(nnnn),TR=Yes command to enable the trace.
16	Trace the functional subsystem CONNECTs and DISCONNECTs only when the $DCT for the printer allows tracing. Use the $T PRT(nnnn),TR=Yes command to enable the trace.
17	Trace statistics associated with checkpoint performance. Trace records include values for each phase of the checkpoint cycle.
18	Trace JES2 $SAVE macro calls for the user environment. This identifier traces all JES2 processors. See trace ID 1 and ID 11.
19	Trace JES2 $RETURN macro calls for the user environment. This identifier traces all JES2 processors. See trace ID 2 and ID 12.
20	Trace the $#GET calls.
21	Trace the interchange between the local node and another node when the 2 nodes are establishing a session. JES2 sends network connect control records (NCC) I, J, K, L, and M during the exchange. See *Network Job Entry (NJE) Formats and Protocols* for a description of NCC records.
22	Trace the NCC “M” records JES2 received that add a node into the network. See *Network Job Entry (NJE) Formats and Protocols* for a description of NCC records.
23	Trace the NCC “M” “N” records JES2 received that remove a node into the network. See *Network Job Entry (NJE) Formats and Protocols* for a description of NCC records.
24	Trace the NCC “M” or “N” records JES2 has rejected that attempted to update the network topography. See *Network Job Entry (NJE) Formats and Protocols* for a description of NCC records.
25	Trace information about the functional subsystem (FSS) checkpoint.
26	Trace the automatic restart manager requests processed by the JES2 main task.
27	Trace information about PSO external writer processing.
28	Trace SYSOUT Application Program Interface (SAPI) information between the user address space and the JES2 main task.

Table 1. Trace Identifiers and Their Meanings (continued)

Trace Identifier	Meaning
29	Trace SSI information between the user address space and the JES2 main task concerning the SAPI.
30	Trace identifier 30 traces all $#POST macro calls made by JES2 processing.
31	Trace identifier 31 traces all $QGET macro calls made by JES2 devices and initiators.
32	Trace identifier 32 traces all $#REM macro calls made by JES2 processing. It can be used to determine why a JOE was deleted.
33	Trace identifier 33 traces all NJE Headers, trailers and NMR records received and transmitted by JES2. A header is traced if tracing was requested for the line (TRACEIO=YES on the LINE statement) or for the adjacent node (TRACE=YES on the NODE statement).
34	Trace identifier 34 traces all data records passed between the JES2 address space and the NETSRV address space. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).
35	Trace identifier 35 traces all control records passed between the JES2 address space and the NETSRV address space. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).
36	Trace identifier 36 traces all data records passed between JES2 code in the NETSRV address space and IAZNJTCP. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).
37	Trace identifier 37 traces all control records passed between JES2 code in the NETSRV address space and IAZNJTCP. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).
38	Trace identifier 38 traces all data records passed between IAZNJTCP and TCP/IP. These records are traced if tracing is requested on the line (TRACE=COMMON=YES on the LINE statement) or NETSRV (TRACE=COMMON=YES on the NETSRV statement).
39	Trace identifier 39 traces TCP/IP API calls issued by IAZNJTCP. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).
40	Trace identifier 40 traces the results of the WLM initiator balancing computation at the beginning of every checkpoint cycle.
41	Trace identifier 41 traces the results of the WLM initiator balancing computation at the end of every checkpoint cycle.
42	Trace identifier 42 traces all $CDCTDYN macro calls that are made by JES2 processing.
43	Trace identifier 43 traces every ENF58 event that is sent.
44	Trace identifier 44 traces every ENF58 event that is received.
45	Trace identifier 45 traces every ENF70 event that is sent.
46	Trace identifier 46 traces every ENF70 event that is received.
47	Trace identifier 47 traces every ENF78 event that is sent.
48-255	These trace identifiers are available for both IBM and customer use. To avoid an overlap of identifiers, begin numbering identifiers with 255 and progress downward.

Modifying the JES2 trace environment

The operator can alter the trace environment at any time through the use of the $T TRACEDEF, $S and $P TRACE(n) commands. The $T TRACEDEF command allows the operator to:

- Redefine the amount of storage dedicated to tracing
- Turn tracing on and off
- Cause logging of the traced data
- Stop the logging of traced data
- Cause JES2 to spin the trace log data set immediately
- Change the SYSOUT class for the trace log data set.

Enter the $D TRACEDEF or $D TRACE(n) command to display the status of the trace environment.

Enter the $S and $P TRACE(n) commands to activate and deactivate previously defined trace identifiers. You can specify the ASID, JOBNAME, or JOB_NUMBER and TCB_ADDRESS when filtering the JES2 trace point.

Trace output samples

The following topics include samples of JES2 trace output with an explanation of the output.

Note: JES2 formats its trace output so all output ends on a fullword boundary. Any extra bytes are added to the end of the trace information and contain zeros.

Trace ID=0 sample

Figure 2 is a sample of trace identifier 0. JES2 automatically starts trace identifier 0. It shows the number of trace tables in effect, and total and recent discards. TOTAL DISCARDS is the number of entries JES2 discarded since JES2 was warm-started or cold-started. RECENT DISCARDS is the number of entries JES2 discarded since the last event JES2 traced successfully. Be aware that adding the recent number to the total number does not always add up to the next total because the recent number is kept in the trace table page and the total number is kept in the $HCCT control block.

```
14.56.13.09  ID =   0 TRACE EVENTS DISCARDED        TRACE TABLES =   25
                      TOTAL DISCARDS =   819744     RECENT DISCARDS =     231
```

Figure 2. Sample Output From Trace Identifier 0

Trace ID=1, ID=2, ID=18, and ID=19 sample

Figure 3 on page 28 is a sample of trace output for trace identifiers 1, 2, 18, and 19. See an explanation of the fields at "Trace record contents" on page 28.

In addition to the tracing of $SAVE and $RETURN macros for the routines listed in the figure, JES2 also traces $SAVEs and $RETURNs associated with a functional subsystem.

```
17.13.49.87783 ID=  1 $SAVE              ASYNC     0C117620  $FREEBUF   ---------------------------------------------
R14-R1 = 00000000/32B495A8_8000F6A2 00000000/00000000_0000DE38 00000000/00000000_00000000 00000000/00000000_0C206000
17.13.49.87784 ID=  2 $RETURN            ASYNC     0C117620  $FREEBUF   ---------------------------------------------
R14-R1 = 00000000/32B495A8_8000F6A4 00000000/00000000_00000000 00000000/00000000_00000000 00000000/00000000_0C206000
17.13.49.87784 ID=  1 $SAVE                        004FF260  H607RSN    ---------------------------------------------
R14-R1 = 00000000/32B495A8_80009120 00000000/40414300_0002A110 00000000/00000000_FFFFFFFF 00000000/00000000_00007C28
17.13.49.87785 ID=  2 $RETURN                      004FF260  H607RSN    ---------------------------------------------
R14-R1 = 00000000/32B495A8_80009120 00000000/40414300_00000000 00000000/00000000_00000040 00000000/00000000_00007C28
17.13.49.88023 ID= 18 $SAVE    ASID 002C           004FF890  SSIWTA     ---------------------------------------------
R14-R1 = 00000000/00000000_83C1E324 00000000/00000000_8B9F6CDC 00000000/00000000_00947584 00000000/00000000_804E2E50
   4E2E50  E2E2D6C2 001C0009 004FC094 00000000   004E2E6C 00000000 00000000          *SSOB.......m.....+..........    *
   4E2E6C  00140000 004E2E80 00000000 00000000   00000000                           *.....+.............             *
17.13.49.88024 ID= 18 $SAVE    ASID 002C           004FF890  HFJLOGTM   ---------------------------------------------
R14-R1 = 00000000/00000000_8BA2E1C6 00000000/00000000_0BA813E8 01FF0014/00000000_00001624 00000000/00000000_7F564E80
17.13.49.88025 ID= 18 $SAVE    ASID 002C           004FF890  HFJOBLOG   ---------------------------------------------
R14-R1 = 00000000/00000000_8BA81506 00000000/00000000_0BA81158 01FF0014/00000000_000000C0 00000000/00000000_00000000
17.13.49.88026 ID= 19 $RETURN  ASID 002C           004FF890  HFJOBLOG   ---------------------------------------------
R14-R1 = 00000000/00000000_8BA81506 00000000/8BA733B6_00000000 01FF0014/00000000_000000C0 00000000/00000000_00000000
17.13.49.88027 ID= 18 $SAVE    ASID 002C           004FF890  HFJDLINE   ---------------------------------------------
R14-R1 = 00000000/00000000_8BA81528 00000000/8BA733B6_004E7120 01FF0014/00000000_00001624 00000000/00000000_7F564E80
17.13.49.88030 ID= 19 $RETURN  ASID 002C           004FF890  HFJDLINE   ---------------------------------------------
R14-R1 = 00000000/00000000_8BA81528 00000000/8BA733B6_00000000 8BA5357A/00000000_00001624 00000000/00000000_7F564E80
17.13.49.88031 ID= 19 $RETURN  ASID 002C           004FF890  HFJLOGTM   ---------------------------------------------
R14-R1 = 00000000/00000000_8BA2E1C6 00000000/8BA733B6_00000000 8BA5357A/00000000_00001624 00000000/00000000_7F564E80
17.13.49.88031 ID= 19 $RETURN  ASID 002C           004FF890  SSIWTA     ---------------------------------------------
R14-R1 = 00000000/00000000_83C1E324 00000000/00000000_00000000 00000000/00000000_00947584 00000000/00000000_804E2E50
   4E2E50  E2E2D6C2 001C0009 004FC094 00000000   004E2E6C 00000000 00000000          *SSOB.......m.....+..........    *
   4E2E6C  00140000 004E2E80 00000000 00000000   00000000                           *.....+.............             *
17.13.49.88066 ID= 18 $SAVE    ASID 002C           004FF890  $RACROUT   ---------------------------------------------
R14-R1 = 00000000/00000000_8BA7DA2C 00000000/00000000_0BA0AA50 8BA5357A/00000000_00000000 00000000/00000000_7F55FCB0
```

Figure 3. Sample Output From Trace Identifiers 1, 2, 18, and 19

Trace record contents

Figure 4 is a sample record, which is broken into fields that are subsequently described.

```
--------------+------+--------+---------+----------+--------+-----------+
     (1)      | (2)  |  (3)   |   (4)   |   (5)    |  (6)   |    (7)    |
--------------+------+--------+---------+----------+--------+-----------+
17.13.49.87783|ID=  1|$SAVE   |         | ASYNC    |0C117620| $FREEBUF  |-----------------------------------------------
--------------+------+--------+---------+----------+--------+-----------+
|================================================== (8) ==========================================================|
R14-R1 = 00000000/32B495A8_8000F6A2 00000000/00000000_0000DE38 00000000/00000000_00000000 00000000/00000000_0C206000
              |      |        |         |          |        |           |
17.13.49.88023|ID= 18|$SAVE   |ASID 002C|          |004FF890| SSIWTA    |-----------------------------------------------
--------------+------+--------+---------+----------+--------+-----------+
|================================================== (8) ==========================================================|
R14-R1 = 00000000/00000000_83C1E324 00000000/00000000_8B9F6CDC 00000000/00000000_00947584 00000000/00000000_804E2E50
   |================================================ (9) ============================================================|
   4E2E50  E2E2D6C2 001C0009 004FC094 00000000   004E2E6C 00000000 00000000          *SSOB.......m.....+..........    *
   4E2E6C  00140000 004E2E80 00000000 00000000   00000000                           *.....+.............             *
```

Figure 4. Fields on Traces 1, 2, 18, and 19

Field **Contents**

1 Time-of-day clock value when JES2 created the trace record.

2 and 3

Trace identifier and function:

ID = 1 $SAVE issued from the JES2 main task and FSS environments.

ID = 2 $RETURN issued from the JES2 main task and FSS environments.

ID = 18

$SAVE issued from the user environment

ID = 19

$RETURN issued from the user environment.

4 Address space identifier (ASID) or job identifier associated with the program issuing the $TRACE macro, if available.

5 $PCE name associated with the $TRACE macro, if available.
6 $PCE address (JES2 main task) or TCB address (outside JES2 main task).
7 1- to 8-character name of the routine that issued the $SAVE or $RETURN macro.
8 Access Registers and corresponding 64-bit Registers 14, 15, 0, and 1.
9 SSOB and extension when tracing entry and exit to any SSI routine.

Trace ID=3 sample

Figure 5 is a sample of trace output for trace identifier 3. JES2 creates these records when processing a $DISTERR macro or when processing a error detected by $CBIO in the user environment. The record traces I/O errors or incorrect control blocks and contains the buffer at the time of error.

```
               ---  JES2 SP 5.2.0 EVENT TRACE LOG  ---  NODE POK      MEMBER IBM1 DATE 94.239
13.34.47.58  ID =   3 $DISTERR STC00083   STCINRDR  03C58410  AT CBIMPL4  IN HASPNUC

JQE/JCT Address (or zero) 03CE0464 MTTR (or zero) 00008D02
  3E45000  C2E4C640 001643C0 00000000 22000000   00000000 00000000 00000000 00000000  *BUF ............................*
  3E45020  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
  3E45040  00000000 00000000 00000000 00000000   03E44000 00000000 00000000 00000000  *.................U .............*
  3E45060  00000000 00000000 C9D6E340 C4C5C1D3   D3D6C340 00000053 A9CA8963 00000000  *........IOT DEALLOC ....z.i.....*
  3E45080  10000800 00008D02 00000000 00000000   00000000 00000000 00000143 0000044C  *...............................*
  3E450A0  00000000 00000000 00000000 00000B80   00000280 00000000 00000000 00000000  *................................*
  3E450C0  00000000 00000000 00000000 00000000   00000000 00000000 00008D03 C0010A00  *................................*
  3E450E0  00000000 00000000 01008F0A 00008D04   80000000 00000000 00000000 00000000  *................................*
```

Figure 5. Sample Output From Trace Identifier 3

Trace record contents

Table 2 is the sample record broken into fields which are described after the table.

Table 2. Fields on Trace 3

1	2	3	4	5	6	7 - end
13.34.47.58	ID = 3	$DISTERR	STC00083	STCINRDR	03C58410	AT CBIMPL4 IN HASPNUC

- Record 1:

  Field **Contents**
  1 Time-of-day clock value when JES2 created the trace record.
  2 and 3
  Identifier and function.
  4 Job identifier associated with program issuing the $TRACE macro, if available.
  5 $PCE name associated with the $TRACE macro, if available.
  6 $PCE address.
  7 - end
  Label and module from which JES2 issued the $DISTERR macro.

- Record 2:

  ```
  JQE/JCT Address (or zero) 03CE0464 MTTR (or zero) 00008D02
  ```

  provides the address in memory of the JQE or JCT of the job for which processing was being done at the time of the $DISTERR. The record also includes the MTTR that provides the spool extent (M), track (TT) and record (R) of the record associated with the $DISTERR.

- Record 3:

  The third record contains the contents of the first 256 bytes of the buffer, if available.

Trace ID=4 sample

Figure 6 is a sample of trace output for trace identifier 4, a bisynchronous communication (BSC) buffer trace. This trace can help you diagnose BSC problems. When active, this trace records both inbound and outbound buffers for network job entry (NJE) and remote job entry (RJE) lines.

The input to this trace is the input/output buffer (IOB) which contains the channel program running on the line being traced. See *Principles of Operation* for more information about channel command words (CCW) and channel status words (CSW).

```
20.49.05.14  ID =   4 BSC-BUFR           MLLM      0260A630  BSC BUFFER TRACE FOR LINE19
    B7290  C2E4C640 00000000 00000000 E1000000   02716B50 00000000 42000000 7FBD48A0  *BUF              ,&           *
    B72B0  000B72F8 0C000206 000B72D8 009F4CC4   000B72D8 00000000 00000000 C5000001  *   8       Q   D   Q        E   *
    B72D0  0700323D 00000000 140B72D0 60C20001   01048BF4 60CA0002 27000000 60C70001  *            -B     4-       -G  *
    B72F0  020B7318 20C40208 00000000 00C00000   00000000 00C00000 00000000 00C00000  *     D                        *
    B7310  00000000 00C00000                                                         *                              *
   CCW     140B72D0 60C20001                                                         *    -B                        *
    B72D0  07                                                                        *                              *
   CCW     01048BF4 60CA0002                                                         *   4-                         *
    48BF4  323D                                                                      *                              *
   CCW     27000000 60C70001                                                         *    -G                        *
   CCW     020B7318 20C40208                                                         *     D                        *
    B7318  323D                                                                      *                              *
```

Figure 6. Sample Output From Trace Identifier 4

This description does not discuss the trace header record because the format of the header is the same as the format for trace identifier 3. See "Trace ID=3 sample" on page 29 for the description the header of trace identifier 3. The trace header record is the record where ID=4 appears.

Trace record contents

Table 3 is the sample record broken into fields which are described after the table.

Note: The text within the asterisks (*) at the end of each record is not described because that text is a translation of the EBCDIC codes in the first part of the record.

Table 3. Fields on Trace 4

1	2	3	4	5	6	7	8	9
B7290	C2E4C640	00000000	00000000	E1000000	02716B50	00000000	42000000	7FBD48A0
B72B0	000B72F8	0C000206	000B72D8	009F4CC4	000B72D8	00000000	00000000	C5000001
B72D0	0700323D	00000000	140B72D0	60C20001	01048BF4	60CA0002	27000000	60C70001

Field **Contents**

1 The address of the I/O buffer or data this record traces. You will see trace entries with the same addresses as JES2 reuses IOBs.

2 - 7 Trace buffer prefix.

8 - first 2 characters
I/O completion flags from the IOB. Successful completion = X'42'. Error = X'46'.

8 - last 4 characters
First 2 sense bytes of the I/O operation. See sense information for individual devices.

9 Event control block (ECB) completion codes and the JES2 pointer to the ECB. Common completion codes are:
X'41' I/O error. Check the sense information.

X'44' IOB intercept. CSW is not valid.
X'48' I/O was purged.
X'7F' Normal completion.
X'FF' Requeued BSC buffer. This value only occurs on a JES2 system.

row 2, field 2 - last 6 characters
This is the beginning CSW without its first byte. This information is 7 bytes long. The first 3 bytes is an address that points 8 bytes past the last CCW completed. The count field in this CSW is the residual data count of the CCW just completed. The actual number of bytes read or written is the difference between this value and the CCW byte count in the CCW just completed.

row 2, field 4 - first 2 characters
The start I/O (SIO) condition code.

row 2, field 4 - next 4 characters
Address of the start of the channel program.

Row 3, field 4
Channel programs.

The output contains CCW entries preceded by the characters CCW. Data associated with a particular CCW immediately follows the CCW. See *z/OS MVS System Messages, Vol 5 (EDG-GFS)* for a complete list of completion codes and information about networking I/O.

See the description of message $HASP094 in *z/OS JES2 Messages* for additional information about some of the previous fields.

Trace ID=5 sample

Trace identifier 5 logs VTAM Request Parameter Lists (RPLs) sent over SNA lines. It can be used to diagnose SNA protocol problems and problems related to the data being sent. In addition to the VTAM RPL, this trace contains sessions status information and buffer control data. Trace identifier 5 records both inbound and outbound RPLs for network job entry (NJE) and remote job entry (RJE) sessions. A trace record is created whenever an RPL is completed (either by VTAM or JES2 processing). Figure 7 on page 32 shows a sample of two trace identifier 5 records.

```
13.06.44.92  ID =   5 SNA-BUFR           MLLM      02F1E500  LU3774   ICESYMB FOR LINE17  ---------------------------

ICE VALUES PRIOR TO PROCESSING RPL:
ICESTAT=12  ICEFLAGS=00  ICERCVST=00  ICESNDST=00  ICEINDEX=11  ICERSPCT=00

BUFFER PREFIX:
 2F16290  C2E4C640 00000000 00000000 16000000   030AA5A0 00000000                   *BUF .............v.....       *

RPL:  TYPE=INQUIRE  DEVCHAR   CONTROL=DATA
 2F162A8  00201A70 800A3A14 000AA642 11000000   00000024 84800000 000FA920 00000000  *.........w.........d.....z.....*
 2F162C8  02F163AC 02F16390 28800000 00000000   00000008 00000008 02109040 00000000  *.1...1..................... ....*
 2F162E8  80800000 A4800000 00000000 00000000   00000000 00000000 00000000 03135820  *....u.............................*
 2F16308  40004010 00000000 00000000 00000000   00060000 02F1E500 00000000 00000000  * . ..................1V.........*
 2F16328  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
 2F16348  00000000 800A3A14 80D2C000 02800000   02F162A8 030AA648 030AA5A0 00000000  *.........K.......1.y..w...v.....*
 2F16368  03135820 000A7340 000A72B0 000A36CE   00000000 000A46CE 00006000 000A56CE  *.......  ..................-.....*
 2F16388  00000000 00000000                                                         *........                       *

DATA:
 2F16390  D0000040 00000000 03135820 D3E4F3F7   F7F44040 D9C5C3D6 D9C44040 C06D0010  *... ........LU3774  RECORD  ....*
 2F163B0  01800000                                                                  *....                           *

13.09.30.97  ID =   5 SNA-BUFR           MLLM      02F1E500  LU3774   ICESYMB FOR R10.CON ---------------------------

ICE VALUES PRIOR TO PROCESSING RPL:
ICESTAT=40  ICEFLAGS=55  ICERCVST=E0  ICESNDST=11  ICEINDEX=00  ICERSPCT=01

BUFFER PREFIX:
 2F16290  C2E4C640 00000000 00000000 16000000   030AAE30 00000000                   *BUF ...................        *

RPL:  TYPE=SEND               SEQ=0003  RESP-TYPE=DR1         CONTROL=DATA    FM-HEADER   ONLY IN CHAIN  BRACKET=EB
 2F162A8  00202270 800A3A14 000AA642 01000000   40041024 80800000 000FA920 00000000  *.........w..... .........z.....*
 2F162C8  02F16390 0100002A 28800000 00000000   00000006 00000190 02109040 00000000  *.1.......................... ....*
 2F162E8  80800003 A2800000 00000000 00000000   00000000 00000000 00000000 03135820  *....s.............................*
 2F16308  00004011 00000000 02F16340 00000000   00060000 030233E8 00000000 00000000  *.. ......1. ...........Y........*
 2F16328  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
 2F16348  00000000 800A3A14 80D2C000 02800010   02F162A8 02F16390 030B94F0 00000000  *.........K.......1.y.1....m0....*
 2F16368  03135820 000A41BC 800A3338 000A36CE   00077078 000A46CE 00006000 000A56CE  *...........................-.....*
 2F16388  00000000 00000000                                                         *........                       *

DATA:
 2F16390  06010000 24000000                                                         *........                       *
```

Figure 7. Sample Output from Trace Identifier 5

A trace record is divided into sections as described in Table 4.

Table 4. Sections in a Trace Identifier 5

Section name	Description
Header line	This contains information about the session and device associated with the RPL.
ICE values	The status information extracted from the $ICE control block before processing the information received in the RPL. See the $ICE control block for the meanings of the status bytes.
Buffer prefix	Control information for the JES2 buffer which contains the RPL. See the $BUFFER control block for an explanation of these fields.
RPL	The RPL associated with the request. This includes information passed to VTAM and a JES2 extension to the RPL. See the RPL control block description in *VTAM Programming* for an explanation of the RPL fields. The JES2 extensions to the RPL for SNA processing are defined in the $MODULE macro.

Table 4. Sections in a Trace Identifier 5 (continued)

Section name	Description
Data	The data that was sent or received by the RPL, if any. The meaning of this data is dependent on the RPL request type, the current state of the session and the type of session (NJE or RJE). If the RPL is not a SEND or RECEIVE DATA request, see *VTAM Programming* for the format of the data for the specific request type. If this is a SEND or RECEIVE DATA, and the RPL contains an FM header, then see *System Network Architecture Formats* to determine the format of the FM header. If the RPL is a SEND or RECEIVE DATA request and is not an FM header, then the format depends on the session type. (NJE or RJE). For further information regarding NJE sessions, see *NJE Formats and Protocols.*

The fields that appear on the header line for trace identifier 5 are illustrated in Table 5.

Table 5. Header Line for Trace Identifier 5

Time	Trace id	Trace name	PCE name	PCE addr	Buffer source	Device name
13.09.30.97	ID = 5	SNA-BUFR	MLLM	02F1E500	LU3774 ICESYMB	FOR R10.CON

In addition to the standard information provided in a JES2 trace header, trace identifier 5 also contains:

Buffer source
This indicates where the buffer came from. It contains either:

session ICESYMB
'session' is the SNA application identifier associated with the RPL. It is the application that the local JES2 is connected to.

REQUEUED BUFFER
Indicates that this is not the first trace entry for this buffer. When the buffer was originally queued to JES2, it could not be processed and was requeued for later processing.

SNA BUFFER TRACE
Indicates that JES2 could not associate a specific session with this buffer.

Device name
The device currently associated with this session. Depending on the state of the session, this can be either:

- a specific device (for example, R10.CON) when the device is active
- the line (for example, LINE10) when there are no active devices
- the logon (for example, LOGON1) when the session is in the process of logging on or off.

The sub header of the RPL contains additional information extracted from the RPL. This information is formatted to simplify interpreting the trace record. For a detailed explanation of these fields, see *VTAM Programming*, Table 6 describes the fields extracted from the RPL.

Table 6. Description of Keywords That May Appear on the RPL Header Line

Keyword	Source RPL Field	Description
TYPE=t	RPLREQ	RPL request type

Table 6. Description of Keywords That May Appear on the RPL Header Line (continued)

Keyword	Source RPL Field	Description
RESPONSE	RPLSRTYP	For a RECEIVE, SEND, or SESSIONC request, when 'RESPONSE' is displayed, this RPL is a response.
SEQ=s	RPLSEQNO	For a RECEIVE, SEND, or SESSIONC request, the sequence number associated with the RPL.
RSP-TYPE=r	RPLVTFL2	If this is not a response ('RESPONSE' is not displayed), then it indicates the type of response the sender expects to receive. If this is a response ('RESPONSE' is displayed), it indicates the type of response sent by the receiver. RESP-TYPE is a combination of the following VTAM response types: **EX** Exception or negative response **DR1** Definite response 1 (formerly known in SNA as an FME response) **DR2** Definite response 2 (formerly known in SNA as an RRN response)
DEVCHAR	RPLOPT9	For an INQUIRE request, indicates device characteristics are to be retrieved.
SESSPARM	RPLOPT10	For an INQUIRE request, indicates session parameters are to be retrieved.
CONTROL=c	RPLCNTRL	Indicates the type of information being sent or received in this RPL.
FM-HEADER	RPLOPT12	For SEND and RECEIVE requests with CONTROL=DATA, indicates that the data contains an FM header. FM headers used by JES2 are mapped by $FMH.
x IN CHAIN	RPLCHN	For requests that can be chained, this indicates which segment in the chain (FIRST, MIDDLE, LAST, or ONLY) is contained in this RPL.
BRACKET=b	RPLRH3	Indicates whether this RPL is to begin or end a bracket. **BB** indicates this RPL begins a bracket. **EB** indicates this RPL ends a bracket. **BB+EB** indicates this RPL both begins and ends a bracket.
RPLRTNCD=	RPLRTNCD	Displays the return code associated with this RPL.
RPLFDB2=	RPLFDB2	Displays additional return code information.
RPLFDBK2=	RPLFDBK2	Displays the sense data associated with this RPL.

Trace ID=6 and ID=7 sample

Figure 8 is a sample of output from trace identifiers 6 and 7. JES2 writes the trace 6 record at the beginning of JES2 initialization and the trace 7 record when JES2 is ending.

```
19.59.22.25  ID =   6 JES2UP             INIT      000A2F40  JES2 INITIALIZATION, OPTIONS= COLD,NOREQ
20.08.24.69  ID =   7 JES2DOWN           COMM      0260A840  JES2 TERMINATION, CODE = $PJ2

20.09.38.80  ID =   6 JES2UP             INIT      000A2F40  JES2 INITIALIZATION, OPTIONS= WARM,NOREQ
```

Figure 8. Sample Output from Trace Identifiers 6 and 7

Trace record contents

Table 7 is the sample record broken into fields which are described after the table.

Table 7. Fields on Traces 6 and 7

1	2	3	4	5	6
19.59.22.25	ID = 6	JES2UP	INIT	000A2F40	JES2 INITIALIZATION, OPTIONS= COLD,NOREQ

Field **Contents**

1 Time-of-day clock value when JES2 created the trace record.

2 Trace identifier.

3 The internal label associated with the trace record.

4 $PCE name associated with the $TRACE macro.

5 $PCE address.

6 A description of the processing associated with this record.

Trace ID=8, ID=9 and ID=10 sample

JES2 provides three independent trace identifiers that you can use to create trace points in installation exits. If you specify these identifiers in $TRACE macros, JES2 writes records tracing data. You can specify variable data to be generated by using the DATA= and LEN= keyword on the $TRACE macro. Figure 9 is a sample of output from trace identifiers 8, 9, and 10. In this example, the DATA= keyword has been specified on a $TRACE macro for trace ID 9 to log the contents of a portion of storage.

```
14.40.04.52  ID =   8 SYMBOL  COMM   0316F008  BEGIN   HAS BEEN REACHED
14.40.04.52  ID =   9 SYMBOL  COMM   0316F008  LOOP    HAS BEEN REACHED
        0  00000000 0000050F BF000000                                 *
14.40.04.52  ID =   9 SYMBOL  COMM   0316F008  LOOP    HAS BEEN REACHED
        0  00000000 0000050F BF000000                                 *
14.40.04.52  ID =   9 SYMBOL  COMM   0316F008  LOOP    HAS BEEN REACHED
        0  00000000 0000050F BF000000                                 *
14.40.04.52  ID =   9 SYMBOL  COMM   0316F008  LOOP    HAS BEEN REACHED
        0  00000000 0000050F BF000000                                 *
14.40.04.52  ID =   9 SYMBOL  COMM   0316F008  LOOP    HAS BEEN REACHED
        0  00000000 0000050F BF000000                                 *
14.40.04.52  ID =  10 SYMBOL  COMM   0316F008  FINISH  HAS BEEN REACHED
```

Figure 9. Sample Output from Trace Identifiers 8, 9 and 10

Trace record contents

Table 8 is the sample record broken into fields which are described after the table.

Table 8. Fields on Traces 8, 9, and 10

Record 1					
1	2	3	4	5	6
14.40.04.52	ID = 9	SYMBOL	COMM	0316F008	LOOP HAS BEEN REACHED

Record 2			
0	00000000	0000050F	BF000000

- Record 1:

Field	Contents
1	Time-of-day clock value when JES2 created the trace record.
2	Trace identifier.
3	The internal label associated with the trace record.
4	$PCE name associated with the $TRACE macro.
5	$PCE address.
6	An acknowledgment that the trace point has been reached.

- Record 2:

  The storage to be traced.

Trace ID=11 and ID=12 sample

Figure 10 is a sample of output from trace identifiers 11 and 12. These records trace the $SAVE macro for processors that have the TR=Yes option specified, and for the $RETURN macro. "Trace ID=1, ID=2, ID=18, and ID=19 sample" on page 27 describes the contents of the records.

```
18.05.27.95609 ID= 11 $SAVE               PRT1      09F601A0  $GETUNIT   ------------------------------------------
R14-R1 = FFFFFFFF/00000000_8002C1A6 FFFFFFFF/00000000_0000B680 00000000/00000000_00000000 00000000/00000000_0A5168B8
18.05.27.95609 ID= 11 $SAVE               PRT1      09F601A0  $DCBDYN    ------------------------------------------
R14-R1 = FFFFFFFF/00000000_8000B6F2 FFFFFFFF/00000000_09D47F50 00000000/00000000_00000000 00000000/00000000_0A5168B8
18.05.27.95611 ID= 12 $RETURN             PRT1      09F601A0  $DCBDYN    ------------------------------------------
R14-R1 = FFFFFFFF/00000000_8000B6F4 FFFFFFFF/00000000_00000000 00000000/00000000_00000000 00000000/00000000_0A5168B8
18.05.27.95611 ID= 12 $RETURN             PRT1      09F601A0  $GETUNIT   ------------------------------------------
R14-R1 = FFFFFFFF/00000000_8002C1A6 FFFFFFFF/00000000_00000000 00000000/00000000_00000000 00000000/00000000_0A5168B8
```

Figure 10. Sample Output from Trace Identifiers 11 and 12

Trace ID=13 sample

JES2 writes trace identifier 13 before and after passing control to exits with tracing enabled. Figure 11 is a sample of output from trace identifier 13.

```
18.52.34.608 ID =  13 $EXIT    ASID 0018           0045F628   ------------------------------------------------------------
  #  37: ENVIRON=         USER  LABEL= RRXIT37   PRE INVOCATION
    R0-R7  = 7F509A88 7F4C7568 7F297CF0 0000000E 08C92400 00951518 00007000 7F297CB0
    R8-R15 = 08B349F8 00000000 7F297CB0 00951518 0825E050 7F330BE0 8825E7D8 0827E0A8
    XPL:  XPLIND=02  XPLCOND=80  XPLRESP=00
        0  5BE7D7D3 01250102 80000028 00000000   7F297CF0 7F297CB0 D1C3E340 08C92000  *$XPL.............@0..@.JCT .I..*
       20  00000000 00000000                                                       *........                       *
18.52.34.608 ID =  13 $EXIT    ASID 0018           0045F628   ------------------------------------------------------------
  #  37: ENVIRON=         USER  LABEL= RRXIT37   POST INVOCATION
    LAST ROUTINE CALLED = DIAGXITC R15-R1 = 00000000 7F509A88 7F4C7568
    XPL:  XPLIND=02  XPLCOND=80  XPLRESP=00
        0  5BE7D7D3 01250102 80000028 00000000   7F297CF0 7F297CB0 D1C3E340 08C92000  *$XPL.............@0..@.JCT .I..*
       20  00000000 00000000                                                       *........                       *
18.52.34.611 ID =  13 $EXIT                       0045F628   ------------------------------------------------------------
  #   6: ENVIRON=      SUBTASK  LABEL= XCSTCUEF  PRE INVOCATION
    R0-R7  = 00000000 0013FA6C 88C3B2AA 08C43808 00000080 20C431F4 08C43808 0013FB88
    R8-R15 = 0013F410 00000001 08C92000 00007000 0001A1C2 7F330BE0 8001A930 00000000

        0  002F0140 0401B401 07C4C5C1 D3D3D6C3   B30201F1 01F0A201 04000000 02B00102  *... .....DEALLOC...1.0s.........*
       20  F0F0AE01 05F0F0F0 F0D2B201 01C1FE                                       *00...0000K...A.                *
18.52.34.611 ID =  13 $EXIT                       0045F628   ------------------------------------------------------------
  #   6: ENVIRON=      SUBTASK  LABEL= XCSTCUEF  POST INVOCATION
    LAST ROUTINE CALLED = DIAGXITC R15-R1 = 00000000 00000000 0013FA6C

        0  002F0140 0401B401 07C4C5C1 D3D3D6C3   B30201F1 01F0A201 04000000 02B00102  *... .....DEALLOC...1.0s.........*
       20  F0F0AE01 05F0F0F0 F0D2B201 01C1FE                                       *00...0000K...A.                *
```

Figure 11. Sample Output from Trace Identifier 13

Trace record contents

Table 9 on page 37 is the sample record broken into fields which are described after the table.

Table 9. Fields for Trace 13

RECORD 1								
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
10.36.42.44	ID = 13	$EXIT						
RECORD 2								
1		**2**		**3**		**4**		
# 6:		ENVIRON= SUBTASK		LABEL= XCSTCUEF		PRE INVOCATION		
RECORD 3								
R0-R7 =	00000000	001344E0	83D33A4A	03A97810	00000087	20A9722C	03A97810	001345D0
RECORD 4								
R8-R15 =	001340D0	00000000	03C49000	00006000	0003FE70	00134618	80040430	00000000
RECORD 5								
0	00360480	6E0107D7	C1D9D4D3	C9C24A01	820CE2E8	E2F14BD7	C1D9D4D3	C9C208E3
RECORD 6								
20	E2D6D2C5	E8F0F046	0103E2C8	D9270105	C3D3D6E2	C5FE		

- Record 1:

  Field **Contents**
  1 Time-of-day clock value when JES2 created the trace record.
  2 and 3
  Identifier and function.
  4 Address space identifier (ASID) or job identifier associated with the program issuing the $TRACE macro, if available.
  5 $PCE name associated with the $TRACE macro, if available.
  6 $PCE address (JES2 main task) or TCB address (outside JES2 main task).
- Record 2:

  Field **Contents**
  1 Exit identifier.
  2 Environment in which the exit resides.
  3 Internal label associated with the exit.
  4 Indicates whether this trace occurred before control passed to the exit or when the exit returned control. This field contains either PRE INVOCATION or POST INVOCATION.
- Record 3 contains the contents of registers 0 through 7 at the time of the trace.
- Record 4 contains the contents of registers 8 through 15 at the time of the trace.
- Records 5 and 6 contain converter text information if the trace entries are for exit 6. Otherwise, records 5 and 6 contain the contents of the $XPL, if the contents of the $XPL are available.

Trace ID=14 and ID=15 sample

JES2 creates trace identifiers 14 and 15 to trace GETDS, RELDS, ORDER, ORDER RESPONSE, SEND, CHKPT, GETREC, and FREEREC functional subsystem interface (FSI) requests. Figure 12 on page 38 is a sample of the output from traces 14 and 15. The record format is similar for these two traces.

The number of records for trace identifier 14 varies depending on the FSI request. Trace identifier 15 is three records long. All records after the first record contain trace information from the FSI including flags and other data.

These traces are only enabled for printers which have tracing turned on by the $T PRT(nnnn),TR=Yes command.

```
16.16.04.97  ID =  14 FSILINK1 JOB00026   PRT111             FSPORDER  R14-R1 = 8005BD92 0005BEC6 00000000 00BF9304
    9A838  C4C3E340 12000041 C6E2E240 000080C0   C6E2C140 01A80104 00F0C000           *DCT     FSS     FSA      0      *
16.16.06.68  ID =  14 FSILINK1 ASID 0016  PRT111             FSMGETDS  R14-R1 = 00FD5656 80000000 009ED640 000CFC58
    CFC58  000000AC 00000003 00010001 00000000   00000000 000CFB70 00000800 00000164  *                               *
    CFC78  000D1A80 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *                               *
    CFC98  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *                               *
    CFCB8  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *                               *
    CFCD8  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *                               *
    CFCF8  00000000 00000000 00000000                                                 *                               *
16.16.06.76  ID =  14 FSILINK1 ASID 0016  PRT111             FSMGDSRT  R14-R1 = 00FD5656 80000000 009ED640 000CFC58
    CFC58  000000AC 00000003 00010001 00000000   00000000 000CFB70 97806400 00000164  *                               *
    CFC78  000D1A80 009C44B8 00000000 00000000   00000000 009B77F0 00000000 00000000  *                       0       *
    CFC98  0000001C 009B77C0 009C42F4 00000000   00000000 00000000 00000000 00000000  *           4                   *
    CFCB8  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *                               *
    CFCD8  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *                               *
    CFCF8  00000000 00000000 00000000 01A80104   00F0C000 00000000 001CD1D6 C2F0F0F0  *               0        JOB000*
    CFD18  F2F88000 00004040 40404040 4040D1C5   E2F24040 4040D1C5 E2D4E2C7 D3C70010  *28            JES2    JESMSGLG  *
    CFD38  02000000 00000000                                                          *                               *
16.16.31.96  ID =  15 FSILINK2 ASID 0016  PRT111             FSMGETRC  R14-R1 = 00FD5656 80000000 009ED640 000CFA78
    CFA78  00000044 00000004 00010001 00000000   00000000 00000000 80008400 00000000  *                               *
    CFA98  00000000 00000000 0000001C 009B77C0   009C42F4 00000000 00000000 00000000  *                   4           *
    CFAB8  00000000                                                                   *                               *
16.16.31.96  ID =  15 FSILINK2 ASID 0016  PRT111             FSMFRERC  R14-R1 = 00FD5656 80000000 009ED640 000CFAC0
    CFAC0  00000038 00000005 00010001 00000000   00000000 00000000 009B46D4 0000001C  *                           M   *
    CFAE0  009B77C0 009C42F4 000CFB00 D1C5E2F2   00000000 00000000                    *       4    JES2               *
16.16.31.96  ID =  15 FSILINK2 ASID 0016  PRT111             FSMGETRC  R14-R1 = 00FD5656 80000000 009ED640 000CFA78
    CFA78  00000044 00000004 00010001 00000000   00000000 00000000 40000000 00000000  *                               *
    CFA98  00000000 00000000 0000001C 009B77C0   009C42F4 00000000 00000000 00000000  *                   4           *
    CFAB8  00000000                                                                   *                               *
16.16.32.03  ID =  14 FSILINK1 ASID 0016  PRT111             FSMGETDS  R14-R1 = 00FD5656 80000000 009ED640 000CFC58
    CFC58  000000AC 00000003 00010001 00000000   00000000 000CFB70 97806400 00000164  *                               *
    CFC78  000D1A80 009C44B8 00000000 00000000   009F6B70 00000000 00000000 00000000  *                 ,             *
    CFC98  0000001C 009B77C0 009C42F4 00000000   00000000 00000000 00000000 00000000  *           4                   *
    CFCB8  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *                               *
    CFCD8  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *                               *
    CFCF8  00000000 00000000 00000000                                                 *                               *
16.16.32.04  ID =  15 FSILINK2 ASID 0016  PRT111             FSMGETRC  R14-R1 = 00FD5656 80000000 009ED640 000CFA78
    CFA78  00000044 00000004 00010001 00000000   00000000 00000000 80008400 00000000  *                               *
    CFA98  00000000 00000000 0000001C 009B7A80   009C42F4 00000000 00000000 00000000  *                   4           *
    CFAB8  00000000                                                                   *                               *
```

Figure 12. Sample Output from Trace Identifiers 14 and 15

Trace record contents

Table 10 is the sample record broken into fields which are described after the table.

Table 10. Fields for Traces 14 and 15

RECORD 1								
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	
16.16.06.76	ID = 14	FSILINK1	ASID 0016	PRT111		FSMGDSRT	R14-R1 = 00FD5656 80000000...	
RECORD 2								
CFC58	000000AC	00000003	00010001	00000000	00000000	000CFB70	97806400	00000164
RECORD 3								
CFC78	000D1A80	009C44B8	00000000	00000000	00000000	009B77F0	00000000	00000000
RECORD 4								
CFC98	0000001C	009B77C0	009C42F4	00000000	00000000	00000000	00000000	00000000
RECORD 5								
CFCB8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
RECORD 6								
CFCD8	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
RECORD 7								
CFCF8	00000000	00000000	00000000	01A80104	00F0C000	00000000	001CD1D6	C2F0F0F0
RECORD 8								
CFD18	F2F88000	00004040	40404040	4040D1C5	E2F24040	4040D1C5	E2D4E2C7	D3C70010
RECORD 9								
CFD38	02000000	00000000						

Trace identifier 14 consists of a differing number of records depending on the FSI request. Table 11 on page 40 shows the number of records for each FSI function traced.

Trace identifier 15, which is written for a GETREC and FREEREC request, is only three records long. The following describes the contents of all the records although some records may not appear for some of the requests.

- Record 1:

Field	Contents
1	Time-of-day clock value when the $TRACE was processed.
2	Trace identifier.
3	FSI function name. This is FSILINK1 for trace 14 and FSILINK2 for trace 15.
4	ASID or job identifier associated with the functional subsystem.
5	Device name associated with the functional subsystem.
6	TCB address.
7	Internal label associated with this trace entry.
8	Contents of registers 14, 15, 0, and 1.

- Records 2, 3, and 4:

  Note: These records contain only the GETDS FSI parameter list when the trace occurs before the FSI allocates a data set.

Bytes	Contents
1 - 3	Device control table (DCT) eyecatcher.
4 - 7	DCT flags DCTSTAT, DCTFLAGS, DCTFLAG2, and DCTFSSL.
8 - 10	FSS eyecatcher.
11 - 14	FSSCB flags FSSTYPE, FSSFLAG1, FSSFLAG2, and FSSFLAG3.
15 - 17	FSA eyecatcher.
18 - 25	FSACB flags FSAFLAG1, FSAFLAG2, FSAFLAG3, FSAFLAG4, FSAFLAGO, FSAFLAGI, and FSAFLAGR, and FSAFLAG5.
26 - 94	The remainder of the FSACB, which is X'42' bytes long. Bytes 15 to 94 is the entire FSACB.

- Records 5 and 6:

  These records can contain either the FSI parameter list (FSIP) or FSWFLAGS. The eyecatcher is FSIP for the FSI parameter list or FSWFLAG for the FSWFLAGS.

  If the information is an FSIP, the data is:

Bytes	Contents
1 - 4	FSIP eyecatcher.
8	The hexadecimal length of the FSI parameter list.
Remainder	The remainder of the record is the FSI parameter list. The number of bytes depends on the value in byte 8.

  If the information is an FSWFLAG, the data is:

Bytes	Contents
1 - 8	FSWFLAG eyecatcher.
9 - 10	The FSWFLAG.
11	Not in use.

12 FSWORDID.

Bytes 9 through 12 are the functional subsystem support processor work area flags (FSSWORK).

13 - 16 RETC eyecatcher.
17 - 20 The return code of the requested FSS function.
21 - 24 FSWK eyecatcher.
25 - 84 The functional subsystem work area, which contains the response to an order.
85 - 92 RETN JIB eyecatcher.
93 - 96 The address of the job information block (JIB) the FSS is returning.

For GETDS, RELDS, and SEND FSI requests after the FSS has allocated a data set, the trace contains the following information:

- Records 7, 8, and 9:

Bytes Contents
13 - 20 The FSACB flags, FSAFLAG1, FSAFLAG2, FSAFLAG3, FSAFLAG4, FSAFLAGO, FSAFLAGI, FSAFLAGR, and FSAFLAG5.
21 - 24 The JIB flags, JIBFLG1, JIBFLG2, JIBFLG3, and JIBFLG4.
25 - 26 The JIB job number, in hexadecimal.
27 - 34 The JIB job identifier. This is either JOB, STC, or TSU and the EBCDIC equivalent of the decimal job number.
35 - 38 The graphic control block (GCB) flags.
39 - 70 The GCB data set identifier. This information is the procedure name, stepname, ddname and data set priority.

Note: The number of records depends on whether records 5 and 6 contain the FSI parameter list or an FSWFLAG. If the output contains the parameter list, it may spill into record 7 causing the output to go to eight or nine records.

Table 11. FSS Orders By Trace Size

2 Record Output	
Order Name	**Function**
FSPORDER	Orders to an FSA. The FSA may be either active or inactive.
3 Record Output	
Order Name	**Function**
FSISEND	Communication between JES2 and the FSA or FSS.
FSICHKPT	Request to FSI to checkpoint the data set.
7 Record Output	
Name	**Function**
RSPRELDS	Response to a RELDS order.
FRSPORDR	Response to an FSPORDER when the FSA is not active.
RSPSTFSA	Response to a start FSA order when the FSA is not active.
FSMGETDS	Order to get a data set before the actual allocation.
FSMRELDS	Order to release a data set after the actual allocation.
FSMSEND	Send an order to an FSA before allocating a data set.
8 or 9 Record Output	
Name	**Function**
ORDSTFSA	Order to start an FSA.
FSMGDSRT FSMRDSRT FSMSNDRT	Response to get data set order after data set allocation. Response to release data set order after data set allocation. Response to order sent to FSA after allocating a data set.

Trace ID=16 sample

JES2 creates trace identifier 16 every time it issues a connect or disconnect order to a functional subsystem. Figure 13 is a sample of the output from a trace identifier 16.

This trace is only enabled for printers that have tracing turned on by the $T PRT(nnnn),TR=Yes command.

```
23.37.19.90  ID =  16 FSICONCT ASID 0015                    FSMCONCT  R14-R1 = 80BBB3F4 809E0808 00BCF8E0 009F1E38
   9F1DF8  00000040 000000FE 00010000 00000000   00000000 00000000 00000080 009F1E38  *                               *
   9F1E18  009F1B90 00000002 0000C074 D1C5E2F2   00000000 00000000 00000000 00000000  *           JES2                *
```

Figure 13. Sample Output from Trace Identifier 16

Trace record contents

Table 12 is the sample record broken into fields which are described after the table.

Table 12. Fields for Trace 16

RECORD 1								
1	**2**	**3**	**4**	**5**	**6**	**7**		
23.37.19.90	ID = 16	FSICONCT	ASID 0015		FSMCONCT	R14-R1 = 80BBB3F4...		
RECORD 2								
9F1DF8	00000040	000000FE	00010000	00000000	00000000	00000000	00000080	009F1E38
RECORD 3								
9F1E18	009F1B90	00000002	0000C074	D1C5E2F2	00000000	00000000	00000000	00000000

- Record 1:

  Field **Contents**

  1 Time-of-day clock value when the $TRACE was executed.

  2 Trace identifier

  3 Functional subsystem interface function name.

  4 Address space identifier associated with the functional subsystem.

  5 Device name associated with the functional subsystem. This column is blank for this record. The column is separated from the previous column by two spaces and is eight characters long.

  6 Internal label associated with this trace entry.

  7 Contents of registers 14, 15, 0, and 1.
- Records 2 and 3:

  Contain the trace FSI parameter, flags and miscellaneous data areas.

Trace ID=17 sample

Trace identifier 17 (CKPTPERF) provides information about checkpoint performance. The formatted output records appear in groups of at most five records unless the installation added checkpoint table entries (CTENT) to the checkpoint. The first field in the first record of each group contains one of the following values, and determines the meaning of the contents of the rest of the fields:

READ 1

This 1-record group describes information about the first read for this checkpoint cycle.

READ 2
This record group describes information about the second read for this checkpoint cycle.

PRIMARY
This record group describes information about the primary write for this checkpoint cycle.

INTERMED
This record group describes information about the intermediate write for this checkpoint cycle.

FINAL
This record group describes information about the final write for this checkpoint cycle.

Figure 14 is a sample of output from trace identifier 17.

```
12.10.17.99  ID =  17 CKPTPERF            CKPT      05B420F8
                   READ1    0000006F 00000001 0000001E 00000064 000001F4 00000004 CKPT1
12.10.17.99  ID =  17 CKPTPERF            CKPT      05B420F8
                   READ2    00000000 00000004 00000014 000000BC 00000002 000000A4 000000A4
                   000009FC 000097E1 00000009
                   00008040
                   00000000 00000003 00000000 00000000 00000000 00000002 00000000 00000003
                   00000000 00000001 00000000 00000001 00000000 00000004 00000000 00000001
                   00000000 00000005 00000000 00000000 00000000 00000000 00000000 00000000
12.10.18.01  ID =  17 CKPTPERF            CKPT      05B420F8
                   PRIMARY  00000109 00000000 00000000 000000BC 00000001 00000103 00000103
                   00000000          00000000 00000009 00002111 CKPT2
                   00000000 00000000 00000000
                   00000001 00000000 00000000 00000000 00000001 00000000 00000001 00000000
                   00000001 00000000 00000001 00000000 00000001 00000000 00000001 00000000
                   00000002 00000000 00000000 00000000 00000000 00000000 00000000 00000000
12.10.18.16  ID =  17 CKPTPERF            CKPT      05B420F8
                   INTERMED 0000003C 00000001 00000009 000000BC 00000001 0000068F 0000068F
                   00000743          00000001 00000005 00002112 CKPT2
                   0000001A 00026852 00000266
                   00000000 00000002 00000000 00000000 00000000 00000000 00000000 00000001
                   00000000 00000000 00000000 00000000 00000000 00000004 00000000 00000001
                   00000000 00000001 00000000 00000000 00000000 00000000 00000000 00000000
12.10.18.32  ID =  17 CKPTPERF            CKPT      05B420F8
                   FINAL    00000033 00000001 00000003 000000BC 00000000 00000000 00000000
                   00000803 00000CF3          00000001 00000005 00002113 CKPT2
                   00000024 0004739A 00000987
                   00000000 00000002 00000000 00000000 00000000 00000000 00000000 00000002
                   00000000 00000000 00000000 00000001 00000000 00000004 00000000 00000001
                   00000000 00000002 00000000 00000000 00000000 00000000 00000000 00000000
```

Figure 14. Sample Output from Trace Identifier 17

Trace record contents

Table 13 is the sample record broken into fields which are described after the table.

Table 13. Fields for Trace 17

RECORD 1							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
12.10.17.99	ID = 17	CKPTPERF		CKPT	05B420F8		
RECORD 2 - 5							
1	2	3	4	5	6	7	8
READ2	00000000	00000004	00000014	000000BC	00000002	000000A4	000000A4
000009FC	000097E1	00000009					
00008040							

Table 13. Fields for Trace 17 (continued)

RECORD 1							
00000000	00000003	00000000	00000000	00000000	00000002	00000000	00000003
00000000	00000001	00000000	00000001	00000000	00000004	00000000	00000001

- Record 1:

  Field **Contents**

  1 Time-of-day clock value when the $TRACE was processed.

  2 Trace identifier.

  3 Function name.

  4 Eight blanks.

  5 $PCE name associated with this trace.

  6 $PCE address.

  7 Eight blanks.

  8 Eight blanks.
- Records 2 through 5, depending on the checkpoint cycle:

Table 14. Description of Trace Identifier 17 READ1 Record 1 by Checkpoint Cycle

Cycle Name	Record	Field	Description
READ1	1	1	Contains the characters “READ1”.
		2	Contains the time (in tenths of milliseconds) that passed from the $EXCP macro that began the READ1 I/O until the I/O completed and the checkpoint $PCE got dispatched.
		3	Contains the number of used pages in the change log.
		4	Contains the current value of MINHOLD.
		5	Contains the current value of MINDORM.
		6	Contains the current value of MAXDORM.
		7	Contains the number of change log records read in READ1.
		8	Contains the name of the data set that contained the current copy of the queues when READ1 was performed. This field contains either “CKPT1” or “CKPT2”.

Table 15. Description of Trace Identifier 17 READ2 Record 1 by Checkpoint Cycle

Cycle Name	Record	Field	Description
READ2	1	1	Contains the characters “READ2”.
		2	Contains the time (in tenths of milliseconds) that passed from the $EXCP that began the READ2 I/O until the I/O completed and the checkpoint $PCE got dispatched.
		3	Contains the total number of pages in the change log.
		4	Contains the number of control blocks in the change log.
		5	Contains the number of $PCEs defined to this member.
		6	Contains the number of $PCEs waiting for access to the checkpoint.
		7	Contains the maximum length of time (in tenths of milliseconds) that a $PCE was waiting for access to the checkpoint.
		8	Contains the average length of time (in tenths of milliseconds) that the $PCEs were waiting for access to the checkpoint.

Table 16. Description of Trace Identifier 17 READ2 Record 2 by Checkpoint Cycle

Cycle Name	Record	Field	Description
READ2	2	1	Contains the number of used bytes in the change log.
		2	Contains the length of time (in tenths of milliseconds) that this member did not hold the checkpoint.
		3	Contains the number of pages which would have been read if the complex had been in duplex mode. This field may be low if the change log overflows. The field is meaningless if the complex is in duplex mode.
		4	Contains the name of the data set READ2 was performed against. This field will contain either “CKPT1” or “CKPT2”.

Table 17. Description of Trace Identifier 17 READ2 Record 3 by Checkpoint Cycle

Cycle Name	Record	Field	Description
READ2	3	1	Contains the total length of PCE wait time (in microseconds).

Table 18. Description of Trace Identifier 17 READ2 Record 4 by Checkpoint Cycle

Cycle Name	Record	Field	Description
READ2	4	1	Contains the number of pages read for the first CTENT
		2	Contains the number of control blocks in the change log for the first CTENT.
		3	Contains the number of pages read for the second CTENT.
		4	Contains the number of control blocks in the change log for the second CTENT.
		5	Contains the number of pages read for the third CTENT.
		6	Contains the number of control blocks in the change log for the third CTENT.
		7	Contains the number of pages read for the fourth CTENT.
		8	Contains the number of control blocks in the change log for the fourth CTENT.
	5 - end		These records have the same format as record 4, but for the fifth through last CTENTs.

Table 19. Description of Trace Identifier 17 PRIMARY WRITE Record 1 by Checkpoint Cycle

Cycle Name	Record	Field	Description
PRIMARY WRITE	1	1	Contains the characters “PRIMARY”.
		2	Contains the time (in tenths of milliseconds) that passed from the $EXCP which started the primary write I/O until the I/O completed and the checkpoint $PCE was dispatched.
		3	Contains the number of used pages in the change log.
		4	Contains the number of control blocks in the change log.
		5	Contains the number of $PCEs defined to this member.
		6	Contains the number of $PCEs that are waiting for the checkpoint write to complete.
		7	Contains the maximum length of time (in tenths of milliseconds) that a $PCE was waiting for the checkpoint write to complete.
		8	Contains the average length of time (in tenths of milliseconds) that the $PCEs were waiting for the checkpoint write to complete.

Table 20. Description of Trace Identifier 17 PRIMARY WRITE Record 2 by Checkpoint Cycle

Cycle Name	Record	Field	Description
PRIMARY WRITE	2	1	Contains the number of used bytes in the change log.
		2	Contains the characters “PRIO AGE” if priority aging contributed to this write.
		3	Contains the number of times the checkpoint $PCE put itself at the bottom of the ready queue before performing this write.
		4	Contains the number of pages that would have been written if the complex had been in duplex mode. This field may be low if the change log overflows. This field is meaningless if the complex is in duplex mode.
		5	Contains the level number of the data set.
		6	Contains the name of the data set the primary write was performed against. This field will contain either “CKPT1” or “CKPT2”.

Table 21. Description of Trace Identifier 17 PRIMARY WRITE Record 3 by Checkpoint Cycle

Cycle Name	Record	Field	Description
PRIMARY WRITE	3	1	Contains the number of $CKPTs issued during the checkpoint cycle.
		2	Contains the length of MVS wait time (in microseconds) during this checkpoint cycle.
		3	Contains the length of $QSUSE time (in microseconds) during this checkpoint cycle.

Table 22. Description of Trace Identifier 17 PRIMARY WRITE Record 4 by Checkpoint Cycle

Cycle Name	Record	Field	Description
PRIMARY WRITE 4	4	1	Contains the number of pages written for the first CTENT.
		2	Contains the number of control blocks in the change log for the first CTENT.
		3	Contains the number of pages written for the second CTENT.
		4	Contains the number of control blocks in the change log for the second CTENT.
		5	Contains the number of pages written for the third CTENT.
		6	Contains the number of control blocks in the change log for the third CTENT.
		7	Contains the number of pages written for the fourth CTENT.
		8	Contains the number of control blocks in the change log for the fourth CTENT.
	5 - end		These records have the same format as Record 4, but for the fifth through last CTENTs.

Table 23. Description of Trace Identifier 17 INTERMEDIATE WRITE Record 1 by Checkpoint Cycle

Cycle Name	Record	Field	Description
INTERMEDIATE WRITE	1	1	Contains the characters "INTERMED".
		2	Contains the time (in tenths of milliseconds) that passed from the $EXCP macro that started the intermediate write I/O until the I/O completed and the checkpoint $PCE got dispatched.
		3	Contains the number of used pages in the change log.
		4	Contains the number of control blocks in the change log.
		5	Contains the number of $PCEs defined to this member.
		6	Contains the number of $PCEs that are waiting for the checkpoint write to complete.
		7	Contains the maximum length of time (in tenths of milliseconds) that a $PCE was waiting for the checkpoint write to complete.
		8	Contains the average length of time (in tenths of milliseconds) that the $PCEs were waiting for the checkpoint write to complete.

Table 24. Description of Trace Identifier 17 INTERMEDIATE WRITE Record 2 by Checkpoint Cycle

Cycle Name	Record	Field	Description
INTERMEDIATE WRITE	2	1	Contains the number of used bytes in the change log.
		2	Contains the characters "PRIO AGE" if priority aging contributed to this write.
		3	Contains the number of times the checkpoint $PCE put itself at the bottom of the ready queue before performing this write.
		4	Contains the number of pages that would have been written if the complex had been in duplex mode. This field may be low if the change log overflows. This field is meaningless if the complex is in duplex mode.
		5	Contains the level number of the data set.
		6	Contains the name of the data set the intermediate write was performed against. This field will contain either "CKPT1" or "CKPT2".

Table 25. Description of Trace Identifier 17 INTERMEDIATE WRITE Record 3 by Checkpoint Cycle

Cycle Name	Record	Field	Description
INTERMEDIATE WRITE	3	1	Contains the number of $CKPTs issued during this checkpoint cycle.
		2	Contains the length of MVS wait time (in microseconds) during this checkpoint cycle.
		3	Contains the length of $QSUSE time (in microseconds) during this checkpoint cycle.

Table 26. Description of Trace Identifier 17 INTERMEDIATE WRITE Record 4 by Checkpoint Cycle

Cycle Name	Record	Field	Description
INTERMEDIATE WRITE	4	1	Contains the number of pages written for the first CTENT.
		2	Contains the number of control blocks in the change log for the first CTENT.
		3	Contains the number of pages written for the second CTENT.
		4	Contains the number of control blocks in the change log for the second CTENT.
		5	Contains the number of pages written for the third CTENT.
		6	Contains the number of control blocks in the change log for the third CTENT.
		7	Contains the number of pages written for the fourth CTENT.
		8	Contains the number of control blocks in the change log for the fourth CTENT.
	5 - end		These records have the same format as record 4, but for the fifth through last CTENTs.

Table 27. Description of Trace Identifier 17 FINAL WRITE Record 1 by Checkpoint Cycle

Cycle Name	Record	Field	Description
FINAL WRITE	1	1	Contains the characters “FINAL”.
		2	Contains the time (in tenths of milliseconds) that passed from the $EXCP macro that started the final write I/O until the I/O completed and the checkpoint $PCE got dispatched.
		3	Contains the number of used pages in the change log.
		4	Contains the number of control blocks in the change log.
		5	Contains the number of $PCEs defined to this member.
		6	Contains the number of $PCEs that are waiting for the checkpoint write to complete.
		7	Contains the maximum length of time (in tenths of milliseconds) that a $PCE was waiting for the checkpoint write to complete.
		8	Contains the average length of time (in tenths of milliseconds) that the $PCEs were waiting for the checkpoint write to complete.

Table 28. Description of Trace Identifier 17 FINAL WRITE Record 2 by Checkpoint Cycle

Cycle Name	Record	Field	Description
FINAL WRITE	2	1	Contains the number of used bytes in the change log.
		2	Contains the length of time (in tenths of milliseconds) that this member held the checkpoint.
		3	Contains the characters “PRIO AGE” if priority aging contributed to this write.
		4	Contains the number of times the checkpoint $PCE put itself at the bottom of the ready queue before performing this write.
		5	Contains the number of pages that would have been written if the complex had been in duplex mode. The field may be low if the change log overflows. This field is meaningless if the complex is in duplex mode.
		6	Contains the level number of the data set.
		7	Contains the name of the data set the final write was performed against. This field will contain either “CKPT1” or “CKPT2”.

Table 29. Description of Trace Identifier 17 FINAL WRITE Record 3 by Checkpoint Cycle

Cycle Name	Record	Field	Description
FINAL WRITE	3	1	Contains the number of $CKPTs issued during this checkpoint cycle.
		2	Contains the length of MVS wait time (in microseconds) during this checkpoint cycle.
		3	Contains the length of $QSUSE time (in microseconds) during this checkpoint cycle.

Table 30. Description of Trace Identifier 17 FINAL WRITE Record 4 by Checkpoint Cycle

Cycle Name	Record	Field	Description
FINAL WRITE	4	1	Contains the number of pages written for the first CTENT.
		2	Contains the number of control blocks in the change log for the first CTENT.
		3	Contains the number of pages written for the second CTENT.
		4	Contains the number of control blocks in the change log for the second CTENT.
		5	Contains the number of pages written for the third CTENT.
		6	Contains the number of control blocks in the change log for the third CTENT.
		7	Contains the number of pages written for the fourth CTENT.
		8	Contains the number of control blocks in the change log for the fourth CTENT.
	5 - end		These records have the same format as record 4, but for the fifth through last CTENTs.

Trace ID=20 sample

Figure 15 on page 50 is a sample of output from trace identifier 20. Trace identifier 20 traces all $#GET macro calls made by devices, including local and remote print and punch devices, spool offload SYSOUT transmitters, and NJE SYSOUT

transmitters. The trace provides counts such as the number of elements searched before work is found, the total number of elements, the number of elements in use, and the work selection list. The trace is also generated for SYSOUT Application Programming Interface (SAPI) device calls. It provides the job number, class, and route code of the output selected.

For NJE transmitters the trace also displays the count of job output elements ($JOE) on the chain and the count of those the transmitter selected.

This record provides a means to analyze selection criteria and queue search overhead to tune work selection criteria.

```
 14.27.35.94 ID = 20 $#GET STC00122 PRT1         $#GET CALL FOR PRT1
               WS = (W,Q,R,PRM,LIM/F,UCS,FCB)
               OUTGRPS DEFINED = 200 OUTGRPS IN USE = 2
               OUTGRPS SCANNED = 1 OUTGRPS THRU WS = 1
               OUTGRP MASK = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFF
               CLASS = Z ROUTE = 00010000 FLAGS = 20A00000
               ELEMENT SELECTED = 1
               CPU TIME USED (SEC) = 0.000024
               $#GET CALLED BY = HASPPRPU 000A6B20 + 0009F2
14.27.39.07 ID = 20 $#GET PRT1 06461390          $#GET CALL FOR PRT1
              WS = (W,Q,R,PRM,LIM/F,UCS,FCB)
              OUTGRPS DEFINED = 200 OUTGRPS IN USE = 2
              OUTGRPS SCANNED = 0 OUTGRPS THRU WS = 0
              FLAGS = 20A02000
              CPU TIME USED (SEC) = 0.000005
              $#GET CALLED BY = HASPPRPU 000A6B20 + 0009F2
14.27.39.07 ID = 20 $#GET PRT1 06461390                      $#GET CALL FOR PRT1
              WS = (W,Q,R,PRM,LIM/F,UCS,FCB)
              OUTGRPS DEFINED = 200 OUTGRPS IN USE = 2
              OUTGRPS SCANNED = 0 OUTGRPS THRU WS = 0
              FLAGS = 2080A000
              FAST EXIT INDICATOR SET - FAST EXIT SUCCESSFUL
              CPU TIME USED (SEC) = 0.000003
              $#GET CALLED BY = HASPPRPU 000A6B20 + 000A76
14.28.59.55 ID = 20 $#GET STC00123 PRT2 06463AB0       $#GET CALL FOR PRT2
              WS = (W,Q,R,PRM,LIM/F,UCS,FCB)
              OUTGRPS DEFINED = 200 OUTGRPS IN USE = 2
              OUTGRPS SCANNED = 1 OUTGRPS THRU WS = 1
              OUTGRP MASK = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFF
              CLASS = A ROUTE = 00010000 FLAGS = 20A08000
              FAST EXIT INDICATOR SET - FAST EXIT SUCCESSFUL
              ELEMENT SELECTED = 1
              CPU TIME USED (SEC) = 0.000011
              $#GET CALLED BY = HASPPRPU 000A6B20 + 0009F2
14.28.59.59 ID = 20 $#GET PRT2 06463AB0                              $#GET CALL FOR PRT2
              WS = (W,Q,R,PRM,LIM/F,UCS,FCB)
              OUTGRPS DEFINED = 200 OUTGRPS IN USE = 0
              OUTGRPS SCANNED = 0 OUTGRPS THRU WS = 0
              FLAGS = 20A0A000
              FAST EXIT INDICATOR SET - FAST EXIT SUCCESSFUL
              CPU TIME USED (SEC) = 0.000004
              $#GET CALLED BY = HASPPRPU 000A6B20 + 0009F2
```

Figure 15. Sample Output from Trace Identifier 20

Trace record contents

Table 31 is the sample record broken into fields which are described after the table.

Table 31. Fields for Trace 20

RECORD 1							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
11.40.09.43	ID = 20	$#GET	JOB00004		PRT2		$#GET CALL FOR PRT2

Table 31. Fields for Trace 20 (continued)

RECORD 1
RECORD 2
WS = (W,R,Q,PRM,LIM,P/F,UCS,FCB)
RECORD 3
OUTGRPS DEFINED = 1250 OUTGRPS IN USE = 2
RECORD 4
OUTGRPS SCANNED = 1 OUTGRPS THRU WS = 1
RECORD 5
OUTGRP MASK = FFFFFAFF FF0900FF FFFFFFFF FFFFFFFF FFFF
RECORD 6
CLASS = A ROUTE = 00010000 FLAGS = 20A40000
RECORD 7
FAST EXIT INDICATOR SET
RECORD 8
ELEMENT SELECTED = 1
RECORD 9
CPU TIME USED (SEC) = 0.000100
RECORD 10
$#GET CALLED BY = HASPPRPU 00067B30 + 0008C0

The trace output contains the following:

- Record 1:

Field	Contents
1	Time-of-day clock value when the $TRACE was processed.
2	Trace identifier.
3	Function.
4	Job name, if available.
5	Three blanks.
6	Device name.
7	User identification associated with the job, if available.
8	Description of trace.

- Record 2:

  The WS= field contains the work selection list for the device.

- Record 3:

  OUTGRPS DEFINED - the total number of output groups. OUTGRPS IN USE - the number of output groups in use.

- Record 4:

  OUTGRPS SCANNED - the number of output groups searched before work is found. OUTGRPS THRU WS - the number of output groups actually compared to the work selection parameter list for this device.

- Record 5:

OUTGRP MASK - the mask of the $JOE the device selected.

Note: The mask is meaningless if CTOKEN is a selection criteria in the work selection list (Record 2 specifies the work selection list).

- Record 6:

  CLASS - the class of the $JOE the device selected. ROUTE - the route code for the $JOE the device selected. An 8-character user identification, if available. If not available, this field is blanks. FLAGS - characteristics of the $#GET call and return codes. The $GTW mapping macro documents this flag.

- Record 7:

  The following information is only present when the indicator (a flag bit in the $GTW work area) is set:

  – FAST EXIT INDICATOR SET - indicates that JES2 took a fast exit from the $#GET processing.
  – FAST EXIT SUCCESSFUL - indicates either the element was selected without doing any queue scanning or JES2 determined that no elements were available for selection without doing any queue scanning.
  – FAST EXIT UNSUCCESSFUL - indicates that some sort of problem occurred and IBM JES2 service should be contacted.

- Record 8:

  ELEMENT SELECTED - the element selected from the output groups that were scanned.

- Record 9:

  CPU TIME USED (SEC) - the amount of CPU time spent doing processing for the $#GET macro.

- Record 10:

  $#GET CALLED BY - the caller of the $#GET macro and the offset where the call was made.

If the trace output represents an offload SYSOUT transmitter, the trace also contains:

- JOBS DEFINED - total number of jobs.
- JOBS SCANNED - the number of jobs searched before work is found.
- JOBS THRU WS instead of OUTGRPS THRU WS - the number of jobs that matched the work selection list.
- JOB MASK instead of OUTGRP MASK for offload devices that can select held output.

Note:

1. If an offload device can select held output, the mask describes the $JQE for the job and the output includes the count of $JOEs for the job.
2. If an offload device can only select held output, and no work is found, the trace omits the job information from the output.
3. If an offload device can select non-held output, but does not select a $JQE, the information in the output describes the $JOE selected.

Trace ID=21 sample

Figure 16 on page 53 is a sample of output from trace identifier 21. This trace is active when:

- The initialization statement that defines an adjacent node has TRACE=YES specified
- The $T LINE(nnnn),TR=Yes command is specified
- The $T NODE(nnnn),TR=Yes command is specified.

These records trace the interchange between the local node and another node when the 2 nodes are establishing a session. See *z/OS MVS System Messages, Vol 5 (EDG-GFS)* for more information about the network connection control (NCC) records.

This trace is only enabled for printers that have tracing turned on by the $T PRT(nnnn),TR=Yes command.

```
16.27.00.38  ID =  21 NPMSGNON            NPM       02933B90
             NCC=I  RECV FROM NODE2   (01) VIA LINE13
             CES=90/054,015:38:34 REST=  100
16.27.00.40  ID =  21 NPMSGNON            NPM       02933B90
             NCC=J  SENT TO   NODE2   (01) VIA LINE13
             CES=90/054,015:39:34 REST=  100
16.27.00.80  ID =  21 NPMSGNON            NPM       02933B90
             NCC=L  RECV FROM NODE2   (01) VIA LINE13
             CES=90/054,015:39:34 REST=  100
```

Figure 16. Sample Output from Trace Identifier 21

Trace record contents

Table 32 is the sample record broken into fields which are described after the table.

Table 32. Fields for Trace 21

RECORD 1							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
16.27.00.38	ID = 21	NPMSGNON		NPM		02933B90	
RECORD 2							
NCC=I RECV FROM NODE2 (01) VIA LINE13							
RECORD 3							
CES=90/054,015:38:34 REST= 100							

The trace has the following format:

- Record 1:

Field	Contents
1	Time-of-day clock value when the $TRACE was executed.
2	Trace identifier.
3	Function.
4	Twelve blanks.
5	Name of the $PCE. This field contains either NPM or CKPT for this record.
6	Eight blanks.
7	$PCE address.

- Record 2:

- NCC - the type of network communication control (NCC) record traced. Values for NCC are:
  - I is a signon record.
  - J is a response signon record.
  - K is a reset record.
  - L is a concurrence record.
- RECV FROM or SENT TO - whether the record is coming from a node or going to a node.
- The name of the node.
- The member of the node connected with this node. The member is enclosed in parenthesis.
- VIA LINE - names the line number through which the record traveled. VIA MLINE - if the record is sent from a member of this multi-access spool configuration. N/A - if this is not a path manager connection.

• Record 3:
- CES - is the connection event sequence associated with the NCC record. The value of this field has the form:

  ```
  YYDDD,0HH:MM:SS
  ```

  where:

  YY is the last 2 digits of the current year.

  DDD is the day of the year.

  HH:MM:SS is the time of the day, in 24-hour format.
- REST - is the resistance of the connection.

Trace ID=22 sample

Figure 17 is a sample output from trace identifier 22. JES2 creates this record to trace add NCC records, spooled nodes attached table entries, $ADD CONNECT commands, and $T CONNECT commands. This trace is active when:

• The initialization statement that defines an adjacent node has TRACE=YES specified

• The $T LINE(nnnn),TR=Yes command is specified

• The $T NODE(nnnn),TR=Yes command is specified.

See *z/OS MVS System Messages, Vol 5 (EDG-GFS)* for more information about the NCC records.

```
16.27.00.80  ID =  22 NPMADD              NPM       02933B90
             NCC=M  SENT TO   NODE2   (01) VIA LINE13
             CONNECTS    NODE1   (01) AND NODE3   (01) REST=  200
             CES=90/054,015:39:34 STATUS=ACTIVE
16.27.00.80  ID =  22 NPMADD              NPM       02933B90
             NCC=M  SENT TO   NODE2   (01) VIA LINE13
             CONNECTS    NODE3   (01) AND NODE4   (01) REST=  200
             CES=90/054,014:12:42 STATUS=ACTIVE
16.27.00.80  ID =  22 NPMADD              NPM       02933B90
             NCC=M  SENT TO   NODE2   (01) VIA LINE13
             CONNECTS    NODE1   (01) AND NODE4   (01) REST=  200
             CES=90/054,015:38:37 STATUS=ACTIVE
```

Figure 17. Sample Output from Trace Identifier 22

Trace record contents

Table 33 is the sample record broken into fields which are described after the table.

Table 33. Fields for Trace 22

RECORD 1							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
16.27.00.80	ID = 22	NPMADD		NPM		02933B90	
RECORD 2							
NCC=M SENT TO NODE2 (01) VIA LINE13							
RECORD 3							
CONNECTS NODE1 (01) AND NODE3 (01) REST= 200							
RECORD 4							
CES=90/054,015:39:34 STATUS=ACTIVE							

The trace has the following format:

- Record 1 has the same format as trace identifier 21.
- Record 2:
  - NCC - the type of network communication control (NCC) record traced. This value is always M for this record.
  - RECV FROM or SENT TO - whether the record is coming from a node or going to a node.
  - The name of the node.
  - The member of the node connected with this node. The member is enclosed in parenthesis.
  - VIA - indicates where this record originated from or was sent. Values for this field are:

    LINEnnnn
    The line to which JES2 sent this record or from which JES2 received this record.

    MLINEn
    The member of the MAS at this node to which this member sent the record or from which this member received the record.

    console identifier
    The console from which the operator entered the $ADD, $DEL, or $T CONNECT command.
- Record 3:
  - CONNECTS - contains the node names and member numbers on either side of the connection that this add record describes. The member number appears in parenthesis.
  - REST - the resistance of the connection.
- Record 4:
  - CES - the connection event sequence as described for trace identifier 21. This field is not available if the connection is a static connection.
  - STATUS - the current status of the connection this record represents. This value can either be ACTIVE or HELD.
- Record 5:
  - OLDCES - the connection event sequence associated with an existing NAT, which this record is replacing.

– OLDSTATUS - the status of this connection before this record was created. This value can be either ACTIVE, INACTIVE, HELD, or NONE.

Note: This record is not created when receiving a record or when the connection is a static connection.

- Record 6:

  FULLPATH determines whether JES2 updates the nodes attached table using the information from this record. This value can be either YES or NO.

Trace ID=23 sample

Figure 18 is a sample output from trace identifier 23. JES2 creates this record to trace subtract NCC records, spooled nodes attached table entries, disconnects generated by JES2 when it updates the nodes attached table (NAT), and $DEL CONNECT commands. This trace is active when:

- The initialization statement that defines an adjacent node has TRACE=YES specified
- The $T LINE(nnnn),TR=Yes command is specified
- The $T NODE(nnnn),TR=Yes command is specified.

See *z/OS MVS System Messages, Vol 5 (EDG-GFS)* for more information about the NCC records.

```
20.03.27.08  ID =  23 NPMSUB              NPM       02934B90
             NCC=N  SENT TO   NODE3   (01) VIA LINE10
             DISCONNECTS NODE1   (01) AND NODE4   (01) REST=  200
             CES=90/054,019:51:28 STATUS=INACTIVE
```

Figure 18. Sample Output from Trace Identifier 23

Trace record contents

Table 34 is the sample record broken into fields which are described after the table.

Table 34. Fields for Trace 23

RECORD 1							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
20.03.27.08	ID = 23	NPMSUB		NPM		02934B90	
RECORD 2							
NCC=N SENT TO NODE3 (01) VIA LINE10							
RECORD 3							
DISCONNECTS NODE1 (01) AND NODE4 (01) REST= 200							
RECORD 4							
CES=90/054,019:51:28 STATUS=INACTIVE							
RECORD 5							

The trace has the following format:

- Record 1 has the same format as trace identifier 21.
- Record 2:
  – NCC - the type of network communication control (NCC) record traced. This value is always N for this record.

- RECV FROM or SENT TO - whether the record is coming from a node or going to a node.
- The name of the node.
- The member of the node connected with this node. The member is enclosed in parenthesis.
- VIA - indicates where this record originated from or was sent. Values for this field are:

  LINEnnnn
  The line to which JES2 sent this record or from which JES2 received this record.

  MLINEn
  The member of the MAS at this node to which this member sent the record or from which this member received the record.

  console identifier
  The console from which the operator entered the $DEL CONNECT.

  FULLPATH
  JES2 determined it could no longer reach this node and disconnected the node.

- Record 3:
  - DISCONNECTS - contains the node names and member numbers on either side of the connection that this subtract record describes. The member number appears in parenthesis.
  - REST - the resistance of the connection.
- Record 4:
  - CES - the connection event sequence as described for trace identifier 21. This field is not available if the connection is a static connection.
  - STATUS - the current status of the connection this record represents. This value can be INACTIVE or HELD.
- Record 5:
  - OLDCES - the connection event sequence associated with an existing NAT that this record is replacing.
  - OLDSTATUS - the status of this connection before this record was created. This value can be either ACTIVE, INACTIVE, HELD, or NONE.

  Note: This record is not created when the connection is a static connection.

Trace ID=24 sample

Figure 19 on page 58 is a sample output from trace identifier 24. JES2 creates this record when it rejects:

- Add or subtract NCC records
- Spooled nodes attached table entries
- $ADD, $T, or $DEL CONNECT commands.

This trace is active when:

- The initialization statement that defines an adjacent node has TRACE=YES specified
- The $T LINE(nnnn),TR=Yes command is specified
- The $T NODE(nnnn),TR=Yes command is specified.

```
16.27.00.80  ID =  24 NPMERR              NPM       02933B90
             NCC=M  RECV FROM NODE2   (01) VIA LINE13
             CONNECTS    NODE1   (01) AND NODE3   (01) REST=  200
             CES=90/054,015:39:34 STATUS=ACTIVE
             OLDCES=90/054,015:39:34 OLDSTATUS=ACTIVE
             IGNORED BECAUSE:MORE RECENT CONNECT EXISTS
16.27.00.80  ID =  24 NPMERR              NPM       02933B90
             NCC=M  RECV FROM NODE2   (01) VIA LINE13
             CONNECTS    NODE2   (01) AND NODE3   (01) REST=  200
             CES=90/054,015:35:56 STATUS=ACTIVE
             OLDCES=90/054,015:35:56 OLDSTATUS=ACTIVE
             IGNORED BECAUSE:MORE RECENT CONNECT EXISTS
16.27.00.80  ID =  24 NPMERR              NPM       02933B90
             NCC=M  RECV FROM NODE2   (01) VIA LINE13
             CONNECTS    NODE1   (01) AND NODE2   (01) REST=  200
             CES=90/054,015:53:58 STATUS=ACTIVE
             OLDCES=90/054,015:53:58 OLDSTATUS=ACTIVE
             IGNORED BECAUSE:MORE RECENT CONNECT EXISTS
16.27.00.91  ID =  24 NPMERR              NPM       02934B90
             NCC=M  RECV FROM NODE4   (01) VIA LINE13
             CONNECTS    NODE1   (01) AND NODE3   (01) REST=  200
             CES=90/054,015:39:34 STATUS=ACTIVE
             OLDCES=90/054,015:39:34 OLDSTATUS=ACTIVE
             IGNORED BECAUSE:MORE RECENT CONNECT EXISTS
16.27.00.91  ID =  24 NPMERR              NPM       02934B90
             NCC=M  RECV FROM NODE4   (01) VIA LINE13
             CONNECTS    NODE3   (01) AND NODE4   (01) REST=  200
             CES=90/054,014:12:42 STATUS=ACTIVE
             OLDCES=90/054,015:37:21 OLDSTATUS=ACTIVE
             IGNORED BECAUSE:EVENT SEQUENCE (CES) NOT VALID
16.27.00.91  ID =  24 NPMERR              NPM       02934B90
             NCC=M  RECV FROM NODE4   (01) VIA LINE13
             CONNECTS    NODE1   (01) AND NODE4   (01) REST=  200
             CES=90/054,015:38:37 STATUS=ACTIVE
             OLDCES=90/054,015:38:37 OLDSTATUS=ACTIVE
             IGNORED BECAUSE:MORE RECENT CONNECT EXISTS
```

Figure 19. Sample Output from Trace Identifier 24

Trace record contents

Table 35 is the sample record broken into fields which are described after the table.

Table 35. Fields for Trace 24

RECORD 1							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
16.27.00.80	ID = 24	NPMERR		NPM		02933B90	
RECORD 2							
NCC=M RECV FROM NODE2 (01) VIA LINE13							
RECORD 3							
CONNECTS NODE1 (01) AND NODE3 (01) REST= 200							
RECORD 4							
CES=90/054,015:39:34 STATUS=ACTIVE							
RECORD 5							
OLDCES=90/054,015:39:34 OLDSTATUS=ACTIVE							
RECORD 6							
IGNORED BECAUSE:MORE RECENT CONNECT EXISTS							

The trace has the following format:

- Record 1 has the same format as trace identifier 21.
- Record 2:
  - NCC - the type of network communication control (NCC) record traced. NCC can have a value of:
    - I is a signon record.
    - J is a response signon record.
    - K is a reset record.
    - L is a concurrence record.
    - M is an add NCC record.
    - N is a subtract NCC record.
  - RECV FROM or SENT TO - whether the record is coming from a node or going to a node.
  - The name of the node.
  - The member of the node connected with this node. The member is enclosed in parenthesis.
  - VIA - indicates where this record originated or was sent. Values for this field are:

    LINEnnnn
    The line to which JES2 sent this record or from which JES2 received this record.

    SPOOL
    The record was sent to or received from another member of the MAS at this node through spool communications.

    console identifier
    The console from which the operator issued the $ADD, $DEL, or $T CONNECT command.
- Record 3:
  - CONNECTS | DISCONNECTS - contains the node names and member numbers on either side of the connection that this add or subtract record describes. The member number appears in parenthesis.
  - REST - the resistance of the connection.
- Record 4:
  - CES - the connection event sequence as described for trace identifier 21.
  - STATUS - the current status of the connection this record represents. This value can be ACTIVE, INACTIVE, or HELD.
- Record 5:
  - OLDCES - the connection event sequence associated with an existing NAT which this record is replacing.
  - OLDSTATUS - the status of this connection before this record was created. This value can be either ACTIVE, INACTIVE, HELD, or NONE.
- Record 6:

  IGNORED BECAUSE: explains briefly why JES2 rejected the record. Values for this are:
  - REASON UNKNOWN
  - NODE NAME NOT RECOGNIZED
  - MEMBER NUMBER NOT VALID
  - STORAGE NOT AVAILABLE
  - RESISTANCE NOT VALID
  - EVENT SEQUENCE (CES) NOT VALID

- NO DEVICES AVAILABLE
- TOD TOLERANCE EXCEEDED
- LINE PASSWORD NOT VALID
- NODE PASSWORD NOT VALID
- LINE NOT TRANSPARENT
- LINE ALREADY ACTIVE
- NCC RECORD TYPE UNRECOGNIZED
- ABEND PROCESSING RECORD
- MORE RECENT CONNECT EXISTS
- CONNECTION INCLUDES LOCAL SYSTEM
- UNSUPPORTED NJE FEATURE FLAGS
- INCORRECT VALUE FOR PATHMGR
- NON-PATH MANAGER CES
- MULTIPLE RECORDS IN SIGNON BUFFER
- OLD SUBTRACT RECORD RECEIVED
- RECEIVED FROM NON-PATH MANAGER NODE
- LINE ON WHICH RECORD RECEIVED NO LONGER ACTIVE
- DUPLICATE PRIMARY-SECONDARY NODES AND MEMBERS
- INCORRECT MULTI TRUNK PROTOCOL
- RECORD HAS DUPLICATE CES BUT UNIQUE RESISTANCE

Trace ID=25 sample

Figure 20 on page 61 is a sample of output from trace identifier 25. This trace gathers information from the following control blocks about the functional subsystem (FSS) checkpoint: FSACB, FSSCB, JIB (including WORK JOE, CHAR JOE, JSPA, JOX, SYSTEM BERT and USER BERT) GCB, PDDB, IAZFSIP AND IAZCHK(JES).

You can look at control blocks that SDSF uses to display information about a data set being printed on a FSS printer. For example, the CHKPAGE and CHKREC fields in the IAZCHK control block contain information about what the function subsystem is passing to JES2 to write to spool. This information would be useful to help determine why incorrect record counts are displayed on SDSF.

For a description of the control blocks and their fields see: *z/OS MVS Data Areas* in the z/OS Internet Library: http://www.ibm.com/systems/z/os/zos/bkserv/.

```
18.10.20.04472 ID= 25 FSMCHKPT ASID 0028  PRT3      004D6B70  R14-R1 = 00FE02CE 804CFF24 7F4F96A0 7F506A60  ------------
  FSACB
  97A506A  C6E2C140 00040001 097CC0E8 00000000   7F4F96C8 004D6B70 00000000 8000707B  *FSA .....@.Y......oH.(,........#*
  97A508A  09F3E388 001490F8 00000000 00007CB0   80000000 F0F0F1F7 D7D9E3F3 40404040  *.3Th...8......@.....0017PRT3    *
  97A50AA  00000000 7F4C1A20 00000000 00000000   00000002 004A4000 7F4AD400 7F4C1A54  *..................... ...M.....*
  97A50CA  27D1C5E2 F24BD3D6 C3C1D34B D7D9E3F3   40404040 40404040 40404040 40404040  *.JES2.LOCAL.PRT3                *
  97A50EA  40404040 40404040 00000000 01A00104   00F0C100 09F3E388 00000000 00000000  *        .........0A..3Th........*
  FSSCB
  97A5076  D9C1E2D7 C2C5D9E8 D7E6E3D9 D8404040   C6E2E2D4 C2E6E340 00007A30 00000000  *RASPBERYPWTRQ   FSSMBWT ........*
  97A5096  00280004 00000001 0000BE00 00000000   00000000 00000001 7F4E3398 0000BE00  *.........................+.q....*
  97A50B6  0000BE01 00000000 09F3E388 00000000   00000000 00000000 00000000 00000000  *.........3Th....................*
  97A50D6  7F4C16F8 804CB000 804CD330 097CB090   7F4F9DA4 004D6D90 00010001 00650000  *...8......L..@.....u.(.........*
  97A50F6  00000000 400085C0 00000000 00000000   00000000 00000000 000000A0 00000001  *.... .e........................*
  97A5116  D5D6D5C5 40404040 D5D6D5C5 40404040   E2E3C3F0 F0F0F1F2 00000000 FFFF0000  *NONE    NONE    STC00012........*
  JIB
  97A5082  D1C9C240 7F4C1D48 00000000 00000000   52020000 00000000 00000000 00000007  *JIB ............................*
  97A50A2  00002D06 0000000D E2E3C3F0 F0F0F1F3   C3CEABD5 D1D5C5E6 01000000 00000002  *........STC00013C..NJNEW........*
  97A50C2  00000000 00000007 00000065 00001A01   00001C02 00000005 00000000 5001A251  *............................&.s.*
  97A50E2  5D55D555 55555555 55555555 82838715   15151515 9C9781B1 B7908C15 82838715  *).N.........bcg......pa.....bcg.*
  97A5102  15151515 55555555 55555555 55555555   55555555 55555555 55555555 8490B79C  *............................d...*
  97A5122  91151515 B7BDB7A4 15151515 D7D6D24B   D1C5E2C9 C44B5BD1 C5E2D5C5 E6E24BE2  *j......u....POK.JESID.$JESNEWS.S*
  97A5142  E3C3F0F0 F0F0F74B C4F0F0F0 F0F1F0F1   4BD1C5E2 D5C5E6E2 40404040 40404040  *TC00007.D0000101.JESNEWS        *
  97A5162  40900050 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ..&............................*
  97A5182  00000000 004EBB78 004EBA98 004A4000   7F4AD000 00000000 7F4AD400 00000400  *.....+...+.q.. ...........M.....*
  97A51A2  00002D04 7F4AB000 004A6DAC 00000001   00000001 00000000 00000000          *............................    *
  JIB ( JSPA )
  97A508E  D1E2D7C1 00784000 C4C5C1D3 D3D6C340   E2E3C3F0 F0F0F1F3 D7D9E3F3 40404040  *JSPA.. .DEALLOC STC00013PRT3    *
  97A50AE  F0F0F1F7 004A43C8 F1404040 40404040   00010001 D3D6C3C1 D3404040 40404040  *0017...H1       ....LOCAL       *
  97A50CE  40404040 40404040 40404040 40404040   40404040 40404040 40404040 40404040  *                                *
  97A50EE  40404040 D1C5E2D5 C5E6E240 C1000000   00000000 00000000 00010054 00000000  *    JESNEWS A...................*
  97A510E  00000000 00000000 00000000 00540003   00000001 C9C2D4E4 E2C5D940 00000000  *....................IBMUSER ....*
  97A512E  00000000 D7D6D24B D1C5E2C9 C44B5BD1   C5E2D5C5 E6E24BE2 E3C3F0F0 F0F0F74B  *....POK.JESID.$JESNEWS.STC00007.*
  97A514E  C4F0F0F0 F0F1F0F1 4BD1C5E2 D5C5E6E2   40404040 40404040 40000000 00000000  *D0000101.JESNEWS         .......*
  JIB ( WORK JOE )
  97A509A  80000000 C1000000 00000000 A100000C   20000000 48000000 00000000 02000000  *....A.........................*
  97A50BA  00040001 09000065 00000000 0000001B   00000000 00000000 00000000 00000000  *................................*
  97A50DA  20000300 00010000 F1404040 40404040   00010001 C3CEABD6 C9C2D4E4 E2C5D940  *........1       ....C..OIBMUSER *
  97A50FA  00000000 02000000                                                         *........                        *
  JIB ( CHAR JOE )
  97A50A6  40000000 00000000 00000000 00000000   E2E3C440 40404040 5C5C5C5C 5C5C5C5C  * ..............STD     *********
  97A50C6  40404040 40404040 40404040 40404040   5C5C5C5C D3C9D5C5 40404040 00000000  *                ****LINE    ....*
  97A50E6  00000000 00000000 00000000                                                *............                    *
  JIB ( JOX )
  97A50B2  00000000 00000000 00002E08 00000000   2D040000 00000000 00000000 00000000  *................................*
  JIB ( SYSTEM BERT )
  97A50BE  00000000 00000000 00000000 00000000                                       *................                *
  JIB ( USER BERT )
  97A50CA
```

```
  GCB
  97A50D6  C7C3C240 00000002 00000000 004A6E48   7F4B0100 7F4C1A20 00002D04 00000002  *GCB ............................*
  97A50F6  00000001 C3CEABD5 0000000D ABD50001   0400FFFF C8000000 00000000 00000000  *....C..N.....N......H...........*
  97A5116  00000000 00000000 00000000 7F4AED90   00000000 40404040 40404040 D1C5E2F2  *....................        JES2*
  97A5136  40404040 D1C5E2D4 E2C7D3C7 00100208   08000000 00000000 00000000 00000000  *    JESMSGLG....................*
  97A5156  00000000 00000000 00000000 004B41E4   004B59A4 004B4DC4 004B5CD0          *..............U...u..(D..*.    *
  PDDB
  97A50E2  20C40085 00002E01 00002E07 00000002   0180C101 00010000 40404040 40404040  *.D.e.............A.....         *
  97A5102  00000800 0000000D 00000000 000003B8   E2E3C440 40404040 E2E3C4F3 5C5C5C5C  *................STD     STD3*****
  97A5122  40404040 40404040 00000000 00000000   C7E3F1F0 5C5C5C5C 5C5C5C5C 5C5C5C5C  *        ........GT10*************
  97A5142  5C5C5C5C 5C5C5C5C 00000000 00000000   0000FFFF FFFF1800 0001F140 40404040  *********...................1     *
  97A5162  40400000 C3CEABD5 00000000 00000000   00000000 00000000 00000000 40404040  *  ..C..N....................    *
  97A5182  40404040 D1C5E2F2 40404040 D1C5E2D4   E2C7D3C7 D3C9D5C5 40404040 5001A251  *    JES2    JESMSGLGLINE    &.s.*
  97A51A2  55559555 55555555 55555555 82838715   15151515 9C9781B1 B7908C15 82838715  *..n.........bcg......pa.....bcg.*
  97A51C2  15151515 55555555 55555555 B7B6969C   808C918C 55555555 55555555 9C9781B1  *..............o...j..........pa.*
  97A51E2  B7908C15 B7BDB7A4 15151515 C9C2D4E4   E2C5D940 00000000 00000000 00008090  *.......u....IBMUSER ............*
  97A5202  10020808 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
  97A5222  00020001 00000000 00000000 00000000   00000000 C9C2D4E4 E2C5D94B C4C5C1D3  *....................IBMUSER.DEAL*
  97A5242  D3D6C34B E2E3C3F0 F0F0F1F3 4BC4F0F0   F0F0F0F0 F24BD1C5 E2D4E2C7 D3C74040  *LOC.STC00013.D0000002.JESMSGLG  *
  IAZFSIP
  97A50EE  0000003C 00000007 00040001 00000000   00000000 00000000 7F4F8000 00000000  *................................*
  97A510E  0000000D 004A6DAC 7F4C1A20 004A6DAC   7F4C1A20 00000000 00000000          *............................    *
  IAZCHK (JES)
  97A50FA  C3C8D240 0098FFFF 00000400 00000000   00000000 00000000 00001A01 00002D04  *CHK .q..........................*
  97A511A  FF00FF01 00000005 00000000 F1404040   40404040 00010001 00000000 00000000  *............1       ............*
  97A513A  00000000 00000000 0000001A 01000005   FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  *................................*
  97A515A  00000005 00000000 FFFFFFFF FFFFFFFF   FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF  *................................*
  97A517A  5C5C5C40 C3D2D7E3 40C4C1E3 C1405C5C   5C404040 40404040                   **** CKPT DATA ***              *
```

Figure 20. Sample Output from Trace Identifier 25

Trace ID=26 sample

JES2 creates a trace identifier 26 record for each automatic restart manager request it processes in the main task. Figure 21 is a sample of the output from two traces for trace identifier 26.

```
18.26.48.97  ID =  26 ARM      JOB00020   RESTART   03937300  FUNCTION=REGISTER    RC=00000000 RSN=00000000 ARMFLAG1=80
 AT START:   JQEFLAG1=04  JQETYPE=41  JQEBUSY=02  JQEDEVID=00  JQEARMID=02
 AT END:     JQEFLAG1=04  JQETYPE=41  JQEBUSY=02  JQEDEVID=00  JQEARMID=02
 SSPJ:
       0  00300101 E2E2D7D1 00000000 E6E2C340   40404040 00000014 A9210DAD 000004B4  *....SSPJ....WSC     ....z.......*
      20  00000002 D1D6C2F0 F0F0F2F0 00000000                                       *....JOB00020....                *
18.29.18.90  ID =  26 ARM      JOB00020   RESTART   03937300  FUNCTION=RESTART     RC=00000000 RSN=00000000 ARMFLAG1=80
 AT START:   JQEFLAG1=06  JQETYPE=41  JQEBUSY=00  JQEDEVID=00  JQEARMID=02
 AT END:     JQEFLAG1=04  JQETYPE=41  JQEBUSY=00  JQEDEVID=00  JQEARMID=02
 SSPJ:
       0  00300103 E2E2D7D1 00000000 E6E2C340   40404040 00000014 A9210DAD 00000000  *....SSPJ....WSC     ....z.......*
      20  00000000 00000000 00000000 00000000                                       *................                *
```

Figure 21. Sample Output from Trace Identifier 26

Trace record contents

Table 36 is the sample record broken into fields which are described after the table.

Table 36. Fields for Trace 26

RECORD 1								
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
18.26.48.97	ID = 26	ARM	JOB00020	RESTART	03937300	REGISTER	00000000 00000000	ARMFLAG1=80
RECORD 2								
JQEFLAG1=04 JQETYPE=41 JQEBUSY=02 JQEDEVID=00 JQEARMID=02								
RECORD 3								
JQEFLAG1=04 JQETYPE=41 JQEBUSY=02 JQEDEVID=00 JQEARMID=02								
RECORD 4								
00300101 2E2D7D1 00000000 E6E2C340 40404040 00000014 A9210DAD 000004B4								
00000002 D1D6C2F0 F0F0F2F0								

- Record 1:

Field	Contents
1	Time-of-day clock value when the $TRACE was executed.
2	Trace identifier
3	Function being traced.
4	Job ID to which the request applies, if available.
5	PCE name
6	PCE address
7	The automatic restart manager function requested.
8	Return code and reason code
9	Contents of ARMFLAG1 in the $ARMWORK PCE

- Record 2:

  Contents of the JQE fields at the start of request processing:

  JQEFLAG1

  JQETYPE

  JQEBUSY

  JQEDEVID

JQEARMID

- Record 3:

  Contents of the JQE fields at the end of request processing:

  JQEFLAG1
  JQETYPE
  JQEBUSY
  JQEDEVID
  JQEARMID

- Record 4:

  Contents of the SSPJ control block.

Trace ID=27 sample

Figure 22 on page 64 is a sample of output from trace identifier 27. Trace ID 27 gathers information about Process SYSOUT (PSO) external writer and conversational terminal systems processing. The trace gathers information from the $PSO control block at the start of request processing and at the end of request processing.

```
                ---  JES2 SP 5.2.0 EVENT TRACE LOG  ---  NODE POK      MEMBER IBM1 DATE 94.307  ---
11.48.19.99  ID =  27 PSO      ASID 001D                     BEFORE   ------------------------------------------------
  0  0391DD20 5BD7E2D6 00000000 AA1FFDF2   00000000 00000000 00000000 00000000  *.j..$PSO.......2.............*
 20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
100  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
120  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
160  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
180  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
1A0  60100000 00000000 00000000 00000000   00000000 00000000 D3D6C3C1 D3404040  *-......................LOCAL   *
1C0  40404040 40404040 00000000 00000000   00000000 00000000 00000000 00000000  *        .....................*
1E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
200  C1404040 40404040 004F8D90 00000000   00000000 00000000 00000000 00000000  *A       .....................*
220  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
240  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
260  00000000 00000000 00000000 00000000   00000000 004FD058 00000000 00000000  *.............................*
280  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
2A0  00000000 00000000 00000000 00000000   00000000 00010000 00000000 5001A055  *.........................&;...*
2C0  95555555 55555555 55555555 55555555   55555555 55555555 55555555 55555555  *n............................*
2E0  55555555 55555555 55555555 55555555   55555555 55555555 55555555 09151515  *.............................*
300  15151515 55555555 55555555 00000000   00000000 00000000 00000000 00000000  *.............................*
320  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
340  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
360  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
380  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
3A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
3C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
3E0  00000000 00000000 00000000 00000000                                       *.................            *
11.48.19.99  ID =  27 PSO      ASID 001D                     AFTER    ------------------------------------------------
  0  0391DD20 5BD7E2D6 00000000 AA1FFDF2   00000000 00000000 00000000 00000000  *.j..$PSO.......2.............*
 20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
 E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
100  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
120  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
160  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
180  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
1A0  60100000 00000000 00000000 00000000   00000000 00000000 D3D6C3C1 D3404040  *-......................LOCAL   *
1C0  40404040 40404040 00000000 00000000   00000000 00000000 00000000 00000000  *        .....................*
1E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
200  C1404040 40404040 004F8D90 00000000   00000000 00000000 00000000 00000000  *A       .....................*
220  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
240  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
260  00000000 00000000 00000000 00000000   00000000 004FD058 00000004 00000000  *.............................*
280  00000000 03901550 00000000 004F8D90   00FA2200 00FD2542 00000000 00000000  *.......&;....................*
2A0  00000000 00000000 00000000 00000000   00000000 00010000 00000000 5001A055  *.........................&;...*
2C0  95555555 55555555 55555555 55555555   55555555 55555555 55555555 55555555  *n............................*
2E0  55555555 55555555 55555555 55555555   55555555 55555555 55555555 09151515  *.............................*
300  15151515 55555555 55555555 00000000   00000000 00000000 00000000 00000000  *.............................*
320  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
340  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
360  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
380  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
3A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
3C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.............................*
3E0  00000000 00000000 00000000 00000000                                       *.................            *
```

Figure 22. Sample Output from Trace Identifier 27

Trace ID=28 sample

Figure 23 on page 65, Figure 24 on page 66, Figure 25 on page 67, Figure 26 on page 68 and Figure 27 on page 69 show sample output for trace identifier 28. Trace ID 28 gathers information about the SSOB function-dependent area of the IAZSSS2 macro for SYSOUT application program interface before the macro call from the application's address space to JES2.

For a description of the SSOB extension for the SYSOUT application program interface macro see *z/OS JES2 Data Areas Volume 5*.

Attention: The formatted portion of the records for SSS2*xxxx* fields for TRACE ID=28 and TRACE ID=29 might contain binary data instead of EBCDIC, and vice versa. Therefore, always reference the unformatted data, which is always in binary format.

```
14.06.56.97  ID =  28 SAPIXM   ASID 0013                    BEFORE   ------------------------------------------------
        0  5BE2C1D7 80800000 00000000 00000000   04100100 E2E2E2F2 00000000 00000000  *$SAP.............SSS2........*
       20  00000000 00000000 00000000 00000000   00000000 01000000 04800078 00000000  *...............................*
       40  00000000 00000000 00000006 00000000   00000000 D1F2C3D7 E9F2F4E7 00000000  *...................J2CPZ24X....*

⋮
                                                      00 00000000  *...............................*
      400  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
      420  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
      440  00000000 00000000 00000000 00FBE380   0000004C 00000003 005CCB80 005CCB80  *.............T..........*...*..*
      460  7F666000 7F637000 7F636000 7F631960   048050AE 04805020 00000000 00000000  *..-.......-....-..&;..&;........*
      480  00000000 00000000 00000000 00000000   00000000 043FFBA8 00000000 00000000  *.....................y........*
      4A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
      4C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
      4E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*
      500  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *...............................*

⋮
                                                    00 00000000  *...............................*
```

Figure 23. Sample Output from Trace Identifier 28 Before Call to JES2

```
11.46.32.48028 ID= 28 SAPIXM   ASID 0027           004B3538  AFTER    ----------------------------------------------
FORMATTED DATA, SORTED ALPHABETICALLY FOLLOWS
+0D73 SAPREAS =  F0                    +0458 SAPRETN =  00000000
+0D52 SAP#PDDB=  0000                  +0D50 SAP#SKIP=  0000                  +048C SAPACCT =  7F58ADE8
+04AC SAPANCHR=  00000000              +0D61 SAPAPPL =  E2C1D7C9 D6E4E340     +0464 SAPASCB =  00FC3C00
+0468 SAPASCBT=  0000009C 00000002     +0488 SAPBTOK =  7F58AE77              +0480 SAPCHK  =  7F463000
+0558 SAPCHJOE=
        0  40000000 00800024 00000000 00000006   E2E3C440 40404040 5C5C5C5C 5C5C5C5C  * ..............STD     *********
       20  40404040 40404040 40404040 40404040   5C5C5C5C D3C9D5C5 40404040 00000000  *               ****LINE    ....*
       40  00000000 00000000 00000002                                               *............                     *
+0008 SAPCKEY =  8C                    +0D54 SAPCLFT =  0000
+0A6C SAPCTKN =
        0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       40  00000000 00000000 00000000 00000000                                       *................                *
+0490 SAPDTKN =  7F58AE9B              +0000 SAPEYE  =  $SAP                  +0005 SAPFLAGJ=  00
+0004 SAPFLAG1=  60                    +0D70 SAPFLAG2=  48                    +0D72 SAPFLAG3=  00
+0006 SAPFLGJ2=  20                    +0478 SAPIOT  =  7F467000              +04B4 SAPIOTC =  00000000
+04B8 SAPIOTF =  00000000              +04BC SAPIOTP =  00000000              +04B0 SAPIOTW =  00000000
+047C SAPJCT  =  7F465000              +0448 SAPJNOHI=  00000000              +0444 SAPJNOLO=  00000000
+05A4 SAPJQEAR=
        0  10010000 00000000 01000000 00C10000   C31D0186 00001C02 00010001 00000000  *.............A..C..f............*
       20  00000000 C3D3C1E2 E2E3D140 C9C2D4E4   E2C5D940 00000000 00000000 00000002  *....CLASSTJ IBMUSER ............*
       40  11000003 00000000 0000000A 00010000   40404040 40404040 00010000 FFFFFFFF  *................        ........*
+0D71 SAPMCLAS=  00                    +0007 SAPMSTRV=  08                    +04D0 SAPMTRB =  09841BC8
+000C SAPNEXT =  00000000              +0498 SAPNDH  =  00000000              +0494 SAPNJH  =  00000000
+0D56 SAPONODE=  0000                  +0474 SAPOTCB =  004B3538
+076C SAPPDDB =
        0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      100  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      120  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      160  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+04C4 SAPPDDBF=  0000                  +04C2 SAPPDDBO=  0000                  +04C6 SAPPDDBP=  0000
+04C0 SAPPDDBW=  0000
+08EC SAPPDDB2=
        0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      100  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      120  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      160  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
+045C SAPPRIV =  00000000 00000000     +0D58 SAPRBA  =  00000000 00000000     +0438 SAPROUTE=  00000000
+044C SAPROUTN=  00000000              +04CC SAPSDB  =  00000000              +04C8 SAPSJB  =  099B6A50
+0D78 SAPSTCK =  C31D01AC 2D0D5A20     +049C SAPSWB  =  00000000              +04A4 SAPSWBTK=  00000000 00000000
+0470 SAPTCB  =  004B3538              +04A0 SAPTJEV =  00000000              +0D60 SAPTYPE =  01
+043C SAPUSER =  00000000 00000000     +0450 SAPUSERN=                        +0484 SAPWAVE =  7F464CB0
                                                    *........                      *
```

Figure 24. Sample Output from Trace Identifier 28 After Call to JES2 (1 of 4)

```
+066C SAPWCJOE=
       0  40000000 00800024 00000000 00000006   E2E3C440 40404040 5C5C5C5C 5C5C5C5C  * ..............STD     *********
      20  40404040 40404040 40404040 40404040   5C5C5C5C D3C9D5C5 40404040 00000000  *                ****LINE    ....*
      40  00000000 00000000 00000002                                               *............                        *
+04D8 SAPWJTOF=  00000000               +04D4 SAPWKOFF=  00000270              +0D6C SAPWRNUM=  80000001
+0C84 SAPWS   =
       0  4DD1D6C2 D5C1D4C5 615D4000 00000000   00000000 00000000 00000000 00000000  *(JOBNAME/) .....................*
      20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      C0  00000000 00000000                                                         *........                          *
+0ABC SAPWSP  =
       0  E6E2D740 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *WSP ............................*
      20  1000FFFF FFFF0000 09D55150 09D559E0   00000000 00000000 00000000 00000000  *.........N.&.N.................*
      40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      80  09D4D0DC 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *.M..............................*
      A0  00000000 00000000 00000000 00000000   00000000 C3D3C1E2 E2E3D140 00000000  *....................CLASSTJ ....*
      C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     100  00000000 00000000 80000001 0000009C   00000002 C9C2D4E4 E2C5D94B E2C1D7C9  *...................IBMUSER.SAPI*
     120  D6E4E340 40000D00 00000000 00000000   00000000 00000000 00000000 00000000  *OUT  ..........................*
     140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     160  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     180  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     1A0  00000000 00000000 00000000 040F0000   00000000 00001000 00000000 00000000  *................................*
     1C0  00000000 FFFFFFFF                                                         *........                          *
+0604 SAPWWJOE=
       0  80000000 C1800034 00000000 20000009   20000007 48000005 00000007 02000000  *....A...........................*
      20  00000000 09000000 00001C0C 00000024   00000000 00000000 00000000 00001B06  *................................*
      40  0D000100 00010000 F1404040 40404040   00010001 C31D0189 C9C2D4E4 E2C5D940  *........1       ....C..iIBMUSER *
      60  00000000 02000000                                                         *........                          *
+0720 SAP2CJOE=
       0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      40  00000000 00000000 00000000                                               *............                        *
+06B8 SAP2WJOE=
       0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      60  00000000 00000000                                                         *........                          *
+04F0 SAPWKJOE=
       0  80000000 C1800034 00000000 20000009   20000007 48000005 00000007 02000000  *....A...........................*
      20  00000000 09000000 00001C0C 00000024   00000000 00000000 00000000 00001B06  *................................*
      40  0D000100 00010000 F1404040 40404040   00010001 C31D0189 C9C2D4E4 E2C5D940  *........1       ....C..iIBMUSER *
      60  00000000 02000000                                                         *........                          *
+0003 SSS2REAS=  00                     +02E6 SSS2RET1=  00                     +02E7 SSS2RET2=  00
+02E8 SSS2RET3=  00                     +02E9 SSS2RET4=  00                     +02EA SSS2RET5=  00
+03CC SSS2ACCT=  00000000               +0170 SSS2AGE =                         +0010 SSS2APL1=
+0008 SSS2APPL=  SAPIOUT                +0228 SSS2BTOK=  00000000               +02F4 SSS2BYCT=  00000000
+0368 SSS2CDS =  00000000               +0154 SSS2CHAR=                         +032C SSS2CHR1=
+0330 SSS2CHR2=                         +0334 SSS2CHR3=                         +0338 SSS2CHR4=
+02D4 SSS2CLAR=                         +0134 SSS2CLAS=                         +01EA SSS2CLFT=  0000
+0110 SSS2CLSL=                                                                 +022C SSS2COPY=  0000
+0230 SSS2CPYG=  00                     +005A SSS2CREA=                         +0250 SSS2CRER=
+018C SSS2CTKN=  00000000               +03E4 SSS2DATE=                         +01C0 SSS2DCLS=  00
+01D8 SSS2DDES=  00000000 00000000 00000000 00000000 0000                       +0268 SSS2DESR=
+0082 SSS2DEST=                         +00A6 SSS2DES2=                         +0310 SSS2DDND=
+01C8 SSS2DFOR=  00000000 00000000      +01D0 SSS2DPGM=  00000000 00000000      +01EC SSS2DPRI=  00
+02DE SSS2DSID=                         +029A SSS2DSN =
```

Figure 25. Sample Output from Trace Identifier 28 After Call to JES2 (2 of 4)

```
+01BC SSS2DSP1=  00                    +01BD SSS2DSP2=  00                    +0378 SSS2DSTR=  00000000
+0028 SSS2ECBP=  00000000              +0004 SSS2EYE =  SSS2                  +014C SSS2FCB =
+0370 SSS2FCBR=  00000000              +0361 SSS2FLSC=  00                    +0168 SSS2FLSH=
+00C0 SSS2FORM=
+0282 SSS2FORR=                        +0052 SSS2JBIH=                        +004A SSS2JBIL=
+0240 SSS2JBIR=                        +0258 SSS2JDVT=                        +021C SSS2JEST=
+0042 SSS2JOBN=  C3D3C1E2 E2E3D140     +0238 SSS2JOBR=                        +0000 SSS2LEN =  0428
+0363 SSS2LINC=  00                    +0140 SSS2LMAX=  00000000              +013C SSS2LMIN=  00000000
+02EC SSS2LNCT=  00000000              +040B SSS2LSAB=                        +03F0 SSS2MBR =
+02DC SSS2MLRL=  0000                  +0164 SSS2MOD =                        +035C SSS2MODT=
+003E SSS2MSC1=  00                    +0408 SSS2MXRC=                        +03F8 SSS2NACT=
+036C SSS2NJED=  00000000              +03F4 SSS2NJEJ=  00000000              +03BC SSS2NOTN=
+03C4 SSS2NOTU=                        +0190 SSS2ODST=  00000000 00000000     +033C SSS2OGNM=
+0248 SSS2OJBI=                        +03D8 SSS2ORG =                        +02F0 SSS2PGCT=  00000000
+00B8 SSS2PGMN=                        +027A SSS2PGMR=                        +0148 SSS2PMAX=  00000000
+0144 SSS2PMIN=  00000000              +03A0 SSS2PNAM=                        +0300 SSS2PRCD=
+0362 SSS2PRIO=  00                    +0324 SSS2XEQ =  00000000 00000000     +0062 SSS2PRMO=
+0260 SSS2PRMR=                        +002C SSS2RBA =  00000000 00000000     +02FC SSS2RCCT=
+035D SSS2RFLS=                        +02EB SSS2RFOR=                        +0358 SSS2RMOD=
+03B4 SSS2ROOM=                        +016C SSS2SECT=                        +02C8 SSS2SEGM=  00000000
+0037 SSS2SEL1=  E4                    +0038 SSS2SEL2=  00                    +0039 SSS2SEL3=  00
+003A SSS2SEL4=  00                    +003B SSS2SEL5=  00                    +003C SSS2SEL6=  00
+0308 SSS2STPD=                        +0318 SSS2SWBT=  00000000 00000000     +0320 SSS2SWTU=  00000000
+03E8 SSS2SYS =                        +03E0 SSS2TIME=  00000000
+0292 SSS2TJID=                        +028A SSS2TJN =                        +0364 SSS2TOD =  00000000
+0024 SSS2TYPE=  01                    +0150 SSS2UCS =                        +0374 SSS2UCSR=  00000000
+0034 SSS2UFLG=  00                    +0400 SSS2USID=                        +0002 SSS2VER =  02
+0174 SSS2VOL =                                                               +02D0 SSS2WRSN=  00000000
+02CC SSS2WRTN=  00000000              +03D0 SSS2XEQ =
TRCID= 28 UNFORMATTED DATA FOLLOWS
       0  5BE2C1D7 60002008 8C000000 00000000   04280200 E2E2E2F2 E2C1D7C9 D6E4E340  *$SAP-.............SSS2SAPIOUT *
      20  00000000 00000000 00000000 00000000   00000000 01000000 00000000 00000000  *................................*
      40  00000000 000000E4 00000000 00000000   0000C3D3 C1E2E2E3 D1400000 00000000  *.......U.........CLASSTJ ......*
      60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     100  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     120  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     160  00000000 00000000 00000000 00000000   00000000 00000000 00000000 09DF8078  *................................*
     180  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     1A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     1C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     1E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     200  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     220  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     240  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     260  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     280  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     2A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     2C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     2E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     300  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     320  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     340  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     360  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     380  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     3A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     3C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     3E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     400  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     420  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     440  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     460  00000000 00FC3C00 0000009C 00000002   004B3538 004B3538 7F467000 7F465000  *..........................&.*
     480  7F463000 7F464CB0 7F58AE77 7F58ADE8   7F58AE9B 00000000 00000000 00000000  *..............Y................*
```

Figure 26. Sample Output from Trace Identifier 28 After Call to JES2 (3 of 4)

```
4A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
4C0  00000000 00000000 099B6A50 00000000   09841BC8 00000270 00000000 00000000  *..........&.....d.H............*
4E0  00000000 00000000 00000000 00000000   80000000 C1800034 00000000 20000009  *...................A..........*
500  20000007 48000005 00000007 02000000   00000000 09000000 00001C0C 00000024  *................................*
520  00000000 00000000 00000000 00001B06   0D000100 00010000 F1404040 40404040  *........................1       *
540  00010001 C31D0189 C9C2D4E4 E2C5D940   00000000 02000000 40000000 00800024  *....C..iIBMUSER ........ .......*
560  00000000 00000006 E2E3C440 40404040   5C5C5C5C 5C5C5C5C 40404040 40404040  *........STD     ********        *
580  40404040 40404040 5C5C5C5C D3C9D5C5   40404040 00000000 00000000 00000000  *        ****LINE    ............*
5A0  00000002 10010000 00000000 01000000   00C10000 C31D0186 00001C02 00010001  *................A..C..f........*
5C0  00000000 00000000 C3D3C1E2 E2E3D140   C9C2D4E4 E2C5D940 00000000 00000000  *........CLASSTJ IBMUSER ........*
5E0  00000002 11000003 00000000 0000000A   00010000 40404040 40404040 00010000  *....................         ....*
600  FFFFFFFF 80000000 C1800034 00000000   20000009 20000007 48000005 00000007  *........A.....................*
620  02000000 00000000 09000000 00001C0C   00000024 00000000 00000000 00000000  *................................*
640  00001B06 0D000100 00010000 F1404040   40404040 00010001 C31D0189 C9C2D4E4  *............1       ....C..iIBMU*
660  E2C5D940 00000000 02000000 40000000   00800024 00000000 00000006 E2E3C440  *SER ........ ..............STD *
680  40404040 5C5C5C5C 5C5C5C5C 40404040   40404040 40404040 40404040 5C5C5C5C  *    ********                 *****
6A0  D3C9D5C5 40404040 00000000 00000000   00000000 00000002 00000000 00000000  *LINE    .......................*
6C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
6E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
700  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
720  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
740  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
760  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
780  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
7A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
7C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
7E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
800  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
820  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
840  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
860  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
880  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
8A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
8C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
8E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
900  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
920  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
940  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
960  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
980  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
9A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
9C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
9E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
A00  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
A20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
A40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
A60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
A80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
AA0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 E6E2D740  *.........................WSP *
AC0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 1000FFFF  *................................*
AE0  FFFF0000 09D55150 09D559E0 00000000   00000000 00000000 00000000 00000000  *.....N.&.N.....................*
B00  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
B20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 09D4D0DC  *.........................M..*
B40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
B60  00000000 00000000 00000000 00000000   C3D3C1E2 E2E3D140 00000000 00000000  *................CLASSTJ ........*
B80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
BA0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
BC0  00000000 80000001 0000009C 00000002   C9C2D4E4 E2C5D94B E2C1D7C9 D6E4E340  *...............IBMUSER.SAPIOUT *
BE0  40000D00 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ..............................*
C00  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
C20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
C40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
C60  00000000 00000000 040F0000 00000000   00001000 00000000 00000000 00000000  *................................*
C80  FFFFFFFF 4DD1D6C2 D5C1D4C5 615D4000   00000000 00000000 00000000 00000000  *....(JOBNAME/) ................*
CA0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
CC0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
CE0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
D00  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
D20  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
D40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
D60  01E2C1D7 C9D6E4E3 40000000 80000001   480000F0 00000000 C31D01AC 2D0D5A20  *.SAPIOUT ..........0....C.......*
```

Figure 27. Sample Output from Trace Identifier 28 After Call to JES2 (4 of 4)

Trace ID=29 sample

Figure 28 on page 70, Figure 29 on page 71 and Figure 30 on page 72 show sample output from trace identifier 29. Trace 29 gathers information about the SSOB function-dependent area of the IAZSSS2 macro for SYSOUT application program interface before the macro call from the application's address space to the JES2 main task.

For a description of the SSOB extension for the SYSOUT application program interface macro see *z/OS JES2 Data Areas Volume 5*.

Attention: The formatted portion of the records for SSS2*xxxx* fields for TRACE ID=28 and TRACE ID=29 might contain binary data instead of EBCDIC, and vice versa. Therefore, always reference the unformatted data, which is always in binary format.

```
14.06.56.97  ID =  29 SAPISSI  ASID 0013                    BEFORE   ------------------------------------------------
        0  04100100 E2E2E2F2 00000000 00000000   00000000 00000000 00000000 00000000  *....SSS2........................*
       20  00000000 01000000 04800078 00000000   00000000 00000006 00000000 00000000  *................................*
       40  00000000 D1F2C3D7 E9F2F4E7 00000000   00000000 00000000 00000000 00000000  *....J2CPZ24X....................*
       60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
       E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      100  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      120  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      160  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      180  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      1A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      1C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      1E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      200  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      220  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      240  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      260  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      280  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      2A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      2C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      2E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      300  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      320  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      340  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      360  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      380  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      3A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      3C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      3E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      400  00000000 00000000 00000000 00000000                                       *................                *
```

Figure 28. Sample Output from Trace Identifier 29 Before Call to JES2

```
11.46.32.48541 ID= 29 SAPISSI  ASID 0027           004B3538  AFTER    ------------------------------------------------
FORMATTED DATA, SORTED ALPHABETICALLY FOLLOWS
+0003 SSS2REAS=  00                    +02E6 SSS2RET1=  E0                    +02E7 SSS2RET2=  00
+02E8 SSS2RET3=  20                    +02E9 SSS2RET4=  00                    +02EA SSS2RET5=  20
+03CC SSS2ACCT=  7F58ADE8              +0170 SSS2AGE =                        +0010 SSS2APL1=
+0008 SSS2APPL=  SAPIOUT               +0228 SSS2BTOK=  7F58AE77              +02F4 SSS2BYCT=  00000000
+0368 SSS2CDS =  00000001              +0154 SSS2CHAR=                        +032C SSS2CHR1=  ****
+0330 SSS2CHR2=  ****                  +0334 SSS2CHR3=  ****                  +0338 SSS2CHR4=  ****
+02D4 SSS2CLAR=  A                     +0134 SSS2CLAS=                        +01EA SSS2CLFT=  0000
+0110 SSS2CLSL=                                                               +022C SSS2COPY=  0001
+0230 SSS2CPYG=  00                    +005A SSS2CREA=                        +0250 SSS2CRER=  IBMUSER
+018C SSS2CTKN=  00000000              +03E4 SSS2DATE=                        +01C0 SSS2DCLS=  00
+01D8 SSS2DDES=  00000000 00000000 00000000 00000000 0000                     +0268 SSS2DESR=  LOCAL
+0082 SSS2DEST=                        +00A6 SSS2DES2=                        +0310 SSS2DDND=  JESMSGLG
+01C8 SSS2DFOR=  00000000 00000000     +01D0 SSS2DPGM=  00000000 00000000     +01EC SSS2DPRI=  00
+02DE SSS2DSID=                        +029A SSS2DSN =  IBMUSER.CLASSTJ.JOB00010.D0000002.JESMSGLG
+01BC SSS2DSP1=  00                    +01BD SSS2DSP2=  00                    +0378 SSS2DSTR=  7F58AE9B
+0028 SSS2ECBP=  00000000              +0004 SSS2EYE =  SSS2                  +014C SSS2FCB =
+0370 SSS2FCBR=  5C5C5C5C              +0361 SSS2FLSC=  FF                    +0168 SSS2FLSH=
+00C0 SSS2FORM=
+0282 SSS2FORR=  STD                   +0052 SSS2JBIH=                        +004A SSS2JBIL=
+0240 SSS2JBIR=  JOB00010              +0258 SSS2JDVT=                        +021C SSS2JEST=
+0042 SSS2JOBN=  C3D3C1E2 E2E3D140     +0238 SSS2JOBR=  CLASSTJ               +0000 SSS2LEN =  0428
+0363 SSS2LINC=  00                    +0140 SSS2LMAX=  00000000              +013C SSS2LMIN=  00000000
+02EC SSS2LNCT=  0000000E              +040B SSS2LSAB=                        +03F0 SSS2MBR =  IBM1
+02DC SSS2MLRL=  0085                  +0164 SSS2MOD =                        +035C SSS2MODT=
+003E SSS2MSC1=  00                    +0408 SSS2MXRC=                        +03F8 SSS2NACT=
+036C SSS2NJED=  00000000              +03F4 SSS2NJEJ=  00000000              +03BC SSS2NOTN=  POK
+03C4 SSS2NOTU=  IBMUSER               +0190 SSS2ODST=  00000000 00000000     +033C SSS2OGNM=  1.1.1
+0248 SSS2OJBI=  JOB00010              +03D8 SSS2ORG =  POK                   +02F0 SSS2PGCT=  00000000
+00B8 SSS2PGMN=                        +027A SSS2PGMR=                        +0148 SSS2PMAX=  00000000
+0144 SSS2PMIN=  00000000              +03A0 SSS2PNAM=                        +0300 SSS2PRCD=
+0362 SSS2PRIO=  90                    +0324 SSS2XEQ =  00000000 00000000     +0062 SSS2PRMO=
+0260 SSS2PRMR=  LINE                  +002C SSS2RBA =  00000000 00000000     +02FC SSS2RCCT=
+035D SSS2RFLS=  ****                  +02EB SSS2RFOR=  D                     +0358 SSS2RMOD=  ****
+03B4 SSS2ROOM=                        +016C SSS2SECT=                        +02C8 SSS2SEGM=  00000000
+0037 SSS2SEL1=  E4                    +0038 SSS2SEL2=  00                    +0039 SSS2SEL3=  00
+003A SSS2SEL4=  00                    +003B SSS2SEL5=  00                    +003C SSS2SEL6=  00
+0308 SSS2STPD=  JES2                  +0318 SSS2SWBT=  00000000 00000000     +0320 SSS2SWTU=  00000000
+03E8 SSS2SYS =  SY1                   +03E0 SSS2TIME=  00409FD6
+0292 SSS2TJID=                        +028A SSS2TJN =                        +0364 SSS2TOD =  C31D0186
+0024 SSS2TYPE=  01                    +0150 SSS2UCS =                        +0374 SSS2UCSR=  5C5C5C5C
+0034 SSS2UFLG=  00                    +0400 SSS2USID=                        +0002 SSS2VER =  02
+0174 SSS2VOL =                                                               +02D0 SSS2WRSN=  00000000
+02CC SSS2WRTN=  00000000              +03D0 SSS2XEQ =  POK
```

Figure 29. Sample Output from Trace Identifier 29 After Call to JES2 (1 of 2)

```
TRCID= 29 UNFORMATTED DATA FOLLOWS
       0  00000000 04280200 E2E2E2F2 E2C1D7C9   D6E4E340 00000000 00000000 00000000  *........SSS2SAPIOUT ............*
      20  00000000 00000000 01000000 00000000   00000000 00000000 000000E4 00000000  *............................U....*
      40  00000000 0000C3D3 C1E2E2E3 D1400000   00000000 00000000 00000000 00000000  *......CLASSTJ ..................*
      60  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
      E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     100  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     120  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     160  00000000 00000000 00000000 00000000   09DF8078 00000000 00000000 00000000  *................................*
     180  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     1A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     1C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     1E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     200  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     220  00001000 80000001 00000000 7F58AE77   00010000 00000000 00000000 C3D3C1E2  *.............................CLAS*
     240  E2E3D140 D1D6C2F0 F0F0F1F0 D1D6C2F0   F0F0F1F0 C9C2D4E4 E2C5D940 00000000  *STJ JOB00010JOB00010IBMUSER ....*
     260  00000000 D3C9D5C5 40404040 D3D6C3C1   D3404040 40404040 40404040 40404040  *....LINE    LOCAL               *
     280  40404040 4040E2E3 C4404040 40404040   40404040 40404040 40404040 4040C9C2  *      STD                       IB*
     2A0  D4E4E2C5 D94BC3D3 C1E2E2E3 D14BD1D6   C2F0F0F0 F1F04BC4 F0F0F0F0 F0F0F24B  *MUSER.CLASSTJ.JOB00010.D0000002.*
     2C0  D1C5E2D4 E2C7D3C7 40400000 00000000   00000000 00000000 C1000000 00000000  *JESMSGLG  ..............A.......*
     2E0  00854040 40404040 4040E000 200020C4   0000000E 00000000 00000000 00000310  *.e        .....D................*
     300  00000000 40404040 40404040 D1C5E2F2   40404040 D1C5E2D4 E2C7D3C7 00000000  *....        JES2    JESMSGLG....*
     320  00000000 00000000 00000000 00000000   5C5C5C5C 5C5C5C5C 5C5C5C5C 5C5C5C5C  *................****************
     340  F14BF14B F1404040 40404040 40404040   40404040 40404040 40400000 5C5C5C5C  *1.1.1                     ..*****
     360  005C5C5C 5CFF9000 C31D0186 00000001   00000000 5C5C5C5C 5C5C5C5C 7F58AE9B  *.****...C..f........********....*
     380  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
     3A0  00000000 40404040 40404040 40404040   40404040 40404040 40404040 40404040  *....                            *
     3C0  D7D6D240 40404040 C9C2D4E4 E2C5D940   7F58ADE8 D7D6D240 40404040 D7D6D240  *POK     IBMUSER ...YPOK     POK *
     3E0  40404040 00409FD6 0108282F E2E8F140   40404040 C9C2D4F1 00000000 00000000  *    . .O....SY1     IBM1........*
     400  00000000 40404040 40404040 00000000   00000000 00000000 00000000 00000000  *....        ....................*
     420  00000000 00000000 00000000 00000000                                       *................                *
```

Figure 30. Sample Output from Trace Identifier 29 After Call to JES2 (2 of 2)

Trace ID=30 sample

Figure 31 on page 73 is a sample of output from trace identifier 30. Trace identifier 30 traces all $#POST macro calls made by JES2 processing. The trace provides counts such as the number of devices scanned, the number of work selection calls made, and the amount of CPU time spent in the $#POST service.

This record, in conjunction with trace identifier 20, provides a means to analyze selection criteria and queue search overhead to tune work selection criteria.

```
18.02.49.59   ID =  30 $#POST   STC00424   SPIN      05ECB268  $#POST TYPE=JOE  MASPOST=YES ------
                JOB = $TRCLOG  (STC00424)        OUTGRP =  4.00001.00001
                DEVICES SCANNED      =        0  DEVICES POSTED       =        0
                PSO WRITERS SCANNED  =        0  PSO WRITERS POSTED   =        0
                SAPI WRITERS SCANNED =        0  SAPI WRITERS POSTED  =        0
                WORK SELECTION CALLS =        0
                C PU TIME USED (SEC)  =        0.000003
                $#POST   CALLED BY   = HASPJOS  0006E000 + 000F50
18.02.51.09   ID =  30 $#POST   STC00424   SPIN      05ECB268  $#POST TYPE=JOE  MASPOST=YES --------
                JOB = $TRCLOG  (STC00424)        OUTGRP =  4.00001.00001
                DEVICES SCANNED      =       75  DEVICES POSTED       =        1
                PSO WRITERS SCANNED  =        0  PSO WRITERS POSTED   =        0
                S API WRITERS SCANNED =       0  SAPI WRITERS POSTED  =        0
                WORK SELECTION CALLS =       75
                CPU TIME USED (SEC)  =        0.003435
                $#POST   CALLED BY   = HASPSPIN 000F9000 + 0015CE
18.06.49.61   ID =  30 $#POST   JOB00431   HOPE      05ECBB88  $#POST TYPE=JOE  MASPOST=YES -------
                JOB = DEALLOC  (JOB00431)        OUTGRP =  1.00001.00001
                DEVICES SCANNED      =        0  DEVICES POSTED       =        0
                PSO WRITERS SCANNED  =        0  PSO WRITERS POSTED   =        0
                SAPI WRITERS SCANNED =        0  SAPI WRITERS POSTED  =        0
                WORK SELECTION CALLS =        0
                CPU TIME USED (SEC)  =        0.000022
                $#POST   CALLED BY   = HASPJOS  0006E000 + 000F50
18.06.49.63   ID =  30 $#POST   JOB00431   HOPE      05ECBB88  $#POST TYPE=JQE  MASPOST=YES -------
                JOB = DEALLOC  (JOB00431)
                JOES SCANNED         =        1
                DEVICES SCANNED      =       75  DEVICES POSTED       =        0
                PSO WRITERS SCANNED  =        0  PSO WRITERS POSTED   =        0
                SAPI WRITERS SCANNED =        0  SAPI WRITERS POSTED  =        0
                WORK SELECTION CALLS =       75
                CPU TIME USED (SEC)  =        0.002907
                $#POST   CALLED BY   = HASPHOPE 0006C000 + 000A5A
18.06.56.39   ID =  30 $#POST   JOB00431   COMM      06023400  $#POST TYPE=JOE  MASPOST=YES -------
                JOB = DEALLOC  (JOB00431)        OUTGRP =  1.00001.00001
                DEVICES SCANNED      =        4  DEVICES POSTED       =        4
                PSO WRITERS SCANNED  =        0  PSO WRITERS POSTED   =        0
                SAPI WRITERS SCANNED =        0  SAPI WRITERS POSTED  =        0
                WORK SELECTION CALLS =        0
                CPU TIME USED (SEC)  =        0.000010
                $#POST   CALLED BY   = HASPCOMM 0004B000 + 00D1
```

Figure 31. Sample Output from Trace Identifier 30

Trace record contents

Table 37 is the sample record broken into fields which are described after the table.

Table 37. Fields for Trace 30

RECORD 1						
1	**2**	**3**	**4**	**5**	**6**	**7**
18.02.49.59	ID = 30	$#POST	STC00424	SPIN	05ECB268	$#POST TYPE=JOE MASPOST=YES
RECORD 2						
JOB = $TRCLOG (STC00424) OUTGRP = 4.00001.00001						
RECORD 3						
JOES SCANNED = 1						
RECORD 4						
DEVICES SCANNED = 75 DEVICES POSTED = 1						
RECORD 5						

Table 37. Fields for Trace 30 (continued)

RECORD 1
PSO WRITERS SCANNED = 2 PSO WRITERS POSTED = 0
RECORD 6
SAPI WRITERS SCANNED = 2 SAPI WRITERS POSTED = 0
RECORD 7
WORK SELECTION CALLS = 75
RECORD 8
CPU TIME USED (SEC) = 0.000003
RECORD 9
$#POST CALLED BY = HASPJOS 0006E000 + 000F50

The trace output contains the following:

- Record 1:

Field	Contents
1	Time-of-day clock value when the $TRACE was processed.
2	Trace identifier.
3	Function.
4	Job ID, if available.
5	Name of processor which issued the $#POST.
6	PCE address of processor which issued the $TRACE.
7	Description of $#POST call.

- Record 2:

  The job name and job id (JOB=) and output group identifier (OUTGRP=) of the output which has just become available. OUTGRP= is only displayed for $#POST TYPE=JOE type records

- Record 3:

  JOES SCANNED - For $#POST TYPE=JQE, the number of JOEs associated with this JQE that will be posted.

- Record 4:

  DEVICES SCANNED - The number of JES2 devices which were scanned to see if they could process this output.

  DEVICES POSTED - The corresponding number of devices that were awakened because they are able to process the output.

- Record 5:

  PSO WRITERS SCANNED - The number of PSO writers which were scanned to see if they could process this output.

  PSO WRITERS POSTED - The corresponding number of writers that were awakened because they are able to process the output.

- Record 6:

  SAPI WRITERS SCANNED - The number of SAPI writers which were scanned to see if they could process this output.

  SAPI WRITERS POSTED - The corresponding number of writers that were awakened because they are able to process the output.

- Record 7:

  WORK SELECTION CALLS - The total number of calls to the work selection routine on this $#POST.

- Record 8:

  CPU TIME USED (SEC) - The amount of CPU time spent doing processing for the $#POST macro
- Record 9:

  $#POST CALLED BY - The caller of the $#POST macro and the offset where the call was made

Note: Most of the preceding information is issued only for $#POST TYPE=JQE and TYPE=JOE. Abbreviated $TRACE records are issued for $#POST TYPE=XMIT and $#POST TYPE=MSG.

Trace ID=31 sample

Figure 32 on page 76 is a sample of output from trace identifier 31. Trace identifier 31 traces all $QGET macro calls made by JES2 devices and initiators. The trace provides counts such as the number of jobs scanned, the number of $DOGJQE calls made, the amount of CPU time spent in the $QGET service, and the amount of CPU time spent in $EXITs 14, 49, or both. For initiators, the applicable initiator information such as initiator id and jobclass or service class information.

This record provides a means to analyze selection criteria and queue search overhead to tune work selection criteria.

```
15.22.48.89   ID =  31 $QGET    JOB00408   EXEC      05EC6160  $QGET CALL FOR JES INIT(2) NAME=2  -----------------
                JES INITIATOR CLASS LIST    = AB
                JQES DEFINED         =      500  JQES IN USE          =      221
                JQES SCANNED         =       33  $DOGJQE CALLS        =       31
                $QGET RETURN CODE    =        0  ELEMENT SELECTED     =       33
                CPU TIME USED (SEC)  =     0.001346
                $QGET    CALLED BY   = HASPXEQ  00160000 + 000990
15.23.01.98   ID =  31 $QGET               EXEC      05EC6160  $QGET CALL FOR JES INIT(10) NAME=10   ----------------
                JES INITIATOR CLASS LIST    = A
                JQES DEFINED         =      500  JQES IN USE          =      221
                JQES SCANNED         =       32  $DOGJQE CALLS        =       30
                $QGET RETURN CODE    =        4
                CPU TIME USED (SEC)  =     0.001237
                $QGET    CALLED BY   = HASPXEQ  00160000 + 000990
15.23.34.31   ID =  31 $QGET    JOB00375   EXEC      05EC6160  $QGET CALL FOR WLM INIT SRVCLASS=DISCRETN ------------
                JQES DEFINED         =      500  JQES IN USE          =      186
                JQES SCANNED         =        1  $DOGJQE CALLS        =        1
                $QGET RETURN CODE    =        0  ELEMENT SELECTED     =        1
                CPU TIME USED (SEC)  =     0.000087
                CPU TIME USED (X49)  =     0.000004
                EXIT 49 SKIPPED JQES =        0
                $QGET    CALLED BY   = HASPXEQ  00160000 + 001332
15.24.46.56   ID =  31 $QGET               EXEC      05EC6160  $QGET CALL FOR WLM INIT SRVCLASS=DISCRETN ------------
                JQES DEFINED         =      500  JQES IN USE          =      181
                JQES SCANNED         =        0  $DOGJQE CALLS        =        0
                $QGET RETURN CODE    =        4
                CPU TIME USED (SEC)  =     0.000008
                $QGET    CALLED BY   = HASPXEQ  00160000 + 001332
15.24.55.46   ID =  31 $QGET    JOB00411   CNVT      05EC6830  $QGET CALL FOR QUEUE=CNVT     ------------------------
                JQES DEFINED         =      500  JQES IN USE          =      217
                JQES SCANNED         =       32  $DOGJQE CALLS        =       31
                $QGET RETURN CODE    =        0  ELEMENT SELECTED     =       32
                CPU TIME USED (SEC)  =     0.001217
                $QGET    CALLED BY   = HASPCNVT 00049000 + 0000EE
15.24.57.57   ID =  31 $QGET               CNVT      05EC6830  $QGET CALL FOR QUEUE=CNVT     ------------------------
                JQES DEFINED         =      500  JQES IN USE          =      218
                JQES SCANNED         =       30  $DOGJQE CALLS        =       30
                $QGET RETURN CODE    =        4
                CPU TIME USED (SEC)  =     0.001156
                $QGET    CALLED BY   = HASPCNVT 00049000 + 0000EE
15.24.59.54   ID =  31 $QGET    JOB00408   HOPE      05ECBB88  $QGET CALL FOR QUEUE=OUTPUT   ------------------------
                JQES DEFINED         =      500  JQES IN USE          =      221
                JQES SCANNED         =       16  $DOGJQE CALLS        =       16
                $QGET RETURN CODE    =        0  ELEMENT SELECTED     =       16
                CPU TIME USED (SEC)  =     0.000649
                $QGET    CALLED BY   = HASPHOPE 0006C000 + 0000C6
15.25.02.01   ID =  31 $QGET               HOPE      05ECBB88  $QGET CALL FOR QUEUE=OUTPUT   ------------------------
                JQES DEFINED         =      500  JQES IN USE          =      221
                JQES SCANNED         =       15  $DOGJQE CALLS        =       15
                $QGET RETURN CODE    =        4
                CPU TIME USED (SEC)  =     0.000668
                $QGET    CALLED BY   = HASPHOPE 0006C000 + 0000C6
15.25.35.41   ID =  31 $QGET    STC00340   PURGE     0601F470  $QGET CALL FOR QUEUE=PURGE    ------------------------
                JQES DEFINED         =      500  JQES IN USE          =      222
                JQES SCANNED         =        9  $DOGJQE CALLS        =        9
                $QGET RETURN CODE    =        0  ELEMENT SELECTED     =        9
                CPU TIME USED (SEC)  =     0.000385
                $QGET    CALLED BY   = HASPTRAK 00153000 + 0019FA
15.25.35.41   ID =  31 $QGET               PURGE     0601F5E0  $QGET CALL FOR QUEUE=PURGE    ------------------------
                JQES DEFINED         =      500  JQES IN USE          =      222
                JQES SCANNED         =        9  $DOGJQE CALLS        =        8
                $QGET RETURN CODE    =        4
                CPU TIME USED (SEC)  =     0.000315
                $QGET    CALLED BY   = HASPTRAK 00153000 + 0019FA
15.26.42.64   ID =  31 $QGET    JOB00406   ROUT.JT1  06034378  $QGET CALL FOR QUEUE=XMIT     ------------------------
                JQES DEFINED         =      500  JQES IN USE          =      213
                JQES SCANNED         =       26  $DOGJQE CALLS        =        1
                $QGET RETURN CODE    =        0  ELEMENT SELECTED     =       26
                CPU TIME USED (SEC)  =     0.000076
                $QGET    CALLED BY   = HASPNJT  0009B000 + 0001B4
```

Figure 32. Sample Output from Trace Identifier 31

```
15.27.19.78  ID =  31 $QGET              ROUT.JT1  06034378  $QGET CALL FOR QUEUE=XMIT     ------------------------
               JQES DEFINED         =      500  JQES IN USE          =      222
               JQES SCANNED         =       28  $DOGJQE CALLS        =        0
               $QGET RETURN CODE    =        4
               CPU TIME USED (SEC)  =     0.000013
               $QGET    CALLED BY   = HASPNJT  0009B000 + 0001B4
15.28.24.71  ID =  31 $QGET    JOB00426   L15.JT1   06034100  $QGET CALL FOR L15.JT1           --------------------
               WS = (/)
               WORK SELECTION MASK = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFF
               JQES DEFINED         =      500  JQES IN USE          =      225
               JQES SCANNED         =      190  $DOGJQE CALLS        =      137
               JQES THROUGH WS      =      121
               $QGET RETURN CODE    =        0  ELEMENT SELECTED     =      190
               CPU TIME USED (SEC)  =     0.001694
               $QGET    CALLED BY   = HASPNJT  0009B000 + 00015C
15.29.43.62  ID =  31 $QGET              L15.JT1   06034100  $QGET CALL FOR L15.JT1           --------------------
               WS = (/)
               JQES DEFINED         =      500  JQES IN USE          =      182
               JQES SCANNED         =      132  $DOGJQE CALLS        =      104
               JQES THROUGH WS      =        0
               $QGET RETURN CODE    =        4
               CPU TIME USED (SEC)  =     0.001263
               $QGET    CALLED BY   = HASPNJT  0009B000 + 00015C
```

Trace record contents

Table 38 is the sample record broken into fields which are described after the table.

Table 38. Fields for Trace 31

RECORD 1						
1	**2**	**3**	**4**	**5**	**6**	**7**
15.22.48.89	ID = 31	$#GET	JOB00408	EXEC	05C6160	$QGET CALL FOR JES INIT(2) NAME=2
RECORD 2						
WS = (/)						
RECORD 3						
WORK SELECTION MASK = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFF						
RECORD 4						
JES INITIATOR CLASS LIST = AB						
RECORD 5						
JQES DEFINED = 500 JQES IN USE = 221						
RECORD 6						
JQES SCANNED = 33 $DOGJQE CALLS = 31						
RECORD 7						
JQES THROUGH WS = 0						
RECORD 8						
$QGET RETURN CODE = 0 ELEMENT SELECTED = 33						
RECORD 9						
OPTIMIZATION ALLOWED = YES						
RECORD 10						

Table 38. Fields for Trace 31 (continued)

RECORD 1
CPU TIME USED (SEC) = 0.001346
RECORD 11
$QGET CALLED BY = HASPXEQ 00160000 + 000990

The trace output contains the following:

- Record 1:

Field	Contents
1	Time-of-day clock value when the $TRACE was processed.
2	Trace identifier.
3	Function.
4	Job name, if available.
5	Name of processor or device which issued the $QGET
6	PCE address of processor which issued the $TRACE
7	Description of $QGET call

- Record 2:

  The WS= field contains the work selection list for the device. It is included only by those devices which use work selection lists, namely offload and network job transmitters.

- Record 3:

  WORK SELECTION MASK - the work selection mask corresponding to the $JQE selected. It is included only by those devices which use work selection lists, namely offload and network job transmitters, and then only when work is actually selected.

- Record 4:

  JES INITIATOR CLASS LIST - specifies the class list that was passed to the JES initiator on the $QGET call. Note that this list may not necessarily include all of the classes defined to the initiator for certain types of calls. The class list is only included in the trace for $QGET calls from JES initiators.

- Record 5:

  JQES DEFINED - total number of JQEs defined (JOBDEF JOBNUM) JQES IN USE - total number of JQEs currently in use.

- Record 6:

  JQES SCANNED - number of jobs searched before work was found. $DOGJQE CALLS - number of jobs for which it was necessary to call the $DOGJQE service to determine whether the work is selectable.

- Record 7:

  JQES THROUGH WS - The number of jobs for which the work selection routine was called. It is included only by those devices which use work selection lists, namely offload and network job transmitters.

- Record 8:

  $QGET RETURN CODE - The return code from the $QGET service. 0 indicates that work was was selected; 4 indicates that no work was found. ELEMENT SELECTED - The element selected out of all the jobs that were scanned. This is only included for return code 0.

- Record 9:

  OPTIMIZATION ALLOWED= indicates whether class list optimization is allowed for JES initiators, or service class optimization is allowed for WLM initiators. One of the following values will be displayed:
  - YES - indicates optimization is allowed.
  - YES (X14) - indicates that $EXIT 14 turned on class list optimization.
  - NO - indicates optimization is not allowed.
  - NO (X49) - indicates that $EXIT 49 turned off optimization.
  - NO (UNKNOWN) - indicates that optimization has been turned off for some unknown reason.
- Record 10:

  CPU TIME USED (SEC) - The amount of CPU time spent doing processing for the $QGET macro.
- Record 11:

  $QGET CALLED BY - The caller of the $QGET macro and the offset where the call was made.

The trace may also contain the following information about installation exits taken out of the $QGET service:

- CPU TIME USED (X14) - The amount of CPU time spent in installation $EXIT 14. This data does not appear if $EXIT 14 was not entered.
- EXIT 14 RETURN CODE - The return code that was returned by $EXIT 14. This data does not appear if $EXIT 14 was not entered.
- CPU TIME USED (X49) - The amount of CPU time spent in installation $EXIT 49. This data does not appear if $EXIT 49 was not entered.
- EXIT 49 SKIPPED JQES - The number of JQEs for which $EXIT 49 indicated that the job is not selectable. This data does not appear if $EXIT 49 was not entered.

Trace ID=32 sample

Figure 33 is a sample of output from trace identifier 32. Trace identifier 32 traces all $#REM macro calls made by any JES2 PCE. The trace provides the Work JOE (JOETYPE=X'80') and an indication of whether the $#REM was done as part of user exit processing or by non-exit processing.

This record provides a means of determining when a JOE is removed and whether the code requesting the removal is likely to be IBM code or not.

```
16.58.17.07  ID =  32 $#REM    STC00008   L19.ST1   06B16180  JOE REMOVAL -- EXIT NUMBER IN CONTROL 1 ------------------
        0  0CE00080 FD9C6EFF FFFFFFFF FFFFFF07   0E000000 000000FF FFFFFFFF FFFFFFFF  *.............................*
       20  FFFFFF00 FC55C0FF FFFFFFFF FFFFFF04   0C000081 1306B804 0C000080 FFE10000  *...................a...........*
       40  08000081 9546A000 08000081 95596804   0C000081 141880FF FFFFFF00 00100400  *...an......an......a...........*
       60  02000000 04001600                                                        *........                       *
16.59.17.35  ID =  32 $#REM    STC00009   COMM      06B743C8  JOE REMOVAL ----------------------------------------------
        0  80000000 D88000F4 00000000 00000006   20000000 50000005 00000000 02000000  *....Q..4...........&...........*
       20  00000000 09000000 0000CD05 0000001D   00000000 00000000 00000000 0000CC04  *.............................*
       40  03000100 00010000 F1404040 40404040   00010001 B7CC7E5F C9C2D4E4 E2C5D940  *........1       ......=.IBMUSER *
       60  00000000 04000000                                                        *........                       *
```

Figure 33. Sample Output from Trace Identifier 32

Trace ID=33 sample

Figure 34 on page 81 is a sample of output from trace identifier 33. Trace identifier 33 traces all NJE and Spool Offload traffic either being transmitted/offloaded or received/reloaded. Specify TRACE=YES on the device and activate trace id 33 in order to have the data generated.

The field names in the record identify the sections of the record. The following sections are identified in the formatted trace output:

- JOB HEADER GENERAL SECTION
- JOB HEADER UNKNOWN GENERAL SECTION
- JES2 SECTION OF THE JOB HEADER
- JES2 SPOOL OFFLOAD SECTION OF THE JOB HEADER
- JES2 AFFINITY SECTION OF THE JOB HEADER
- UNKNOWN JES2 SECTION OF THE JOB HEADER
- JOB SCHEDULING SECTION OF THE JOB HEADER
- UNKNOWN JOB SCHEDULING SECTION OF THE JOB HEADER
- SECURITY SECTION OF THE JOB HEADER
- UNKNOWN SECURITY SECTION OF THE JOB HEADER
- ACCOUNTING SECTION OF THE JOB HEADER
- UNKNOWN ACCOUNTING SECTION OF THE JOB HEADER
- RSCS SECTION OF THE JOB HEADER
- JES3 SECTION OF THE JOB HEADER
- POWER SECTION OF THE JOB HEADER
- USER SECTION OF THE JOB HEADER
- UNKNOWN SECTION OF THE JOB HEADER
- DATA SET HEADER GENERAL SECTION
- 3800 SECTION OF THE DATA SET HEADER
- RECORD CHARACTERISTICS CHANGE SECTION
- DATA SET HEADER UNKNOWN GENERAL SECTION
- DATA STREAM CHARACTERISTICS SECTION OF THE DATA SET HEADER
- UNKNOWN DATA STREAM CHARACTERISTICS SECTION OF THE DATA SET HEADER
- SECURITY SECTION OF THE DATA SET HEADER
- UNKNOWN SECURITY SECTION OF THE DATA SET HEADER
- JES2 SPOOL OFFLOAD SECTION OF THE DATA SET HEADER
- JES2 TP OFFLOAD SECTION OF THE DATA SET HEADER
- UNKNOWN JES2 SECTION OF THE DATA SET HEADER
- RSCS SECTION OF THE DATA SET HEADER
- JES3 SECTION OF THE DATA SET HEADER
- POWER SECTION OF THE DATA SET HEADER
- USER SECTION OF THE DATA SET HEADER
- UNKNOWN SECTION OF THE DATA SET HEADER
- JOB TRAILER GENERAL SECTION
- JOB TRAILER UNKNOWN GENERAL SECTION
- JES2 SPOOL OFFLOAD SECTION OF THE JOB TRAILER
- UNKNOWN JES2 SECTION OF THE JOB TRAILER

- ACCOUNTING SECTION OF THE JOB TRAILER
- UNKNOWN ACCOUNTING SECTION OF THE JOB TRAILER
- RSCS SECTION OF THE JOB TRAILER
- JES3 SECTION OF THE JOB TRAILER
- POWER SECTION OF THE JOB TRAILER
- USER SECTION OF THE JOB TRAILER
- UNKNOWN SECTION OF THE JOB TRAILER

```
16.58.15.86  ID =  33 NJEHDR   STC00008   L19.ST1   06B16180  TRANSMIT JOB HEADER     ---------------------------------
JOB HEADER PREFIX
           NJHLEN=0170 NJHFLAGS=00 NJHSEQ=00
JOB HEADER GENERAL SECTION
           JOBNAME=DEALLOC  JOB NUMBER=8  ORIGIN=POK     IBMUSER  EXECUTION=POK     IBMUSER
        0  00D40000 0008D0D8 400F0401 00000000   00000000 00000000 C4C5C1D3 D3D6C340  *.M.....Q ..............DEALLOC *
       20  00000000 00000000 00000000 00000000   00000000 00000000 B7CC7E27 DE3310A4  *.........................=....u*
       40  D7D6D240 40404040 C9C2D4E4 E2C5D940   D7D6D240 40404040 C9C2D4E4 E2C5D940  *POK     IBMUSER POK     IBMUSER *
       60  D7D6D240 40404040 40404040 40404040   D7D6D240 40404040 40404040 40404040  *POK             POK             *
       80  E2E3C440 40404040 00000002 00015180   3B9AC618 000F423F 40404040 40404040  *STD     ..........F.....        *
       A0  40404040 40404040 40404040 40404040   40404040 40404040 40404040 40404040  *                                *
       C0  40404040 0000001D 00000008 00000000   00000000                           *    ................            *
JES2 SECTION OF THE JOB HEADER
        0  00348400 01000000 00000000 00000000   00000000 00000000 00000000 00000000  *..d............................*
       20  00000000 00000000 00000000 00000000   00000000                           *...................              *
JOB SCHEDULING SECTION OF THE JOB HEADER
        0  000C8A00 000F423F 7FFFFD78                                               *............                     *
SECURITY SECTION OF THE JOB HEADER
        0  00588C00 00048000 50012204 0000C000   00000000 00000000 D7D6D240 40404040  *........&..............POK     *
       20  C9C2D4E4 E2C5D940 D7D6D240 40404040   00000000 00000000 E2E3C3C9 D5D9C4D9  *IBMUSER POK     ........STCINRDR*
       40  00000000 00000000 C9C2D4E4 E2C5D940   E2E8E2F1 40404040                   *........IBMUSER SYS1            *
16.58.15.86  ID =  33 NJEHDR   STC00008   L19.ST1   06B16180  TRANSMIT DATA SET HEADER ---------------------------------
DATA SET HEADER PREFIX
           NDHLEN=00EC NDHFLAGS=00 NDHSEQ=00
DATA SET HEADER GENERAL SECTION
        0  00740000 C3C1D3C1 C3404040 00000000   00000000 40404040 40404040 D1C5E2F2  *....CALAC   ........        JES2*
       20  40404040 D1C5E2D4 E2C7D3C7 000200D8   0000000F 2AC40085 01000000 00000000  *    JESMSGLG...Q.....D.e........*
       40  00000000 00000000 00000000 00000000   00000000 40404040 40404040 D1C5E2D4  *....................        JESM*
       60  E2C7D3C7 80800000 D3C9D5C5 40404040   00000000                           *SGLG....LINE    ....            *
SECURITY SECTION OF THE DATA SET HEADER
        0  00588C00 00040000 50012204 0000C000   00000000 00000000 D7D6D240 40404040  *........&..............POK     *
       20  C9C2D4E4 E2C5D940 D7D6D240 40404040   00000000 00000000 E2E3C3C9 D5D9C4D9  *IBMUSER POK     ........STCINRDR*
       40  00000000 00000000 C9C2D4E4 E2C5D940   E2E8E2F1 40404040                   *........IBMUSER SYS1            *
DATA STREAM CHARACTERISTICS SECTION OF THE DATA SET HEADER
        0  001C8900 001C0000 00000000 00000000   00000000 40404040 40404040           *..i.................            *
        0  001C8900 001C0000 00000000 00000000   00000000 40404040 40404040           *..i.................            *
16.58.16.03  ID =  33 NJEHDR   STC00008   L19.ST1   06B16180  TRANSMIT JOB TRAILER     ---------------------------------
JOB TRAILER PREFIX
           NJTLEN=0040 NJTFLAGS=00 NJTSEQ=00
JOB TRAILER GENERAL SECTION
        0  00300000 00D00000 B7CC7E28 07BC1308   B7CC7E28 39935E83 00000000 0000001D  *.........=.......=..l.c........*
       20  00000000 00000000 000F0100 41000000                                       *................                 *
ACCOUNTING SECTION OF THE JOB TRAILER
        0  000C8900 00000000 00000000                                               *..i.........                    *
```

Figure 34. Sample Output from Trace Identifier 33

Trace ID=34 sample

Figure 35 on page 82 is a sample of output from trace identifier 34. Trace identifier 34 traces all data records passed between the JES2 address space and the NETSRV address space. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).

```
19.47.00.657 ID =  34 NJETCP   ASID 0029           004FF1F8  NETSRV1   LINE25       ------------------------------
TRACE OF IAZYTNMS  FROM JES2-MAIN    TO JES2-NETSRV
QUEUE TIME:      1.979265
11CC  F0C929D7 D6D24040 40404001 00000000   00648000 40404040 40404040 40404040  *0I.POK     .........
11EC  40404040 00150000 00                                                 *    .....                     *
```

Figure 35. Sample Output from Trace Identifier 34

Trace ID=35 sample

Figure 36 is a sample of output from trace identifier 35. Trace identifier 35 traces all control records passed between the JES2 address space and the NETSRV address space. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).

Note: The address after the ASID xxxx field is the TCB address of the task creating the trace record.

```
19.47.00.667 ID =  35 NJETCP   ASID 0029 004FF1F8  NETSRV1   LINE25 ----------------------------------
TRACE OF IAZYTNRQ  FROM JES2-MAIN    TO JES2-NETSRV
REQUEST TYPE:  NRQTYPE_ENABLE_SESSP
BUFFER SIZE=32768  FEAT=15000000  JTNUM=3  JRNUM=1  STNUM=3  SRNUM=4
QUEUE TIME:      0.002303
```

Figure 36. Sample Output from Trace Identifier 35

Trace ID=36 sample

Figure 37 is a sample of output from trace identifier 36. Trace identifier 36 traces all data records passed between JES2 code in the NETSRV address space and IAZNJTCP. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).

Note: The address after the ASID xxxx field is the TCB address of the task creating the trace record.

```
19.47.05.636 ID =  36 NJETCP ASID 0029 004E72B8  NETSRV1   L25.JT1      ---------------------------------
TRACE OF JOB-HDR   FROM JES2-NETSRV  TO IAZNJTCP
8C99F28  017B0000 00D40000 0016C1C1 40090201   00000000 00000000 00000000 C9C2D4E4  *.#...M....AA ..............IBMU*
8C99F48  E2C5D9F3 00000000 00000000 E6C1E2C9   D2404040 00000000 00000000 BE9642E3  *SER3........WASIK   .........o.T*
8C99F68  C9226800 D7D6D240 40404040 C9C2D4E4   E2C5D940 E6E2C340 40404040 40404040  *I...POK     IBMUSER WSC         *
8C99F88  40404040 D7D6D240 40404040 40404040   40404040 D7D6D240 40404040 40404040  *    POK             POK         *
8C99FA8  40404040 E2E3C440 40404040 0000000F   00000078 00000FA0 00000064 40404040  *    STD     .................    *
8C99FC8  40404040 40404040 40404040 40404040   40404040 40404040 40404040 40404040  *                                 *
8C99FE8  40404040 40404040 00000000 00000016   00000000 00000000 00348400 00000000  *        ..................d.....*
8C9A008  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  *................................*
8C9A028  00000000 00000000 00000000 000C8A00   00000028 05F5DD18 000B8D00 00000008  *.....................5..........*
8C9A048  00010000 588C0000 04000050 01320700   03C00000 00000000 000000D7 D6D24040  *...........&..............POK  *
8C9A068  40404000 00000000 00000000 00000000   00000000 00000000 000000C9 D5E3D9C4  *   .......................INTRD*
8C9A088  D9404000 00000000 000000C9 C2D4E4E2   C5D940E2 E8E2F140 404040            *R  ........IBMUSER SYS1         *
```

Figure 37. Sample Output from Trace Identifier 36

Trace ID=37 sample

Figure 38 on page 83 is a sample of output from trace identifier 37. Trace identifier 37 traces all control records passed between JES2 code in the NETSRV address

space and IAZNJTCP. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).

Note: The address after the ASID xxxx field is the TCB address of the task creating the trace record.

```
19.47.17.752 ID = 37 NJETCP ASID 0029 004FF1F8 NETSRV1 LINE25 ---------------------------------
TRACE OF IAZYTNRQ  FROM JES2-NETSRV  TO IAZNJTCP
REQUEST TYPE:  NRQTYPE_HALT_CONN
```

Figure 38. Sample Output from Trace Identifier 37

Trace ID=38 sample

Figure 39 is a sample of output from trace identifier 38. Trace identifier 38 traces all data records passed between IAZNJTCP and TCP/IP. These records are traced if tracing is requested on the line (TRACE=COMMON=YES on the LINE statement) or NETSRV (TRACE=COMMON=YES on the NETSRV statement).

Note: The address after the ASID xxxx field is the TCB address of the task creating the trace record.

```
19.47.41.473 ID = 38 NJETCP ASID 0029 004E7120  NETSRV1   LINE25       ---------------------------------
TRACE OF DATA FROM IAZNJTCP     TO TCP/IP
8C67990  C1C3D240 40404040 D7D6D240 40404040   00000000 E6E2C340 40404040 093901E1  *ACK     POK     ....WSC     ....*
8C679B0  00                                                                       *.                           *
```

Figure 39. Sample Output from Trace Identifier 38

Trace ID=39 sample

Figure 40 is a sample of output from trace identifier 39. Trace identifier 39 traces TCP/IP API calls issued by IAZNJTCP. These records are traced if tracing is requested on the line (TRACE=JES=YES on the LINE statement) or NETSRV (TRACE=JES=YES on the NETSRV statement).

```
19.47.41.474 ID = 39 NJETCP ASID 0029 004E7120 NETSRV1 LINE25 ---------------------------------
INPUT TO TCP/IP SERVICE: RECV
8CE3D78  00000080 00000000 00000000 08C5B650   0000000A 00000000 00000000 00000000  *.............E.&.................*
8CE3D98  00000000 00000000 00000000 00000000   00000000 00000086 00010400 00000000  *........................f........*
8CE3DB8  00000000 00000000 0000044F 00000000   00000040 00000000 00000000 00000000  *...................  .............*
8CE3DD8  08B1F430 00000010 08B1F440 00000000   00000000 00000000 00000000 00000000  *..4.......4 ....................*
8CE3DF8  00000000 00                                                              *.....                       *
```

Figure 40. Sample Output from Trace Identifier 39

Trace ID=40 sample

Figure 41 on page 84 is a sample of output from trace identifier 40. Trace identifier 40 traces the results of the WLM initiator balancing computation at the beginning of every checkpoint cycle.

```
10.11.02.204 ID =  40 WLMGOALS            CKPT      087DE0B8  SERVICE Class -- BESTEVER
--------------------------------
MEMBER QAFF    SATURATED  MAFF  SAFF  IACT  JACT  GACT  STLN  MAFG

M008   Y       N             0     0     0     0 UNLIM UNLIM UNLIM
CHECKPOINT DATA - HOLD=30 MINDORM=100 MAXDORM=500 ACTUAL HOLD=30 ACTUAL DORMANCY=119
```

Figure 41. Sample Output from Trace Identifier 40

The trace output contains the following:

- Member is the member name for the line of data.
- QAFF is "Y" if the member is included in the QAFF for the service class.
- SATURATED is "Y" if the member is at or above its goal for multi-member-affinity jobs.
- MAFF is the number of jobs eligible for the member that also have affinity for other active members.
- SAFF if the number of jobs eligible for the member that do not have affinity for other active members.
- IACT is the number of WLM initiators active on the member.
- JACT is the number of WLM managed batch jobs active on the member.
- GACT is the goal for active jobs on the member (UNLIM means that the goal is not limited).
- STLN is the number of jobs that have affinity to the member and are "allowed" to be "stolen" by other members.
- MAFG is the number of multi-affinity jobs that can be selected on this member (UNLIM means not limited). This value shows only for the member writing the trace record.
- HOLD is from MASDEF HOLD=.
- MINDORM is from MASDEF DORMANCY=(xxxx).
- MAXDORM is from MASDEF DORMANCY=(,xxxx).
- ACTUAL HOLD and ACTUAL DORMANCY are the values for the current checkpoint cycle.

Trace ID=41 sample

Figure 42 on page 85 is a sample of output from trace identifier 41. Trace identifier 41 traces the results of the sampling data computation that is passed to WLM at the end of every checkpoint cycle.

```
10.11.02.509 ID =  41 WLMBQS                        002DD0B8  JOBQSAMP
-----------------------------------------------
Service Class - BESTEVER Token - 010C8001

Registered - (BCG1,M008)

Sampling Data - SYS_QUEUE=0 SYS_INEL=0 SYS_LIMIT=0

LOCAL_QUEUE=0 CONSTRAINT_AFFINITY=0 LOCAL_INEL=0

Service Class - DISCRETN Token - 080C8001

Registered - (BCG1,M008)

Sampling Data - SYS_QUEUE=0 SYS_INEL=45 SYS_LIMIT=0

LOCAL_QUEUE=0 CONSTRAINT_AFFINITY=0 LOCAL_INEL=45

Service Class - FAST     Token - 090C8001

Registered - (BCG1,M008)

Sampling Data - SYS_QUEUE=0 SYS_INEL=1 SYS_LIMIT=0

LOCAL_QUEUE=0 CONSTRAINT_AFFINITY=0 LOCAL_INEL=1
```

Figure 42. Sample Output from Trace Identifier 41

The trace output contains the following:

Service Class
The service class name

Token An internal value used by WLM

Registered
The JES2 MAS members on which the service class has been registered with WLM

Sampling data
Data used by WLM to manage the batch backlog. The sampling data is for the Service class named on the Service class line.

SYS_QUEUE
MAS queue delay. Jobs eligible to be initiated somewhere in the MAS.

SYS_INEL
MAX ineligible. Jobs not eligible to be initiated on any member in the MAS. This includes operator held jobs, jobs held because of duplicate Name, jobs busy on a JES transmitter (offload and keep).

SYS_LIMIT
MAX limited. Jobs are not eligible because the JES job queue or queues that are related to these jobs have reached their limit of executing jobs. See XEQCOUNT and XEQMEMBER on the JOBCLASS initialization statement.

LOCAL_QUEUE
Local queue delay. Jobs eligible to be initiated on this system.

CONSTRAINT_AFFINITY
Constraint affinity. Jobs eligible to be initiated on constraint systems only.

LOCAL_INEL
Local ineligible. Jobs not eligible to be initiated on this member.

This includes operator held jobs, jobs held because of duplicate name, jobs busy on a JES transmitter (offload and keep).

Trace ID=42 sample

Figure 43 on page 87 is sample output from trace identifier 42. Trace identifier 42 traces all $CDCTDYN macro calls that are made by JES2 processing.

```
------------------------------------------------------------
16.27.55.24950 ID= 42 $CDCTDYN           COMM    0BCB2688   Parms = 00000000 0C2E1980 00022938 01600000 ------------
  CDCT
       0  000001EF 81DCA040 C3C4C3E3 C0E00000 000001EF 81DC9C60 000001EF 81DCA420 *....a.. CDCT........a..-....a.u.*
      20  000001EF 81DCA420 00000000 00000000 000001EF 81DC90C0 000001EF 81DCF1A0 *....a.u.............a.......a.1.*
      40  00000000 00000000 00000000 00000000 000001EF 81E01000 CABDDE72 EBB5B618 *....................a...........*
      60  D7D9E3F1 40404040 40400120 00010000 00000000 00000000 00000000 00000000 *PRT1      ......................*
      80  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
      A0  00000000 00000000 F0404040 00194DE6 6BD86BD9 6BD7D9D4 6BD3C9D4 61C66BE4 *........0   ..(W,Q,R,PRM,LIM/F,U*
      C0  C3E26BC6 C3C25D40 40404040 40404040 40404040 40404040 40404040 40404040 *CS,FCB)                          *
      E0  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *                                *
     100  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *                                *
     120  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *                                *
     140  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *                                *
     160  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *                                *
     180  40404040 40404040 40404040 40404040 40000009 2E012609 2A28050A 03000000 *                 ...............*
     1A0  00000000 00000000 00000000 00000100 00404040 40404040 40000000 00000000 *................        .......*
     1C0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000C1C2 *............................AB*
     1E0  C3D3E7E9 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *CLXZ                            *
     200  40404000 00000000 00000000 00000000 00000000 00000000 00000000 0000E2E3 *   ............................ST*
     220  C4404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *D                               *
     240  40404040 40404040 40404040 40404040 40404040 40404040 40404040 4040D3C9 *                              LI*
     260  D5C54040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *NE                              *
     280  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *                                *
     2A0  40404040 40404040 40404040 40404040 40404040 40405C5C 5C5C5C5C 5C5C5C5C *                       ***********
     2C0  5C5C5C5C 5C5C0000 0000005C 05010000 5101D7D9 E3F14040 40404040 F0F0F0F2 *******.....*......PRT1      0002*
     2E0  00000000 0800E2E8 F1404040 4040D5F1 D4F14040 4040E2E8 E2C8C9C7 C8400000 *......SY1     N1M1    SYSHIGH ..*
     300  24000000 00640000 00000003 5C5C5C5C 5C5C5C5C 9000C4D9 C1C9D5C5 C4400000 *............********..DRAINED ..*
     320  00000000 00000054 05020000 00000000 00000000 F6404040 5C5C5C5C 00000000 *.....................6   ****....*
     340  00000000 00000000 FFFFFFFF 00000000 FFFFFFFF 00000000 00000000 00000000 *................................*
     360  00000000 00000000 00000000 00000000 00000000 00000000 00000034 05200000 *................................*
     380  80200001 40000000 40404040 40404040 00000001 000F423F 40404040 40404040 *.... ...        ........        *
     3A0  00000000 00000000 00000000 00000030 05030000 00000000 00000000 40404040 *............................    *
     3C0  40404040 40404040 40404040 012CC000 00004040 40404040 40400000 00000000 *            ......        ......*
  DCT
       0  00000000 0C2E1980 C4C3E340 033C0000 0BCC6A00 52000140 00000000 00000000 *........DCT ........... ........*
      20  00000000 0BCC6A00 00002040 40000000 00000000 00000000 00000000 0C2E1640 *...........  .................. *
      40  D7D9E3F1 40404040 00EFF190 00000000 E2E8E2C8 C9C7C840 000001EF 81DCA040 *PRT1    ..1.....SYSHIGH ....a.. *
      60  00200001 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
      80  00000000 F0F0F0F2 00000000 00000000 00000000 00000000 00000000 00000000 *....0002........................*
      A0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
      C0  00000000 00000000 40404040 40404040 40404040 40404040 40404040 40404040 *........                        *
      E0  B1E80104 02090000 0BB8FBA0 0BB8F738 0BB8F860 0BB8F6A0 0BB8F578 0BB8FCB0 *.Y.............7...8-..6...5.....*
     100  0BB8F448 0BB8FA80 0BB8F328 00000000 00000000 00000000 00000000 00000000 *..4.......3....................*
     120  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
     140  00000000 00000000 00000000 0BB8B0BC 00010000 40404040 40404040 00000000 *....................        ....*
     160  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
     180  40404040 40404040 40404040 40404040 00000000 00000000 01800000 00000001 *                 ...............*
     1A0  000F423F 00000000 00000000 00000000 00000000 00000000 00000000 10000000 *................................*
     1C0  BDDE4C7C 00000000 00000000 D7D9E3F1 40404040 40404040 40404040 40202000 *..<@........PRT1             ...*
     1E0  01000000 00000000 FFFFFFFF E2E3C440 40404040 C1C2C3D3 E7E94040 40404040 *............STD     ABCLXZ      *
     200  40404040 40404040 40404040 40404040 40404040 40404040 40000000 00000000 *                              .......*
     220  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
     240  00000000 00000000 00000000 00000000 00000000 00000000 00000000 E2E3C440 *............................STD *
     260  40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 *                                *
     280  40404040 40404040 40404040 40404040 40404040 40404040 40404040 F6404040 *                              6   *
     2A0  F0404040 5C5C5C5C 5C5C5C5C 00000000 FFFFFFFF 00000000 40404040 40404040 *0   ********............        *
     2C0  01000000 00000000 00000000 000A0000 00000000 00000000 00000000 00000000 *................................*
     2E0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
     300  00000000 00000000 5C5C5C5C 00000000 00008908 00000000 5C5C5C5C 5C5C5C5C *........****......i.....*********
     320  5C5C5C5C 5C5C5C5C 5C5C5C5C 00000000 00640000 0000012C 00000000 00000000 **************.....................*
     340  00000000                                                                 *....                          *
```

Figure 43. Sample Output from Trace Identifier 42

The trace output contains the following:

- The parameter list that was passed to the $CDCTDYN macro. This includes information about the caller of the $CDCTDYN macro.
- The control blocks (CDCT/DCT/APT/SCK/RAT/XREQ) that are associated with the request. The control block data includes the address of the control block.

Trace ID=43 sample

Figure 44 is a sample of output from trace identifier 43. Trace identifier 43 traces every ENF58 event that is sent.

```
17.11.57.74920 ID= 43 ENF58    JOB00024          004FF260   Member=IBM1 Type=SEND Token Change  ------------
Token type=Client  Bit Mask=FFF0FFF0 00060000
Job Key=C9AF89E2 IOT MTTR=00005409 Sort Key=404D0000 DS Key=00000002 Instance=1 OUTGRP=1.1.1
 Old CTKN  022A0100 404D0000 FFF00000 00060000 00000018 C9AF89E2 00000002 00005409  *.... (...0.........I.iS.......*
       20  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000100  *................................*
       40  00000000 00000000 00000000 00000080                                      *................                *
 New CTKN  022A0100 404D0000 FFF0FFF0 00060000 00000018 C9AF89E2 00000002 00005409  *.... (...0.0........I.iS.......*
       20  F1404040 40404040 00010001 00000000 00000000 00000000 00000000 00000100  *1       ........................*
       40  00000000 00000000 00000000 00000080                                      *................                *
$CTKNENF CALLED BY   = HASPHOPE 0B74E668 + 000D5E  APAR=OA38935
        0  C5D5C6F5 F8400419 00000100 0A00C9C2 D4F14040 40400000 00000000 00000000  *ENF58 ........IBM1    .........*
       20  00000000 00000000 022A0100 404D0000 FFF00000 00060000 00000018 C9AF89E2  *............ (...0.........I.iS*
       40  00000002 00005409 00000000 00000000 00000000 00000000 00000000 00000000  *................................*
       60  00000000 00000100 00000000 00000000 00000000 00000080 022A0100 404D0000  *............................ (..*
       80  FFF0FFF0 00060000 00000018 C9AF89E2 00000002 00005409 F1404040 40404040  *.0.0........I.iS........1      *
       A0  00010001 00000000 00000000 00000000 00000000 00000100 00000000 00000000  *................................*
       C0  00000000 00000080 C5D5C6F5 F8E70038 12561142 0112159F 00000000 00000000  *........ENF58X..................*
       E0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  *................................*
```

Figure 44. Sample Output from Trace Identifier 43

The trace output contains the following:

- The Job number, Job key, DS key, outgroup for which the ENF58 event was issued.
- CALLED BY = The module, offset, and APAR level of the module that is signalling the ENF58 event.
- The Member issuing the ENF58 event.

Trace ID=44 sample

Figure 45 is a sample of output from trace identifier 44. Trace identifier 44 traces the ENF58 event that is being received.

```
------------------------------------------------------------
17.11.57.74920 ID= 43 ENF58    JOB00024        004FF260   Member=IBM1 Type=RECEIVE Token Change          ------------
Token type=Client  Bit Mask=FFF0FFF0 00060000
Job Key=C9AF89E2 IOT MTTR=00005409 Sort Key=404D0000 DS Key=00000002 Instance=1 OUTGRP=1.1.1
 Old CTKN  022A0100 404D0000 FFF00000 00060000 00000018 C9AF89E2 00000002 00005409 *.... (...0.........I.iS........*
       20  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000100 *................................*
       40  00000000 00000000 00000000 00000080                                     *................                 *
 New CTKN  022A0100 404D0000 FFF0FFF0 00060000 00000018 C9AF89E2 00000002 00005409 *.... (...0.0........I.iS........*
       20  F1404040 40404040 00010001 00000000 00000000 00000000 00000000 00000100 *1       ........................*
       40  00000000 00000000 00000000 00000080                                     *................                 *

        0  C5D5C6F5 F8400419 00000100 0A00C9C2 D4F14040 40400000 00000000 00000000 *ENF58 ........IBM1    .........*
       20  00000000 00000000 022A0100 404D0000 FFF00000 00060000 00000018 C9AF89E2 *............ (...0.........I.iS*
       40  00000002 00005409 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
       60  00000000 00000100 00000000 00000000 00000000 00000080 022A0100 404D0000 *............................ (..*
       80  FFF0FFF0 00060000 00000018 C9AF89E2 00000002 00005409 F1404040 40404040 *.0.0........I.iS........1      *
       A0  00010001 00000000 00000000 00000000 00000000 00000100 00000000 00000000 *................................*
       C0  00000000 00000080 C5D5C6F5 F8E70038 12561142 0112159F 00000000 00000000 *........ENF58X..................*
       E0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
```

Figure 45. Sample Output from Trace Identifier 44

The trace output contains the following:

- The Job number, Job key, DS key and outgroup for which the ENF58 event was issued.
- The Member issuing the ENF58 event.

Note:

1. ENF580 events that are signalled by other members in the MAS are traced. ENF58 events that are signalled by the same member and ENF58 events that are signalled by members in another MAS are not traced.
2. Traces might not be available for secondary subsystems or if SYSLOG is not active.

Trace ID=45 sample

Figure 46 is a sample of output from trace identifier 45. Trace identifier 45 traces the ENF70 event that is being sent.

```
17.11.56.01740 ID= 45 ENF70    JOB00024 004AE7B8  IBMUSERA Member=IBM1 Type=SEND     Select   --------------
JESPLEX Id=12561142 0112159F Job Key=C9AF89E2 MAXCC=00000000 Selected by phase processing   Phase=Input
JOBCLASS=A        SRVCLASS=........
$JOBENF  CALLED BY   = HASCINJR 0B2AA000 + 0013B2  APAR=OA38935
        0  C5D5C6F7 F0400100 00000090 00780100 0000C9C2 D4F14040 40401256 11420112 *ENF70 ............IBM1    ......*
       20  159FC9C2 D4E4E2C5 D9C1D1D6 C2F0F0F0 F2F4D1D6 C2F0F0F0 F2F4D7D6 D2404040 *..IBMUSERAJOB00024JOB00024POK   *
       40  4040C140 40404040 40400000 00000000 00000000 0112159F 005E79B3 00000000 *  A       ..............;......*
       60  80000000 0000C9AF 89E20000 00180000 00050000 00000000 C5D5C6F7 F0E70018 *......I.iS..............ENF70X..*
       80  00000000 00000000 00000400 00000000                                     *.................           *
```

Figure 46. Sample Output from Trace Identifier 45

The trace output contains the following:

- The Job number, Job key, Job class and Service class of the Job whose phase change resulted in the ENF70 event that is being signalled.
- CALLED BY = The module, offset, and APAR level of the module that is issuing the ENF70 event.
- The JESPLEX ID and Member that is issuing the ENF70 event.

Trace ID=46 sample

Figure 47 is a sample of output from trace identifier 46. Trace identifier 46 traces the ENF70 event that is being received.

```
17.11.56.01740 ID= 45 ENF70    JOB00024           004AE7B8  IBMUSERA Member=IBM1 Type=RECEIVE Select ---------
JESPLEX Id=12561142 0112159F Job Key=C9AF89E2 MAXCC=00000000 Selected by phase processing  Phase=Input
JOBCLASS=A        SRVCLASS=........

        0  C5D5C6F7 F0400100 00000090 00780100 0000C9C2 D4F14040 40401256 11420112 *ENF70 ............IBM1    ......*
       20  159FC9C2 D4E4E2C5 D9C1D1D6 C2F0F0F0 F2F4D1D6 C2F0F0F0 F2F4D7D6 D2404040 *..IBMUSERAJOB00024JOB00024POK   *
       40  4040C140 40404040 40400000 00000000 00000000 0112159F 005E79B3 00000000 *  A       ..............;......*
       60  80000000 0000C9AF 89E20000 00180000 00050000 00000000 C5D5C6F7 F0E70018 *......I.iS..............ENF70X..*
       80  00000000 00000000 00000400 00000000                                     *.................           *
```

Figure 47. Sample Output from Trace Identifier 46

The trace output contains the following:

- The Job number, Job key, Job class and Service class of the Job whose phase change resulted in ENF70 being signalled.
- The JESPLEX ID and Member that is issuing the ENF70 event.

Note:

1. ENF58 events that are signalled by other members in the MAS are traced. ENF58 events that are signalled by the same member and ENF58 events that are signalled by members in another MAS are not traced.

2. Traces might not be available for secondary subsystems or if SYSLOG is not active.

Installation-defined trace events

Trace identifier (TID) tables can define new event trace identifiers or override JES2-defined trace identifiers. Use the $TIDTAB macro to create JES2 installation tables and table elements. Normal table pair processing extends the JES2 TID table with the installation-supplied table.

For details about using JES2 table pairs, see *z/OS JES2 Installation Exits*.

Creating a trace table using the $TRACE macro

Issue a $TRACE macro in an installation exit routine to record register information when the trace identifier is active, (assuming an entry in $TIDTAB for identifier 255) as follows:

```
      STM R0,R15,$REGSAVE
label $TRACE ID=255,LEN=16*4,DATA=$REGSAVE,NAME=$USER
```

The STM instruction stores registers 0 through 15 in storage at location $REGSAVE. The DATA parameter passes the location of the registers to the $TRACE macro.

For information on the $TRACE macro keywords and defining JES2 tables, see *z/OS JES2 Macros*.

Storage considerations

The PAGES and TABLES parameters of the TRACEDEF initialization statement specify the amount of storage that the trace facility can use. If you later determine that this amount of storage is inadequate, the operator can enter a $T TRACEDEF command to modify the number of trace tables ($T TRACEDEF,TABLES=n), or the size of the trace log data set ($T TRACEDEF,LOG=(SIZE=n)). You should be aware of the amount of storage being used for trace records to prevent total depletion of ECSA and CSA storage.

If, when logging trace information, JES2 cannot keep pace with the events being logged, JES2 discards the new data and the system issues message $HASP654. To correct this situation, either deactivate specific trace identifiers or increase the number of trace tables.

The LOG=(SIZE=n) subparameter on the TRACEDEF statement allows you to specify the maximum size that the trace log may attain before JES2 queues the log for printing. When JES2 logs a trace table, JES2 reuses the trace table.

Chapter 4. Using the JES2 DEBUG facility

You can use the DEBUG facility to trap unauthorized alterations of checkpoint-resident data, a job queue element (JQE), or a job output element (JOE) by specifying the $T DEBUG=Y|N command. It can also be used (if additional parameters are included with the command) to record certain JES2 events and activities. For example, you can specify whether to count certain events, provide certain $HASP095 error information to the operator, verify the integrity of a newly created checkpoint version, or, start or stop monitoring updates made to the checkpoint data set.

Because the DEBUG facility checks all checkpoint records before processing them, DEBUG causes performance degradation. Use the $T DEBUG=Y|N command only when you experience problems that you suspect are checkpoint I/O problems. Also, because of performance degradation, IBM suggests that you do not use the DEBUG command with the CKPT or VERSION parameters specified in a production environment.

DEBUG can detect:

- A failure to issue a $QSUSE or a $CKPT macro by another member in the multi-access spool (MAS) configuration.
- Random overlays of the 4-kilobyte pages, the job queue, and job output table (JOT). However, the facility cannot detect all unauthorized alterations. DEBUG will not detect the error if a routine changes a checkpointed element either without:
  - Issuing a $QSUSE macro. It is possible that the routine already has exclusive control of the checkpoint data set.
  - First issuing a $CKPT macro, and another a checkpoint element in the same block was altered validly during the same checkpoint cycle.
- Unauthorized updates to the checkpoint in a multi-access spool environment.
- Problems with the application copy of the checkpoint subtask.
- Access logging including warnings for JES2 output work selection for:
  - JES2 devices (such as local and remote printers and punches, and NJE and offload SYSOUT transmitters).
  - External writer (XWTR) requests.
  - SYSOUT application program interface (SAPI) requests (SSI function code 79).
  - With SECURITY=YES specified, you will receive RACF® messages, but no $HASP186 message for profiles in warn mode.

Starting and stopping the DEBUG facility

Start the DEBUG facility by specifying DEBUG=YES in the initialization stream or by entering a $T command with the DEBUG=YES parameter. When you want to turn off the DEBUG facility, enter a $T DEBUG=NO command. Display the status of the DEBUG facility by entering the $D DEBUG command. JES2 responds to the $D command with message $HASP827, to display the status of DEBUG.

If an error occurs, JES2 issues a $K01 error code.

A JES2 dump contains output from the DEBUG facility. In the dump, register 1 contains the checkpoint this JES2 image expected and register 14 contains the checkpoint as it appears on the checkpoint data set. The checkpoint on the data set should enable the installation to determine the cause of the error.

Determine why JES2 issued a $HASP186 message

Set the DEBUG facility to monitor security related processing by issuing a $TDEBUG,SECURITY=YES command when you receive a $HASP186 message with no corresponding ICH408I message. Perform the following action(s) as appropriate to generate another output selection:

- Modify the output group with an operator command.
- Drain the JES2 device. Start it again after the drain completes.
- Stop and restart the external writer address space.
- Stop and restart the JES2 address space.

If you do not perform any of the actions, JES2 will remember that the output group is not eligible and will not issue another SAF request. JES2 only reissues the SAF request when you take one of the appropriate actions listed and JES2 subsequently attempts to select the output group for the same device or devices.

After you've taken the appropriate action(s) and recreated the conditions while DEBUG SECURITY=YES is set, RACF will now log the access and issue messages such as the ICH408I message (RACF either issues the messages, or in cases requested by JES2, returns them to JES2 who then issues the messages). Use the RACF messages that accompany the $HASP186 message to determine why $HASP186 was issued.

Chapter 5. Using IPCS for diagnosis

The interactive problem control system (IPCS) provides an interactive, online facility for diagnosing software failures. IPCS formats and analyzes dumps to produce reports that can be viewed online or printed. A diagnostician can request specific information in the report based on:

- Class of output
- Control blocks
- Devices
- Job number
- Job queue
- Job output queue
- Job class
- Network characteristics
- Processors
- Subtasks
- MAS member data
- Checkpoint control blocks on both DASD and coupling facility structures.

JES2 support for IPCS is panel-driven, however, a few commands can be used in batch or in line mode.

Provide for JES2 IPCS Support

IPCS support is for dump analysis only and does not affect a running system. You can dynamically update or replace the JES2 IPCS parmlib member and IPCS panels and modules. If you make updates to these IPCS parts, and then actually use JES2 formatting options, you might need to return to TSO READY mode or even logon again, and you might also need to drop and re-initialize your dump.

Attention:

- You must be proactive to install JES2 IPCS.
- Make sure JES2 IPCS support works before you experience an emergency.
- Make sure you have JES2 IPCS support for all combinations of JES2 and MVS releases in production. (*z/OS JES2 Diagnosis* provides information about JES2 IPCS support.)

Dump level - diagnosing level considerations

- **Same MVS level, same JES2 level:** When you are diagnosing JES2 dumps on the same level of the system as the system on which the dumps were taken, do the following:
  1. Ensure that SHASPARM is specified in the MVS system PARMLIB concatenation.
  2. Ensure that SHASMIG is specified in the STEPLIB concatenation.
  3. Ensure that SHASPNL0 is specified in the ISPPLIB concatenation.

  This ensures that IPCS can find the SMP/E-maintained copy of JES2 data.

- **Same MVS level, different JES2 levels:** If the JES2 levels differ between the system on which the dump was taken and the system on which the dump will be examined using IPCS, but the level of MVS is the same, do the following:
  1. Add an IPCSPARM DD statement to your LOGON PROC, and specify the SHASPARM library that corresponds to the JES2 level in the dump in the first position in the IPCSPARM DD concatenation. Also specify the SYS1.PARMLIB and the SYS1.IBM.PARMLIB data sets, plus any other data sets containing IPCS parameters that you normally need when using IPCS.
  2. Specify the SHASMIG library that corresponds to the JES2 level in the dump in the first position in the STEPLIB concatenation.
  3. Specify the SHASPNL0 library that corresponds to the JES2 level in the dump in the first position in the ISPPLIB concatenation.
- **Different MVS levels, different JES2 levels:** If both the JES2 and system levels differ between the system on which the dump was taken and the system on which the dump will be examined using IPCS, do the following:
  1. Add an IPCSPARM DD statement to your LOGON PROC, and specify the SHASPARM library that corresponds to the JES2 level in the dump in the first position in the IPCSPARM DD concatenation. Also, specify the SYS1.PARMLIB and SYS1.IBM.PARMLIB data sets, plus any other data sets containing IPCS parameters that correspond to the system level in the dump.
  2. Specify the SHASMIG and MIGLIB libraries that correspond to the system in the dump in the first and second positions in the STEPLIB concatenation.
  3. Specify the SHASPNL0 and SBLSPNL0 libraries that correspond to the system in the dump in the first position in the ISPPLIB concatenation.

Note:

1. IPCS requires libraries in addition to the libraries that JES2 uses. You might need to provide other concatenations. For more information see *z/OS TSO/E Customization*.
2. If dump analysis leads into data structures for other elements or products, you might need to add more data sets to the concatenations to complete the diagnosis. Refer to the appropriate product publications for information about establishing the correct IPCS environments for diagnosing other elements and products.

JES2 IPCS support install verification

Your JES2 IPCS support may be at the wrong level if you are unable to use JES2 IPCS support to analyze a dump and experience any of the following:

- JES2 control block structure names aren't recognized
- Panels aren't available or they're out of date
- Dump data doesn't line up with JES2 formatters
- Numerous error messages are issued

Procedure to verify your JES2 IPCS support level

Use the following procedure to verify your JES2 IPCS support level. This procedure can detect many, but not all, install and service level problems.

Ensure you have the correct JES2 IPCS PARMLIB member

Go to IPCS option 2.6 (or issue **2.6;L JES2**) and scroll down until you find "JES2" as shown in the following example:

```
------------------ IPCS MVS DUMP COMPONENT DATA ANALYSIS ----------------------
OPTION ===>                                                 SCROLL ===> CSR

To display information, specify "S option name" or enter S to the left
of the option desired.  Enter ? to the left of an option to display
help regarding the component support.

S Name     Abstract
  IMSDUMP  IMS analysis
  IOSCHECK Active input/output requests
  IRLM     IMS Resource Lock Manager analysis
  JESXCF   JESXCF Address Space Analysis
  JES2     JES2 analysis for HJE7708, service level 0
  JES3D    JES3 analysis
  LISTEDT  Format eligible device table
  LLATRACE Library Lookaside trace
  LOGDATA  LOGREC formatter
  LOGGER   System logger formatter
  LPAMAP   Map link pack area
  MERGE    Merge GTF/CTRACE output
```

Ensure the JES2 FMID in the panel matches the JES2 FMID in the dump you are diagnosing.

Ensure that the JES2 service level in the dump you're diagnosing is within the service level range shown in the panel. See “Using IPCS panels to analyze JES2 data in a dump” on page 97 to determine the JES2 FMID and service level.

Note: The service level for a given JES2 release is set in the &J2SLVL assembly time variable. &J2SLVL is initially zero for each release and increases when significant changes are shipped in the service stream. See "Determining the JES2 Release Level" in *z/OS JES2 Installation Exits*.

Ensure you have the correct JES2 IPCS panel library

Go to IPCS option 2.6 (or issue **2.6;S JES2**) and select JES2 to display the primary JES2 panel as shown in the following example:

```
IPCS  JES2  Format  Trace  Debug
 ---------------------- JES2 Component Data Analysis --------------------------
 Option ===>                                                  Scroll ===> CSR
 Enter JES2 name ===> JES2

 Select desired option for JES2 dump:             These panels are for
    1 JES2 base display                           JES2 FMID:  HJE7720
    2 JES2 job control blocks                     Service level:  0
    3 JES2 job output control blocks
    4 JES2 devices
    5 JES2 processors
    6 JES2 subtasks
    7 JES2 control blocks
    8 JES2 NJE/RJE Control blocks
    9 JES2 MAS member data
   10 JES2 checkpoint control blocks
   11 JES2 BERT control blocks
   12 JES2 monitor data


 Select desired option for non-JES2 dump:
  101 Select JES2 control blocks for non-JES2 address space

 Debug JES2 IPCS support:
  999 Display module information for HASMFMTM and related modules
 1000 Set JES2 IPCS runtime debug options; current status is NORMAL

 Enter UP and DOWN commands to scroll the list of options.
 Enter END command to terminate JES2 data analysis.
```

Ensure the JES2 FMID in the panel matches the JES2 FMID in the dump you are diagnosing.

Ensure that the JES2 service level in the dump you're diagnosing is within the service level range shown in the panel. See “Using IPCS panels to analyze JES2 data in a dump” on page 97 to determine the JES2 FMID and service level.

Note: The IPCS service level for a given JES2 release is set in the &J2_IPCS_LEVEL assembly time variable. &J2_IPCS_LEVEL is initially zero for each release and increases with changes to either the JES2 control block mappings, data structures, or IPCS formatters that are changed significantly in the service stream. See "Determining the JES2 release level" *z/OS JES2 Installation Exits*

Ensure you have the correct JES2 SHASMIG library

To Display the modules linked with HASMFMTM, issue the following IPCS command:

- `IP VERBX HASMFMTM 'JES2,999'`

to obtain a display shown in the following example:

```
*** JES2 IPCS Install/Service Information ***

JES2 FMID = HJE7720
JES2 release = z/OS 1.7
JES2 product level = 36
JES2 IPCS formatter level = 0

Module information for HASMFMTM
$MIT: 20348000
+0000 MITID.... MIT MITNAME.. HASMFMTM MITVRSN.. z/OS 1.7 MITUVRSN. DRZ707 MITUSER.. MITCBV
... 01
+0025 MITENVIR. I MITLEN... 0050 MITMVRSN. 6
+0029 MITFLAG1. 28
Byte MITFLAG1 bit MIT1IBMB: IBM base module
Byte MITFLAG1 bit MIT1PTF: PTFNUM exists
+002A RSV..... 0000 MITFMID.. HJE7720 MITDATE.. 03/11/05 MITTIME.. 19.04 MITMODSZ. 00D8E0
MITENTAD. 20355858
+0048 MITXMAPA. 00000000 MITAPARN. 203558D0
Last APAR = NONE
Last PTF = NONE

Module information for HASMTABL
$MIT: 20356000
+0000 MITID.... MIT MITNAME.. HASMTABL MITVRSN.. z/OS 1.7 MITUVRSN. DRZ707 MITUSER.. MITCBV
... 01
+0025 MITENVIR. I MITLEN... 0050 MITMVRSN. 6
+0029 MITFLAG1. 28
Byte MITFLAG1 bit MIT1IBMB: IBM base module
Byte MITFLAG1 bit MIT1PTF: PTFNUM exists
+002A RSV..... 0000 MITFMID.. HJE7720 MITDATE.. 03/11/05 MITTIME.. 21.13 MITMODSZ. 04EAE0
MITENTAD. 203A4AA8
+0048 MITXMAPA. 00000000 MITAPARN. 203A4AD0
Last APAR = NONE
Last PTF = NONE
```

Ensure JES2 FMID in display matches JES2 FMID in the dump you're diagnosing.

Ensure JES2 service level in the dump you're diagnosing is within the usable range shown in the display. See “Using IPCS panels to analyze JES2 data in a dump” to determine the JES2 FMID and service level.

Note: The IPCS service level for a given JES2 release is set in the &J2_IPCS_LEVEL assembly time variable. &J2_IPCS_LEVEL is initially zero for each release and increases with changes to either the JES2 control block mappings, data structures, or IPCS formatters that are changed significantly in the service stream. See "Determining the JES2 release level" *z/OS JES2 Installation Exits*.

Using IPCS panels to analyze JES2 data in a dump

Your Time Sharing Options/Extensions (TSO/E) logon procedure should include the data sets that contain the JES2 panels, models, and find routines. If the logon procedure does not contain these data sets, you can create a TSO/E CLIST or a Restructured Extended Executor Language (REXX) exec to concatenate the needed data sets with other data sets you need for ISPF. See *z/OS MVS IPCS User's Guide* and *z/OS MVS IPCS Customization* for information about allocating these libraries.

From the JES2 Component Data Analysis menu, you can select the information you want to display from the dump. You can request dump information based on job and job output control blocks, devices, processors, subtasks, network control blocks, and MAS members. Most of the options will place you at another panel to request more information. All of the panels have help screens to aid you in data entry, and the details of those panels will not be repeated here.

Use the JES2 base display for every dump containing JES2 data, including dumps of jobs executing under JES2 control. This display includes information about the JES2 subsystem and reports numerous exception conditions.

The following example JES2 base display reports a common problem where the JES2 IPCS support doesn't match the JES2 in the dump:

```
*** JES2 Base Display ***

 Subsystem "JES2" is in address space ASID(X'0017')
 Dump for JES2 release="z/OS 1.5", Product level=35, Service level=0 (pointed
      to by SSCTSUSE); CVTPRODI=HBB7708
 Maximum extended region size for "JES2" is 1,928M (per LDAELIM)
 *** WARNING:  ASCBDSP1=80
      System set non-dispatchable and this ASCB is not exempt (per ASCBSSND
           bit)
 *** WARNING:  "JES2" abending, abended, or hot starting (per CCTHASP in $HCCT)

 *** NOTICE:  $QSUSE is in effect (per $QSONDA bit in $STATUS in $HCT)
 *** WARNING:  This member is terminating (per $SYSEXIT bit in $STATUS in $HCT)
 *** NOTICE:  Checkpoint is reserved (per $CKPTRSV bit in $STATUS in $HCT)
 *** NOTICE:  Update mode $PREBERTs exist ($PBELST¬=0 in $HCT)

  $HCCT: 00B46630
     +0000  CCTVRSN.. 0A        RSV..... 00000000  000000
     +0008  CCTOFSTB. 0729A050  CCTLMT1.. 07365DB0  CCTPVRSN. SP 5.3.0
     +0018  CCTBLNKS.
     +0038  CCTZEROS. 00000000  00000000  00000000  00000000  00000000
     +004C            00000000  00000000  00000000  00000000  00000000
     +0060            00000000  00000000  00000000  00000000  00000000
     +0074            00000000
     +0078  CCTFFS... FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF
     +008C            FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF
     +00A0            FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF  FFFFFFFF
     +00B4            FFFFFFFF
     +0078  CCTNEG1.. FFFFFFFF  CCTF1.... 00000001  CCTF2.... 00000002
     +00C0  CCTF4.... 00000004  CCTF6.... 00000006  CCTF8.... 00000008
     +00CC  CCTF12... 0000000C  CCTF16... 00000010  CCTF255.. 000000FF
```

Using the CBFORMAT command to display JES2 control blocks

You can use the **CBFORMAT** command to display the control blocks in Table 39 (using the **STRUCTURE**(*structurename*) parameter to specify the structure).

Note: In some cases, the structure name is actually the name of a DSECT within a macro (preceded by the '$' symbol). For example, $BERTIE corresponds to the BERTIE DSECT within the $BERT macro. In most cases, the structure name and the macro name are the same.

Table 39. JES2-Related Control Blocks Supported by the IPCS CBFORMAT Command

IPCS Structure Name	Description
$ACE	Automatic Command Element
$ACT	Automatic Command Table
$APT	NJE/SNA Application Table
$ARMG	ARM Support JESXCF Message
$BAT	Buffer Auxiliary Table
$BERT	Block Extension Reuse Table. Specify VIEW(X'0600') on the CBFORMAT command

Table 39. JES2-Related Control Blocks Supported by the IPCS CBFORMAT Command (continued)

IPCS Structure Name	Description
$BERTIE	Length and ID of BERT data (BERTIE DSECT in $BERT macro)
$BERTI0	Lock, chaining fields, key (BERTI0 DSECT in $BERT macro)
$BERTTAB	Table pairs to define BERTs (HASP table entries in module HASPTABS)
$BRTMAP	Maps BERTIE name to ID (BRTMAP DSECT in $BERT macro)
$BRTPREF	BERT CTENT prefix area (BRTPREF DSECT in $BERT macro)
$BRTRANS	ECSA tables reflecting $BERTTAB entries (BRTRANS DSECT in $CATBERT macro)
$BTE	Bad track group element
$BUFFER	Buffer DSECT
$CADDR	Common Storage Address Table
$CAT	Class Attribute Table
$CATBERT	Collector Attribute Table for BERTs
$CHARJOE	Job Output Element (Similar to $JOE, but forces the JOE to format as a Characteristic-JOE regardless of its type)
$CKB	Checkpoint Block (CKB DSECT in $CK macro)
$CKG	Checkpoint Generalized Parameter List (CKG DSECT in $CKGPAR macro)
$CKM	Checkpoint Inter-Member Communications Area
$CKPRECV	Checkpoint Recovery Parameter List
$CKW	Checkpoint Work Area
$CKX	Checkpoint Reconfiguration JESXCF Messages
$CMB	Console Message Buffer
$CPEBE	Cell Pool Extent Block Element
$CPINDEX	Cell Pool Index Table
$CPMASTR	Cell Pool Master Element
$CVCB	Checkpoint Version Control Block

$DAS	Direct Access Spool Data Set
	
	On the CBFORMAT command specify one of the following VIEWs:
	`- VIEW(X'0201') for a checkpoint resident $DAS`
	`- VIEW(X'0202') for a $DAS in common storage`

$DCT	Device Control Table
$DSB	Data Space Control Block
$DSCT	Data Set Control Table
$DTE	Daughter Task Element
$DWA	$DILBERT Work Area
$ERA	Error Recovery Area
$EVENT	PERFDATA Events
$EVT	ENF LISTEN Event
$EZA	EZASMI Work Area
$FMH	SNA Function Management Header
$FSACB	FSA Control Block
$FSAXB	FSA Control Block Extension

Table 39. JES2-Related Control Blocks Supported by the IPCS CBFORMAT Command (continued)

IPCS Structure Name	Description
$FSSCB	FSS Control Block
$FSSXB	FSS Control Block Extension
$GPQE	General Purpose Subtask Queue Element
$HASXB	HASP Address Space Extension Block
$HASB	HASP Address Space Block
$HCCT	HASP Common-Storage Communication Table
$HCT	HASP Communication Table
$HFAM	HASP File Allocation Map
$HFAME	HASP File Allocation Map Entry
$HFCT	HASP FSS Communication Table
$HJCT	JES2 Monitor Communication Table
$ICE	Interface Control Element
$INITST	JES2 Initialization Statistics (INITSTAT DSECT in $PERFCB macro)
$IOT	Input/Output Table
$IRE	Internal Reader Tracking Element
$IRIS	Internal Reader Init Statement
$IRWD	Internal Reader Work DSECT
$JCT	HASP Job Control Table
$JCTX	Job Control Table Extension
$JIB	JOE Information Block
$JIBX	JOE Extended Information Block
$JNT	Job Number Table
$JOA	Artificial JOE Formats either: • An artificial JOE built by JES2 from its component parts using the $DOGJOE service • A JOE residing in the in-storage copy of the checkpoint and built by JES2 IPCS support from its component parts and then formatted as a JOA
$JOE	Job Output Element
$JOX	Job Output Element Extension (Only JOE field names defined for the JOX CTENT are formatted)
$JQA	Artificial JQE Formats either: • An artificial JQE built by JES2 from its component parts using the $DOGJQE service • A JQE residing in the in-storage copy of the checkpoint and built by JES2 IPCS support from its component parts and then formatted as a JQA
$JQB	The BERT Portion of an Artificial JQE
$JQE	Job Queue Element (Only JQE field names defined for the JQE CTENT are formatted)
$JQX	Job Queue Element Extension (Only JQE field names defined for the JQX CTENT are formatted)

Table 39. JES2-Related Control Blocks Supported by the IPCS CBFORMAT Command (continued)

IPCS Structure Name	Description
$JRW	Job Receiver Work Area
$JTW	Job Transmitter Work Area
$JWEL	JOE/Writer Exclude List
$KAC	Checkpoint Application Copy
$KIT	Checkpoint Information Table
$KITL	Checkpoint Information Table (Local version that resides in JES2 private area storage, not in checkpoint)
$LMD	Limit Monitoring Data
$LMDE	Limit Monitoring Data Element
$LMT	Load Module Table
$MCT	Master Control Table
$MIT	Module Information Table
$MODMAP	Module Map for HASJES20 and HASPINIT
$MONCB	Monitor Address Space Control Table
$MSD	Monitor Sampling Data
$MSDCSD	Monitor CPU Sampling Data
$MTQH	Main Task Queue Header
$MTRB	Main Task Request Block
$MWT	Monitor Wait Table
$MWTHDR	Monitor Wait Table Header
$NAT	Nodes Attached Table Element
$NCPE	NETSRV Common POST Element
$NIT	Node Information Table
$NITP	Node Information Table Path Elements (NITP DSECT in $NIT macro)
$NJETRC	NETSRV Rolling Trace area
$NSACT	Network Subnet Anchor Table Entry
$NSCT	NETSRV Control Table
$NSST	NETSRV Socket Table
$NSWE	NETSRV Subtask Work Element
$ODPARM	Output Descriptor Parameter Block
$PAD	PROCLIB Allocation Descriptor
$PADDR	Private Storage Routine Address Table
$PADE	PROCLIB Allocation Descriptor Element
$PBEUSER	PREBERT User Stack Element
$PBLK	HAM Protected Block
$PCE	Processor Control Element
$PCL	Persistent Collection Line Element
$PCLJT	PCL Job Transmitter area
$PCLJR	PCL Job Receiver area
$PCLST	PCL SYSOUT Transmitter area

Table 39. JES2-Related Control Blocks Supported by the IPCS CBFORMAT Command (continued)

IPCS Structure Name	Description
$PCLSR	PCL SYSOUT Receiver area
$PCT	Path Manager Control Table
$PDDB	Peripheral Data Definition Block
$PERFCB	Performance Data Anchor Control Block
$PIT	Partition Information Table
$PPB	$PCE Performance Block (PPB DSECT in $PERFCB macro)
$PSCBD	$POST Performance Block (PSCBD DSECT in $PERFCB macro)
$PRE	Processor Recovery Element
$PREBERT	Prefix for BERT Processing
$PRMB	Monitor Probe Message Information Work Area
$PSO	Process SYSOUT Work Area
$PSV	Save Area
$PTPB	$PCE Type Performance Block (PTPB DSECT in $PERFCB macro)
$QSE	Shared Queue Element
$RAT	Remote Attribute Table
$RDT	Remote Destination Table
$RECY	DAS Recovery Element
$RJCB	Job Reader Card Buffer
$SAPID	SYSOUT API Data Area
$SAVEBEG	Checkpointed area of the $HCT
$SCANWA	$SCAN Facility Work Area
$SCID	Summary of Checkpoint Information
$SCK	NJE/TCP Socket
$SCQ	Shared Communications Queue Element
$SDB	Subsystem Data Set Block
$SFRB	Scheduler Facility Request Block
$SJB	Subsystem Job Block
$SJIOB	Subsystem Job I/O Buffer
$SJXB	Subsystem Job Block Extension
$SPUD	Space Utilization Description Block
$SQD	Subtask Queue Descriptor
$SRW	SYSOUT Receiver Work Area
$STAC	STATUS and CANCEL Work Element
$STW	SYSOUT Transmitter Work Area
$STWORK	General Purpose Subtask Work Area
$SXADDR	Scan Exit Routine Address Table
$TBUF	TCP/IP Buffer
$TED	Trace Enablement Descriptor
$TGB	$TGB on the CCTIOERR error queue
$TGBBLOB	$TGB in the BLOB

Table 39. JES2-Related Control Blocks Supported by the IPCS CBFORMAT Command (continued)

IPCS Structure Name	Description
$TQE	Timer Queue Element
$TRCA	Termination Recovery Control area
$TRE	JES2 TCB Recovery Element
$TRX	JES2 TCB Recovery Element Extension
$WAIT	$WAIT In-line Parameter List
$WAVE	Work Access Verification Element
$WLMD	Work Load Manager Data Bundle
$WORKJOE	Job Output Element (Similar to $JOE, but forces the JOE to format as a Work-JOE regardless of its type)
$WSA	Work Selection Work Area
$WSC	JES2 WLM Service Class Queue Anchor
$WSP	Work Selection Parameter List
$WTCB	$WAIT Performance Control Block (WTCB DSECT in $PERFCB macro)
$XECB	Extended Event Control Block
$XIT	Exit Information Table
$XMAE	JES2 XCF MAS Member Status Block (XMAQENT DSECT in $XMAS macro)
$XMAS	Cross MAS Coupling Block
$XRQ	JES2 XCF Request Block
$XRT	Exit Routine Table
JESCT	Job Entry Subsystem Communication Table
JESCTX	JESCT Pageable Extension (JESPEXT DSECT in JESCT data area)
JSAB	Job Scheduler Address Space Control Block
SSCT	Subsystem Communications Vector Table
SSPJ	Persistent JCL Interface SSOB Extension
SSVT	Subsystem Vector Table

Using IPCS in batch or line mode

z/OS MVS IPCS User's Guide contains an example of how to invoke IPCS in a batch job or in line mode using subcommands. To access the JES2 information in the dump, you must enter the following subcommand:

```
VERBEXIT JES2VX 'jesname'
```

This subcommand formats and prints the following information from the dump:

- HCCT
- $HCT
- Current $PCE
- Save areas ($PSVs) related to the current $PCE
- Routine name associated with each $PSV
- JES2 module and offset of the caller, from which each routine was called
- $DCT related to the current $PCE
- $JQE related to the $DCT

- $JOEs related to the $PCE.

The optional parameter, *JESNAME*, allows the installation to specify the name of a particular JES2 address space to analyze. If *JESNAME* is not coded, the formatter analyzes the address space with the name JES2.

Assuring JES2 can be serviced using IPCS

CAUTION: IBM does not recommend or support modifications to JES2 source code. If you assume the risk of modifying JES2, then also assure your modifications do not impact JES2 serviceability using IPCS. Otherwise, IBM service LEVEL2 support might not be able to read JES2 dumps taken for problems unrelated to the modifications.

Avoid expanding JES2 control blocks. Use alternatives such as:

1. Use fields dedicated for installation use that appear in many major control blocks. Place your data, or a pointer to your data, in these fields. However, beware of setting storage address in checkpointed or SPOOL resident control blocks.
2. Use $JCTX services rather than modifying $JCT.
3. Use table pairs and dynamic tables. For example, use dynamic $BERTTABs with CBOFF=* instead of modifying $JQE.

This is a partial list. Evaluate your specific situation and take the appropriate action.

Chapter 6. Using symptom records for diagnosis

JES2 creates and writes symptom records to the logrec data set. These records are written for:

- JES2-detected problems
- Documenting non-error related JES2 spool space recovery actions

Controlling operator notification of JES2 symptom records

If you want JES2 to notify your installation operator that an error was detected and a symptom record created, specify the SYMREC=YES parameter on the DEBUG initialization statement or $T DEBUG command. Specifying SYMREC=YES, causes JES2 to issue the $HASP805 message when symptom records are created for JES2 detected problems. The message includes the issuing module and a description of the error.

Some JES2 symptom records report non-error related recovery actions such as recovering lost spool space. The $HASP805 message is not issued for non-error related recovery actions. If you create your own symptom records using the $SYMREC macro and want to suppress the $HASP805 message, specify DEBUGMSG=NO on your $SYMTAB macro.

Where to find JES2 symptom records

If you are experiencing JES2 problems or you have created a new JES2 exit, look for JES2 symptom records. Symptom records can be found in:

- The logrec data set
- Dumps

Using IPCS to analyze JES2 symptom records

Use the LOGDATA option of the IPCS Component Analysis panel of the IPCS dialog or the VERBEXIT LOGDATA line mode command to display all of the symptom records in the LOGREC buffer of a dump. To locate the JES2 symptom records, search the output for the component identifier of SC1BH. Figure 48 on page 106 shows a sample of a JES2 symptom record.

```
TYPE:  SYMPTOM RECORD       REPORT:  SOFTWARE EDIT REPORT           DAY YEAR
                                                    REPORT DATE: 260  90
SCP:   VS 2 REL 3                                    ERROR DATE: 260  90
                            MODEL:  3090                            HH MM SS.TH
                            SERIAL: 140471                  TIME: 06:41:03.83
SEARCH ARGUMENT ABSTRACT:
    PIDS/SC1BH CSECT/SSISFS RIDS/HASCSJFS

SYSTEM ENVIRONMENT:
    CPU MODEL:  3090               DATE:  260  90
    CPU SERIAL: 140471             TIME:  06:41:03.83
    SYSTEM:     SYSA1              BCP:   MVS
    RELEASE LEVEL OF SERVICE ROUTINE:      HBB4420
    SYSTEM DATA AT ARCHITECTURE LEVEL:     10
    COMPONENT DATA AT ARCHITECTURE LEVEL:  10
    SYSTEM DATA:  00000000 00000000    |........|
COMPONENT INFORMATION:
    COMPONENT ID:             SC1BH
    COMPONENT RELEASE LEVEL:   420
    SERVICE RELEASE LEVEL:
    DESCRIPTION OF FUNCTION:  SJFREQ SWBTU_MERGE FAILURE
    PROBLEM ID:               SWBTSYT
    SUBSYSTEM ID:             JES2
```

Figure 48. Sample JES2 Symptom Record

```
PRIMARY SYMPTOM STRING:
    PIDS/SC1BH    CSECT/SSISFS   RIDS/HASCSJFS

    SYMPTOM            SYMPTOM DATA     EXPLANATION
    ---------------    -------------    -----------
    PIDS/SC1BH         SC1BH            COMPONENT IDENTIFIER
    CSECT/SSISFS       SSISFS           UNRECOGNIZED KEYWORD
    RIDS/HASCSJFS      HASCSJFS         ROUTINE IDENTIFIER

SECONDARY SYMPTOM STRING:
    PRCS/4 PRCS/1905

    SYMPTOM            SYMPTOM DATA     EXPLANATION
    ---------------    -------------    -----------
    PRCS/4             4                RETURN CODE
    PRCS/1905          1905             RETURN CODE

FREE FORMAT COMPONENT INFORMATION:
    KEY = 0105     LENGTH = 000072 (0048)
    +000    E2D1E2D4    01000048    00000000    000001F9    |SJSM.........9|
    +010    02FF6148    004E0000    00000000    000F7794    |../..+.........M|
    +020    00020000    00000000    00000000    00000000    |................|
    +030    00000000    00000000    00000000    00000000    |................|
    +040    00000000    00000000                            |........        |

HEX DUMP OF RECORD:
  HEADER
    +000    4C831800    00000000    0090260F    07474854    |<C.............|
    +010    FF140471    30900000                            |........        |

  SYMPTOM RECORD
    +000    E2D9F3F0    F9F0F1F4    F0F4F7F1    FFFFCA5B    |SR3090140471...$|
    +010    A2B7BE3B    93226B04    40404040    40404040    |S...L.,.        |
    +020    4040E2E8    E2C1F140    4040F5F7    F5F2C8C2    |  SYSA1   5752HB|
    +030    C2F4F4F2    F0400080    00000000    00000000    |B4420 ..........|
    +040    F1F00030    00640070    002B00D4    001000FF    |10.........M....|
    +050    004C010F    00000000    00000000    00000000    |.<..............|
    +060    00000000    00000000    00000000    00000000    |................|
    +070    E2D9F2F1    F1F0E2C3    F1C2C840    40404000    |SR2110SC1BH    .|
    +080    40F4F2F0    40404040    40404040    00000000    | 420        ....|
    +090    00000000    00000000    00000000    E2D1C6D9    |............SJFR|
    +0A0    C5D840E2    E6C2E3E4    6DD4C5D9    C7C540C6    |EQ SWBTU_MERGE F|
    +0B0    C1C9D3E4    D9C54040    40404040    00000000    |AILURE      ....|
    +0C0    00000000    E2E6C2E3    E2E8E340    D1C5E2F2    |....SWBTSYT JES2|
    +0D0    00000000    D7C9C4E2    61E2C3F1    C2C84040    |....PIDS/SC1BH  |
    +0E0    4040C3E2    C5C3E361    E2E2C9E2    C6E24040    |  CSECT/SSISFS  |
    +0F0    40D9C9C4    E261C8C1    E2C3E2D1    C6E240D7    | RIDS/HASCSJFS P|
    +100    D9C3E261    F440D7D9    C3E261F5    F0F54001    |RCS/4 PRCS/505 .|
    +110    050048E2    D1E2D401    00004800    00000000    |...SJSM.........|
    +120    0001F902    FF614800    4E000000    00000000    |..9../..+.......|
    +130    0F779400    02000000    00000000    00000000    |..M.............|
    +140    00000000    00000000    00000000    00000000    |................|
    +150    00000000    00000000    000000                  |...........     |
```

Figure 49. Sample JES2 Symptom Record, cont.

The record contains:

SEARCH ARGUMENT ABSTRACT
The search argument to use when searching problem reporting data bases for a fix for this problem.

SYSTEM ENVIRONMENT
Information about the hardware and software on which the problem occurred.

COMPONENT INFORMATION
Information that identifies the component that detected the error and function the component was performing when the error occurred.

PRIMARY SYMPTOM STRING
The search argument abstract with an explanation of each of the symptoms in the abstract.

SECONDARY SYMPTOM STRING
Additional information about the error. This section of the record contains return and reason codes.

FREE FORMAT COMPONENT INFORMATION
A variable area that contains other information to help isolate the cause of the problem. Examples of the type of information in this area are:

- Parameter lists
- Input to the function
- Information in work areas.

HEX DUMP OF RECORD
The hexadecimal representation of the record in dump output format. This part of the record contains the subsections:

- HEADER
- SYMPTOM RECORD

Use the search argument abstract from the record to search problem reporting data bases for an existing fix for the problem. If no fix exists, report the problem to the IBM Support Center.

Reporting information from the symptom record to IBM

For all problems report the following information from the record:

- System Environment
  - CPU Model
  - BCP
  - Release Level of Service Routine
  - System Data at Architecture Level
  - Component Data at Architecture Level
- Component Information
  - Component ID
  - Component Release Level
  - Service Release Level (if any)
  - Description of Function
  - Problem ID
  - Subsystem ID
- Primary Symptom String

Use Table 40 on page 109 to determine the other information to report to the IBM Support Center.

Table 40. Information to Collect from Symptom Record

Description of Function (appears under Component Information)	**Information to Collect**
NCC RECORD REJECTED	• Secondary Symptom String • Free Format Component Information – The Key value – The free format data, which contains NCC the record that JES2 rejected.
SJFREQ SWBTU_MERGE FAILURE	• Secondary Symptom String – The decimal return codes. • Free Format Component Information – The Key value – The free format data, which contains the parameter list supplied to the SJFREQ service.
SWBTUREQ SPLIT REQUEST FAILURE	• Secondary Symptom String • Free Format Component Information – The Key value – The free format data, which contains the parameter list supplied to the SWBTUREQ service.
SWB MODIFY SUBTASK FAILURE	• Secondary Symptom String • Free Format Component Information – The Key value – The free format that which contains: - The SJF services request block - The subtask work area.
COPY COUNT OR COPY GROUP COUNT INVALID IN XIT	• Secondary Symptom String • Free Format Component Information – The Key value – The free format data, which contains the copy count and copy group count.
$CPOOL ERROR GETTING SMF BUFFERS	• Secondary Symptom String • Free Format Component Information – The Key value – The free format data which contains: - The $CPOOL parameter list. - The extent information parameter list.
$CPOOL ERROR IN QUESMFB	• Secondary Symptom String • Free Format Component Information – The Key value – The free format data which contains the cell information parameter list.

Table 40. Information to Collect from Symptom Record (continued)

Description of Function (appears under Component Information)	Information to Collect
$CPOOL ERROR IN FRESMFB	• Secondary Symptom String • Free Format Component Information – The Key value – The free format data which contains the cell information parameter list.
ICE FREE FAILED	• Secondary Symptom String • Free Format Component Information – The Key value – The free format data which contains the Interface Control Element (ICE).
ICE HAS BEEN FROZEN	• Secondary Symptom String • Free Format Component Information – The Key value – The free format data which contains the Interface Control Element ($ICE).

JES2 symptom records

Table 41 provides the symptom records that JES2 writes to logrec data set.

Table 41. JES2 Symptom Records

Description of Function (appears under Component Information)	Meaning
ALLOCATION ATTEMPT ($STRAK)	A serious error was prevented. During spool allocation, not in the JES2 main task, JES2 determined that a track group marked as available in the track group map is potentially allocated by another job. **System Action:** JES2 does not use the track group and attempts to allocate another. **System Programmer Action:** Contact your IBM Support Center.
ALLOCATION ATTEMPT ($TRACK)	A serious error was prevented. During spool allocation, in the JES2 main task, JES2 determined that a track group marked as available in the track group map was still allocated by another job. **System Action:** JES2 does not use the track group and attempts to allocate another. **System Programmer Action:** Contact your IBM Support Center.
COPY OR COPY GROUP COUNT INVALID	The copy or copy group count returned from an installation Exit 15 call was too high. **System Action:** JES2 sets the count(s) in error to one. **System Programmer Action:** Correct your installation Exit 15.

Table 41. JES2 Symptom Records (continued)

Description of Function (appears under Component Information)	Meaning
$CPOOL ERROR GETTING SMF BUFFER	This symptom record accompanied by $ERROR $SG2. $SG2 indicates that while processing a request to get an SMF buffer, $CPOOL services failed to return a cell. **System Action:** If recovery is attempted from this error, SMF processing will be turned off. **System Programmer Action:** Contact your IBM Support Center.
$CPOOL ERROR IN $QUESMFB	This symptom record is accompanied by $DISTERR at label QSMFDIS in HASPNUC. **System Action:** JES2 does not write the requested SMF record and the buffer for the SMF record is abandoned. **System Programmer Action:** Contact your IBM Support Center.
DISTERR	A JES2 disastrous error occurred. This symptom record is issued whether or not a dump is provided. If a dump is provided, MVS recovery termination issues a second symptom record. This symptom record is accompanied by $ERROR $DIS. $DIS indicates that JES2 encountered a disastrous error. See the preceding $HASP096 message for more diagnostic information. **System Action:** JES2 recovers and continues. Other JES2 action is specific to the error identified in the accompanying $HASP095 message. There may also be additional messages describing JES2 actions. **System Programmer Action:** Contact your IBM Support Center.
DISTERR W/ACTIVE LINKAGE STACK	A JES2 disastrous error occurred. See the preceding $HASP096 message for more diagnostic information. **System Action:** JES2 recovers and continues. Other JES2 action is specific to the error identified in the accompanying $HASP095 message. There may also be additional messages describing JES2 actions. **System Programmer Action:** Contact your IBM Support Center.
ICE FREE FAILED	A serious error was prevented. An attempt was made to free a $ICE that was already freed. **System Action:** JES2 continues. **System Programmer Action:** Contact your IBM Support Center.
ICE HAS BEEN FROZEN	A serious error was prevented. An attempt was made to process an $ICE that was not in a valid state. **System Action:** JES2 does not attempt to use this $ICE. **System Programmer Action:** Contact your IBM Support Center.
JQASUMSK DOES NOT REFLECT VOLSER	JES2 found a bit missing in the SPOOL used mask (JQASUMSK). **System Action:** JES2 corrects the missing volumes it discovers in the SPOOL used mask. **System Programmer Action:** Contact your IBM Support Center.

Table 41. JES2 Symptom Records (continued)

Description of Function (appears under Component Information)	Meaning
NCC RECORD REJECTED	The network path manager (NPM) received a non-valid network connection control (NCC) record. **System Action:** JES2 ignores the non-valid NCC record. **System Programmer Action:** Contact your IBM Support Center.
NETWORK HEADER VALIDATION ERROR	JES2 detected a non-valid network job header or network data set header. **System Action:** JES2 causes a negative close on the network device. **System Programmer Action:** Contact your IBM Support Center.
NODAL MESSAGE RECORD INVALID	JES2 received a nodal message record (NMR) that is not valid. **System Action:** JES2 ignores the non-valid NMR. **System Programmer Action:** Contact your IBM Support Center.
SJFREQ SWBTU_MERGE FAILURE	An attempt to merge scheduler work blocks (SWBs) for an application (such as SDSF) failed. **System Action:** JES2 aborts the merge request. A return code of 8 is returned in register 15 to the issuer of the merge request. **System Programmer Action:** Contact your IBM Support Center.
SPOOL INVALID PURGE ATTEMPT	A serious error was prevented. During spool unallocation, a track group purportedly belonging to a job or output group being purged was found to be allocated to a different job or to be not allocated at all. **System Action:** JES2 ignores the track group. • If the track group does in fact belongs to a different job, it will be freed on behalf of that job at a later time. If the job purported to be the owner later purges without freeing the track group, then the track group will be recovered by JES2 spool examination and recovery processing within seven days from the time the purported owning job purges. • If JES2 finds the track group is already marked “not allocated” in the master track group map, then no further action is needed. **System Programmer Action:** Contact your IBM Support Center.
SPOOL TG RECOVERY (BLOB/PURGE)	JES2 found an unclaimed track group allocated to a cache (called the BLOB) for a member being warm started. This condition came about because the member did not end as a result of a $PJES2 or $PJES2,TERM command and could be precipitated by: • An MVS system failure, system reset, or IPL while JES2 was active • An IPL after a $PJES2,ABEND or $PJES2,ABEND,FORCE command **System Action:** The track group is made available for allocation. **System Programmer Action:** None. This is an informational record only, unless accompanied by other symptom records or disastrous errors.
SPOOL TRACKGROUP RECOVERY	JES2 found a track group that is not owned or listed in the master track group map. **System Action:** JES2 makes the track group available for allocation. **System Programmer Action:** None. This is an informational record only, unless accompanied by other SYMRECs or disastrous errors.

Table 41. JES2 Symptom Records (continued)

Description of Function (appears under Component Information)	Meaning
SWB MODIFY SUBTASK ERROR	An attempt to modify a scheduler work block (SWB) buffer for an application (such as SDSF) failed. **System Action:** JES2 aborts the modify request. The target scheduler block will not be modified. A return code is returned in the Subsystem Option Block (SSOB), field SSOBRETN, and a reason code is returned in the Subsystem Option Block Extension (IAZSSSF), field SSSFREAS. **System Programmer Action:** Contact
SWBTUREQ SPLIT REQUEST FAILURE	An attempt to modify a scheduler work block (SWB) for an application (such as SDSF) failed. **System Action:** JES2 aborts the modify request. The target scheduler block will not be modified. A return code is returned in the Subsystem Option Block (SSOB), field SSOBRETN, and a reason code is returned in the Subsystem Option Block Extension (IAZSSSF), field SSSFREAS. **System Programmer Action:** Contact your IBM Support Center.
USER ADDRESS SPACE DISTERR	JES2 has encountered an I/O error reading or writing a control block to SPOOL from the user environment. The error could have been returned from IOS or it could be a logical error (such as the control block eyecatcher or job key does not match). **System Action:** The error is returned to the caller of $CBIO. JES2's response depends on the reason for the I/O. **System Programmer Action:** If the error is the result of a problem with the SPOOL device, correct the problem with the device and retry the operation. If this is a logical problem with the data being processed, contact your IBM support center.
$WTO PARAMETER LIST ERROR	JES2 detected an incorrect $WTO parameter list. **System Action:** JES2 continues with the incorrect $WTO parameter list. JES2 corrects the $WTO parameter list in some simple cases. **System Programmer Action:** See "Diagnostic procedures for $WTO PARAMETER LIST ERROR symptom records" for more information on diagnosing the problem.
ZAPJOB Request Processed	A request for a ZAPJOB initialization statement or command was processed for the $JQE that appears in the record. **System Action:** JES2 continues without the job represented by the JQE. **System Programmer Action:** None.

Diagnostic procedures for $WTO PARAMETER LIST ERROR symptom records

$WTO PARAMETER LIST ERROR symptom records can be caused by problems in JES2 or in JES2 installation exits. Possible causes include the incorrect use of $WTO, $WTOR and $BLDMSG macro instructions and related data, or the incorrect manipulation of a $CMB data area by a JES2 installation exit 10 routine. Typically, multiple $WTO PARAMETER LIST ERROR symptom records are created for a given type of problem. Format log data from the logrec data set using EREP

and search for "$WTO PARAMETER LIST ERROR". See the *EREP User's Guide* for more information on formatting the log data set.

JES2 validates the $WTO parameter list at key points in JES2 processing and creates a $WTO PARAMETER LIST ERROR symptom record if it discovers an error. Additionally, JES2 corrects some errors. However, JES2 cannot detect or correct every possible error. Some errors cause problems in later processing, including causing JES2 to place CMBs on a queue where they could remain indefinitely. In some cases, this may result in a $HASP050 resource shortage message for CMBs. See "Resource shortages" on page 12 for more information on handling resource shortages.

Finding the data

You will need the secondary symptom string and the free format component information to diagnose the symptom records. The secondary symptom string includes one of the following keywords:

BEFEXIT
Indicates error found before entering EXIT 10.

AFTEXIT
Indicates error found after EXIT 10 returned.

SPOOLIN
Indicates error found in a parameter list passed from another member of the MAS.

The format component information contains data mapped starting at symbol CMBWTOPL in the $CMB data area. Subtract the offset of the CMBWTOPL symbol to the determine the offset of $CMB fields in the free format component information.

Determining why JES2 created a symptom record

Use the secondary symptom string keywords to help determine where the error first occurred. Multiple symptom records can be issued for a $WTO parameter list error as the $CMB data area that contains it moves through JES2 processing on one member of a MAS, on another member of the MAS, and on other nodes.

- If the secondary symptom string indicates SPOOLIN, the parameter list came from another member of your installation's MAS. Check for parameter list error symptom records on the other members of the MAS and continue your diagnosis with those records.
- If the secondary symptom string indicates BEFEXIT and is not preceded by a corresponding record with SPOOLIN, the parameter list originated on this member or came from a NJE line connected to this member. Use the following guidelines when checking fields CMBFMNOD and CMBFMQUL in the parameter list (remember, the parameter list starts at symbol CMBWTOPL) to determine the origin node and member.
  - If the parameter list came from another node in a Nodal Message Record (NMR), use JES2 trace ID 4 or 5, as appropriate, to capture the NMR exactly as you received it, then contact the people at the sending node to continue diagnosing the problem.
  - If the parameter list originated on this node or if CMBFMNOD and CMBFMQUL are both zero, look at other parameter list fields including the message or command text to help determine who issued the message or command.

- If the secondary symptom string indicates AFTEXIT and is not preceded by a corresponding record with BEFEXIT, then installation exit 10 has corrupted the parameter list.

Correcting the error

When you determine the cause of the error, take one of the following steps, as appropriate:

- Contact your IBM Support Center.
- Correct your installation exit.
- If the $WTO parameter list came from another node, use JES2 trace ID 4 or 5 to capture the NMR containing the incorrect information. Contact the people at the site that sent the incorrect NMR for assistance.

Chapter 7. JES2 Health Monitor

The JES2 **health monitor** can assist in determining why JES2 is not responding to requests. It does this through the use of highlighted messages and operator commands. JES2 issues "monitor" messages when conditions exist that can seriously impact JES2 performance. These conditions could indicate problems with the operating system or JES2. They can also indicate transient conditions (such as a multi-address space dump) that is impacting JES2. Conditions that trigger messages are time related and are not otherwise reported by JES2. Other conditions that are the result of operator actions or shortages do not generate messages but are instead displayed in response to monitor commands.

When you suspect a JES2 problem, use the $JD STATUS command to determine if the monitor has detected a possible cause. This command displays information on any condition the monitor has detected that could impact JES2 including conditions for which a highlighted message has been issued. The monitor is single system in nature. Because some problems can have a multi-system impact, the command might need to be issued (or routed) to multiple member of the MAS.

Monitor processing

The JES2 health monitor runs in a separate address space from JES2. There is a corresponding monitor address space for every JES2 address space. The address space name is *jes2*MON where *jes2* is the name of the subsystem that is being monitored. Within the monitor address space, a number of tasks perform the functions needed to determine how the JES2 address space is performing. Use the $JD MONITOR command to display the actual tasks and their current status. This command displays status information on each monitor task and documents the service level of each module that makes up the monitor.

Because most cases of JES2 not responding to requests are caused by the JES2 main task not operating correctly, that is the monitor's primary area of focus. There are some cases where subtasks can cause JES2 to not function properly. Most notable are cases when requests to the security product are not completing in a timely manner. The monitor **does not** monitor requests made to JES2 subtasks.

The monitor uses cross–memory services to examine data in the JES2 address space. The monitor collects data by sampling values at regular intervals and by extracting data needed to build command responses. This process is designed to have a minimal impact on JES2 operations. However, because the processing is not serialized, it is possible that minor discrepancies could arise in the data it collects and displays. Though rare, this can result in messages that display conditions that could not or did not occur. For example, the monitor collects the current instruction address, PCE address, and exit number in three separate operations. It is possible that the current PCE was in the process of transition and the address does not represent code being run by the exit, or the exit is not one that is used by the PCE. Although the condition that caused the message is valid, and each item of information was valid when collected, the combination of the specific information displayed is not correct. A subsequent $JD STATUS command should provide correct information.

Setting up the monitor

The monitor does not require any special setup. It is started automatically when JES2 is started and shuts down when JES2 terminates normally. There are no initialization statements, PARMLIB or PROCLIB statements that need to be updated. However, you may want to define the address space to your security product. The monitor does not access any RACF–protected resources; however, JES2 builds an ACEE based on attributes in the STARTED class. You should also ensure that it is classified to WLM consistent with your JES2 classification.

You cannot turn the monitor off. If the address space terminates because of error or operator command, the JES2 address space will restart the monitor. The overhead associated with the address space is minimal (less than 1% of a CPU). Repeated failures and restarts of the monitor address space will not impact normal system operations.

Monitor Alert Messages

The JES2 health monitor examines the activity of the JES2 main task looking for conditions that indicate a problem. The following table show which conditions are monitored.

Table 42. JES2 Health Monitor Alert Messages

Condition	Monitor Message Number	Repeat interval	Condition Normal Message	Exclusive Alert
Main task in an MVS wait (other than the normal wait)	$HASP9201	30 Seconds	$HASP9301	Yes
Main task in a loop	$HASP9202	30 Seconds	$HASP9301	Yes
PCE dispatch for a long period	$HASP9203	30 Seconds	$HASP9301	Yes
Main task never entering normal MVS wait	$HASP9204	30 Seconds	$HASP9301	Yes
Main task waiting for the local lock	$HASP9208	30 Seconds	$HASP9301	Yes
Main task is non-dispatchable	$HASP9209	30 Seconds	$HASP9301	Yes
Main task waiting because of a page fault	$HASP9210	30 Seconds	$HASP9301	Yes
Main task not running (at normal wait for extended period of time)	$HASP9211	30 Seconds	$HASP9301	Yes
MVS not dispatching JES2 main task	$HASP9212	30 Seconds	$HASP9301	Yes
Checkpoint lock held for a long period	$HASP9207	30 Seconds	$HASP9302	No
PCE in a wait for a BERT lock	$HASP9205	120 Seconds	None	No
PCE in a wait for a JOB lock	$HASP9206	120 Seconds	None	No
Long JES2 command processing	$HASP9213	120 Seconds	None	No

These conditions can occur normally when JES2 is running. However, if the condition persists for a long enough period of time, that might indicate a problem. When JES2 detects a condition that needs monitoring, it is considered an incident. Incidents are divided into three categories depending on their duration.

- **Normal processing:** below a low time threshold (less than 5 seconds), incidents are considered part of normal processing.

- **Tracking:** If the low–time interval is exceeded, the monitor tracks it. Tracking involves creating a data element that describes the incident. These elements can be displayed on a $JD JES command. One condition can trigger multiple incidents to be tracked. For example, a PCE that is looping will trigger a main task loop, a long PCE dispatch, main task not entering normal MVS wait, and checkpoint lock held. The monitor tracks each of the incidents as they cross the low–time threshold.
- **Alerts:** JES2 issues alerts when an incident that is being tracked crosses a second threshold. This threshold is not a time threshold but rather a sampling threshold. Before an incident becomes an alert, a specific number of samples must be collected indicating the condition still exists. On a normally running system, this range is about 8-20 seconds depending on the condition being monitored. This prevents situations where nothing is running in the operating system (including the sampling code) from appearing as a problem specific to JES2. When an incident is considered an alert, JES2 issues a highlighted message to the console. If the alert persists, the highlighted message is re-issued at the interval specified in Table 42 on page 118. When JES2 issues an alert, the monitor stops tracking other incidents if it consider the alert to be exclusive. This is intended to focus attention on the primary problem (for example, a JES2 loop) rather than the secondary effect (main task is not waiting). Alerts for the checkpoint lock being held are not exclusive because of the multi-system impact that the lock being held can have. This alert is designed to focus attention on the correct system when multiple members of a JES2 MAS are not responding.

When the condition that caused the alert no longer exists, JES2 deletes the highlighted message. For condition relating to the main task, JES2 issues a $HASP9301 message when all conditions being tracked or alerted have been cleared. Similarly, if the checkpoint lock was held for a long period, JES2 issues message $HASP9302 when the lock is released.

Monitor notices

In addition to alerts and tracking messages, the monitor can display information on conditions that are not time related in nature. These are called notices and they are only displayed in response to $JD STATUS and $JD JES commands. Notices are conditions that arise from operator commands, resource shortages, or system errors. Often, the information that is displayed in notices is available in message response to other commands and summarized in the notice. Following is a list of some example notices.

- $HASP9151 JES2 ADDRESS SPACE NOT ACTIVE– compare with $HASP095 JES2 CATASTROPHIC ERROR ǀ ABEND...
- $HASP9154 CKPT RECONFIGURATION IN PROGRESS – compare with $HASP285 JES2 RECONFIGURATION *status*
- $HASP9158 JES2 PROCESSING STOPPED, $S NEEDED – compared with $HASP623 MEMBER DRAINING
- $HASP9161 NOT ALL SPOOL VOLUMES ARE AVAILABLE – compare with $HASP424 MEMBER *volser* IS NOT MOUNTED

More information on the causes and responses to notices can be found in *z/OS JES2 Messages*.

Resource monitoring

As part of monitor sampling, the monitor collects resource utilization information for the major JES2 resources. The monitor also collects the low, high, average, and current utilization of each resource and reset it at the start of every hour. Use the $JD DETAILS command to display this information. Though this command displays information similar to some JES2 display command, it provides a single command that can display the information even when JES2 is not responding to normal commands.

Similar to resource utilization, the monitor maintains statistics on JES2 main task CPU samples. The samples are broken down into a number of major categories. The sampling statistics for the current hour can also be displayed in response to the $JD DETAILS command.

Monitor commands

The monitor supports a number of commands to display information it is tracking. All monitor commands start with the JES2 command prefix followed by a 'J'. These commands are routed directly to the monitor address space and are not seen by the JES2 address space. As a result, there are some limitations on monitor commands; following is a list of those limitations.

- JES2 Exit 5 is not called for monitor commands.
- JES2 Exit 10 is not called for monitor command responses.
- Monitor commands can only originate from MVS command sources (consoles, SVC 34, and so on).
- Monitor command are not accepted from JES2 command sources such as internal readers, initialization data sets, $M commands, or RJE and NJE sources.
- Console message display and limit definitions specified on the CONDEF initialization statement through CMDNUM=, RDIRAREA=, DISPLEN=, DISPMAX=, REDIRECT= parameters do not apply to monitor commands.
- Monitor commands can be the object of a JES2 automatic command.

Table 43 lists all the monitor commands and their intended use.

Table 43. JES2 Health Monitor Commands and Intended Use

Command	Description
$JD DETAILS	Displays current hour statistics for resource utilization and CPU sample statistics. The MVS wait table is also displayed.
$JD HISTORY	Displays up to 72 hours worth of resource utilization and CPU sample statistics. The display from this command can be very large (up to 73 lines per resource monitored).
$JD JES	Displays current JES2 information including alerts, trackings, and notices.
$JD MONITOR	Displays status information for each of the monitor subtasks and service information for each module that makes up the monitor.
$JD STATUS	This is the primary diagnostic command for the monitor. It displays the current status of the JES2 being monitored. This display includes all alerts and notices that are current for the JES2 being monitored.

Table 43. JES2 Health Monitor Commands and Intended Use (continued)

Command	Description
$J STOP	This command shuts down the monitor address space. If the JES2 address space is active, the monitor address space is restarted within a few minutes. This command clears any history the monitor has been maintaining.

Chapter 8. Reporting the problem

Before calling the IBM Support Center, it could be advantageous to do the following:

- Try other formats of any commands you have entered or other commands to obtain more information about the failure.
- Look up **all** related messages and take the designated actions. A message you overlook could contain the key that helps you resolve your problem.
- Determine whether the failure only occurs for one job or output. Try changing the class of the job or output. If the problem goes away there could be an error in the class definition.
- If there are equivalent commands in other software, try entering them. Do those commands agree with the JES2 results?
- Change or remove operands from commands or initialization statements. This change or removal could narrow the problem down to a certain area.
- Examine SYSLOG to determine the sequence of events for the error. If the problem is job-related, examine the events for the entire life of the job.
- Examine the COUNT parameter for the RECVOPTS. With the exception of subtasks and disastrous (DISTERR) errors, too high a value for the count parameter could prevent the possibility of collecting a dump when an error occurs and force you to recreate the problem.

  Setting the count parameter too low for subtasks and disastrous errors causes JES2 to suppress automatic dumps that may help you find the cause of the problem.
- Disable the exit, if there is an exit in the failing processing area. If the exit is causing the problem, the failure should stop when you disable the exit.
- Make sure you understand all the considerations for an exit, if the problem is exit-related. Sometimes a slight misunderstanding may cause an unexpected error.
- Use SLIP traps to help collect more information if the situation warrants it. *z/OS JES2 Commands* explains in detail how and when to use SLIP traps.
- Search the problem data bases to see if a fix already exists for your problem. If the search argument you use fails to produce a match, try other variations of the arguments and see if you obtain a match.

When you call the IBM Support Center, follow these steps to help reduce the time you spend with service and minimize your outage:

- Be specific when describing the problem. Describe the details of the problem.
- List the sequence of events leading to the problem.
- Collect enough documentation to help locate the cause of the problem, if this is the first occurrence of the problem. Minimally this is a JES2 dump, SYSLOG, and logrec data set error records. Any additional information, such as traces, are also helpful. If you are sending a tape containing a dump, make sure to include a printed listing of the job used to create the tape.
- You may need a current assembler listing for JES2 and a copy of all exits, and modifications your installation has made to JES2.
- Collect any JCL pertinent to the problem.

- Have all the documentation within reach. You may be asked to look at various parts of a dump, to help locate the problem.
- Use sequence numbers instead of offsets when explaining where in the code the problem resides. Differences in maintenance levels make offsets impractical for relating the error to the source code.
- Identify your maintenance on all problems. This is especially important for JES2 and PSF-related problems. For security problems related to Resource Access Control Facility (RACF), identify the RACF level.

Appendix. Accessibility

Accessible publications for this product are offered through the z/OS Information Center, which is available at www.ibm.com/systems/z/os/zos/bkserv/.

If you experience difficulty with the accessibility of any z/OS information, please send a detailed message to mhvrcfs@us.ibm.com or to the following mailing address:

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size.

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing the z/OS Information Center using a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually

exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol giving information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should refer to separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

- ? means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- ! means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicates that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1!

(KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

- * means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

  Note:

  1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
  2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
  3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.

- + means a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times; that is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Notices

This information was developed for products and services offered in the U.S.A. or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of 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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

COPYRIGHT LICENSE:

This information might contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted

for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (http://www.ibm.com/software/support/systemsz/lifecycle/)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming Interface Information

This publication documents information that is NOT intended to be used as programming Interfaces of JES2.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml (http://www.ibm.com/legal/copytrade.shtml).

Index

Special characters

A

B

C

D

E

H

I

J

K

L

M

N

O

P

R

S

T

U

W

Product Number: 5650-ZOS

Printed in USA

GA32-0993-00