Title: JES2 Performance Data Command ($DPERFDATA)

JES2 for OS/390 and z/OS Performance Data - Updated January 20, 2004

JES2 provides internal performance statistics which can be displayed with the **$D PERFDATA** operator command.

Disclaimer: This is intended to be used under direction of personnel knowledgeable in interpreting JES2 PERFDATA displays. Development may add/change/delete any part thereof as required to improve its usefulness as necessary, so there is no intent to document the externals in the JES2 product publications at this time.

Recent Changes:

APAR OW55693: The SUBTSTAT option was added to display subtask statistics.

z/OS Release 2: The CKPTSTAT option was added to display a summary of checkpoint activity.

OS/390 Release 10: The EVENT option was added to display errors and long-running PCEs.

Overview of the $D PERFDATA displays:

Checkpoint Statistics - $D PERFDATA(CKPTSTAT)

This displays a summary of the checkpoint statistics that are available through the $TRACE ID=17 data. (added in z/OS Rel. 2)

CPU Statistics - $D PERFDATA(CPUSTAT)

This displays a sumarized view of the PCESTAT statistics. (added in OS/390 Rel. 7)

- CPU time and wall clock run time for each PCE
- CPU time and wall clock run time based on PCE type
- I/O counts for each PCE and based on PCE type
- $QSUSE time and checkpoint counts

Event Statistics - $D PERFDATA(EVENT)

This displays significant events such as $DISTERR errors and ABENDs ($ERRORS) and long-running PCEs. (added in OS/390 Rel. 10)

Initialization Statistics - $D PERFDATA(INITSTAT)

This shows the CPU and wall clock time for each Initialization Routine module, WARM start processing, and MVS initialization (from JES2 start until HASPNUC is first entered)

PCE Statistics - $D PERFDATA(PCESTAT)

This displays, for each generic PCE (processor control element) type, the activity, execution time and delays encountered.

- Elapsed and CPU time for each PCE type
- I/O counts for each PCE and based on PCE type
- $WAIT time and count based on $WAIT macro and PCE type.
- $POST reasons, counts and wait times for each $WAIT macro.
- $QSUSE time and checkpoint counts (added in OS/390 Rel. 7)

QSUSE utilization and delay - $D PERFDATA(QSUSE)

This displays the usage and response time for $QSUSE service which is used by many JES2 processes to acquire the checkpoint.

- Module and sequence number of every $QSUSE macro that $WAITed
- Count of times each $QSUSE was entered and accumulated wait time
- Average $WAIT time per $QSUSE macro

WLM Sampling data - $D PERFDATA(SAMPDATA)

This displays a sumary of the Workload Manager (WLM) sampling data used to determine the status of service class queues used for WLM-managed initiators. (added in OS/390 Rel. 4)

- Service classes known to JES2 - registered, by system
- Sampling data for each service class, including the number of jobs in the sysplex (JESplex) with queue delay, ineligible for any system, or limited by the job class maximums.
- The count of jobs with queue delay and ineligible on the local system is also provided.
- Report class sampling data is also given for the sysplex (JESplex).

Subtask Statistics - $D PERFDATA(SUBTSTAT)

This displays, for each JES2 subtask, the count, average queue time, run time, and CPU time. (added with APAR OW55693)

Detailed Message Descriptions:

Checkpoint Statistics

$D PERFDATA(CKPTSTAT) displays a summary of the checkpoint statistics that was available through the $TRACE ID=17 data. (added in z/OS Rel. 2)

```
$HASP660 CKPT PERFORMANCE STATISTICS - INTERVAL=11:10:12.320961,
$HASP660 AVGHOLD=0.318337,AVGDORM=45.305289,TOT$CKPT=3284,
$HASP660 WRITE-4K=0,WRITE-CB=788,OPT$CKPT=2496,OPT4K=0,
$HASP660 IO=R1,COUNT=875,AVGTIME=0.010943,
$HASP660 IO=R2,COUNT=0,AVGTIME=0.000000,TOTAL4K=0,TOTALCB=118,
$HASP660 IO=PW,COUNT=876,AVGTIME=0.004066,TOTAL4K=39,TOTALCB=0,
$HASP660 IO=IW,COUNT=878,AVGTIME=0.003776,TOTAL4K=0,TOTALCB=670,
$HASP660 IO=FW,COUNT=876,AVGTIME=0.003888,TOTAL4K=0,TOTALCB=118
```

The first line shows the elapsed time (**INTERVAL=**) and **CPU** time for the JES2 main task was active since JES2 was started or since the last $T PERFDATA(PCESTAT),RESET was issued.

AVGHOLD	Average hold time in seconds
AVGDORM	Average dormancy time in seconds
TOT$CKPT	Total number of $CKPTs issued
WRITE-4K	Number of 4K pages written in intermediate and final writes
WRITE-CB	Number of control blocks in change log in intermediate and final writes
OPT$CKPT	Number of $CKPTs that were optimized (dual mode)
OPT4K	Number of 4K pages that were optimized (not written out because of the new optimization code in z/OS Release 2)
IO	Name of the I/O operation
COUNT	I/O count
AVGTIME	Average I/O time in seconds
TOTAL4K	Number of 4K pages read/written
TOTALCB	Number of control blocks in the change log

Notes:

- IO=R1 (read 1) never has a 4K or CB (control block) count. These are reflected in the IO=R2 record.

- IO=R2 (read 2) data includes CBs read by R1 in the change log (thus it can have a count even if no I/Os were done).
- TOT$CKPTs include multiple $CKPTs for the same CB. WRITE-CB and TOTALCB do not.
- WRITE-4K and WRITE-CB do NOT include the data written in the primary write.
- PW (primary write) numbers are not included in any of the summary numbers that are given.

CPU Statistics

$D PERFDATA(CPUSTAT) displays CPU statistics for each PCE type without the detailed wait and post statistics. The entries are displayed in order of decreasing CPU time. (CPUSTAT was added in OS/390 Release 7.)

```
$HASP660 $DPERFDATA(CPU*)
$HASP660 CPU PERFORMANCE STATISTICS - INTERVAL=139:40:17.064328,
$HASP660 CPU=6:08.263384,
$HASP660 PCENAME=CKPT,CPU%=56.25,CPU=3:27.184213,TIME=3:41.326383,
$HASP660 QSUSE_TIME=0.026951,IOCOUNT=28294,CKPT_COUNT=195792,
$HASP660 PCENAME=SPOOL,CPU%=8.74,CPU=32.211999,TIME=34.196686,
$HASP660 QSUSE_TIME=33.777164,IOCOUNT=0,CKPT_COUNT=6592904,
$HASP660 PCENAME=TIMER,CPU%=6.79,CPU=25.039045,TIME=27.088692,
$HASP660 QSUSE_TIME=0.000000,IOCOUNT=0,CKPT_COUNT=0,
$HASP660 . . . etc.
```

The first line shows the elapsed time (**INTERVAL=**) and **CPU** time for the JES2 main task was active since JES2 was started or since the last $T PERFDATA(PCESTAT),RESET was issued.

PCENAME=	The name of the PCE type as defined on the $PCETAB statement
TIME=	Elapsed time all PCEs of this type were JES2 dispatched.
CPU=	CPU time that all PCEs of this type have used. (obtained via TIMEUSED macro)
CPU%=	Percentage of the total CPU time used by the JES2 main task.
QSUSE_TIME=	Elapsed time this PCE type held the checkpoint (queues). Added in OS/390 R. 7.
IOCOUNT=	Total number of I/Os that this PCE type has issued (If the I/O is still in progress, it is counted).
CKPT_COUNT=	Number of $QSUSE macros issued. Added in OS/390 Release 7.

Event Statistics

$D PERFDATA(EVENT) tracks the latest 100 significant events such as $DISTERR errors and ABENDs ($ERRORS) and long-running PCEs running for more than 5 seconds. (added in OS/390 Rel. 10)

```
$HASP660 EVENT INFORMATION - INTERVAL=18:43.320667,
$HASP660 TIME=2000.123,20:46:15.60,EVENT=LONG PCE DISPATCH,PCE=
$HASP660 COMM,MOD=HASPNUC,28570000,DATA=$POJQ(*),DURATION=51.909227,
$HASP660 TIME=2000.123,20:51:54.73,EVENT=ABEND/$ERROR,PCE=PRT,
$HASP660 MOD=UNKNOWN,+ 000000,DATA=S0C1,
$HASP660 TIME=2000.123,21:01:28.61,EVENT=$DISTERR,PCE=STCINRDR,
$HASP660 MOD=HASPNUC,68689400,JOB=STC00008,DATA=CBIMPL4,
…
```

Initialization Statistics

$D PERFDATA(INITSTAT) shows the CPU and wall clock time for each Initialization Routine (IR) module, WARM start processing, and MVS initialization from JES2 start until HASPNUC is first entered. IR routines are presented in order of execution. (added via APAR to JES2 SP510.)

```
$HASP660 STATISTICS FROM INITIALIZATION:
$HASP660 ROUTINE=MVSSTART,TIME=1.653267,CPU=0.003280,
$HASP660 ROUTINE=LOADINIT,TIME=0.052663,CPU=0.001788,
$HASP660 ROUTINE=IRMODCHK,TIME=0.000273,CPU=0.000261,
```

```
 ... etc.
$HASP660 ROUTINE=IRFINAL,TIME=0.001327,CPU=0.001215,
$HASP660 ROUTINE=WARMSTRT,TIME=0.312197,CPU=0.012259
```

In the above example,

ROUTINE=	Name of the IR routine.
TIME=	Elapsed (wall clock) time it took for the IR routine to complete. This time includes time spent waiting for a WTOR to be replied to.
CPU=	CPU time used by the listed IR routine. This time is obtained by using the TIMEUSED MVS macro.

In addition to the normal IR routines listed in HASPIRA, 3 additional entries are displayed:

MVSSTART	Time from JES2 start processing until HASPNUC is entered (at label HASP). CPU= is the total CPU time for the main task on entry to HASPNUC. TIME is entry STCK time minus ASCBINTS (Job selection time stamp).
LOADINIT	Time from entry in HASPNUC until HASPIRA starts calling other IR routines (CPU and wall clock time).
WARMSTRT	Total CPU time and wall clock time from entry to HASPWARM for this start until HASPWARM completes processing. It includes CPU time used by all PCEs during that time period.

PCE Statistics

$D PERFDATA(PCESTAT) was added in JES2 SP520. The following is a sample display of the data. There is one entry for each PCE type (unique PCETAB/PCEID). The data is displayed in PCEID order (Second byte as primary sort key, first byte as secondary).

```
$HASP660 PCE PERFORMANCE STATISTICS - INTERVAL=139:29:32.591057,
$HASP660 CPU=6:07.730259,
$HASP660 PCENAME=INTRDR,TIME=0.386847,CPU=0.278241,CPU%=0.07,
$HASP660 QSUSE_TIME=0.206887,IOCOUNT=2134,CKPT_COUNT=7240,
$HASP660   WAIT=IO,MOD=HASPNUC,SEQ=17000000
$HASP660    COUNT=1102,AVGWAIT=0.023403,
$HASP660      POST=IO,COUNT=1102,AVGWAIT=0.023403,
$HASP660   WAIT=CKPT,MOD=HASPNUC,SEQ=28410000
$HASP660    COUNT=28,AVGWAIT=0.705648,CMOD=HASPRDR,CSEQ=56718400,
$HASP660      POST=RESOURCE,COUNT=28,AVGWAIT=0.705648,
$HASP660   WAIT=CKPT,MOD=HASPNUC,SEQ=28410000
$HASP660    COUNT=17,AVGWAIT=0.792405,CMOD=HASPRDR,CSEQ=89178800,
$HASP660      POST=RESOURCE,COUNT=17,AVGWAIT=0.792405,
$HASP660   WAIT=CKPT,MOD=HASPNUC,SEQ=28410000
$HASP660    COUNT=225,AVGWAIT=0.326621,CMOD=HASPJQS,CSEQ=03330000,
$HASP660      POST=RESOURCE,COUNT=225,AVGWAIT=0.326621,
$HASP660   WAIT=POST,XECB,MOD=HASPNUC,SEQ=99676000
$HASP660    COUNT=549,AVGWAIT=0.002100,
$HASP660      POST=XECB,COUNT=549,AVGWAIT=0.002100,
$HASP660   WAIT=OPER,XECB,MOD=HASPRDR,SEQ=02226200
$HASP660    COUNT=7,AVGWAIT=1:21.131489,
$HASP660      POST=XECB,COUNT=7,AVGWAIT=1:21.131489,
$HASP660   ... etc.
```

For each PCE type, there are 3 types of records listed. The first is the PCE type statistics. This lists general data about a PCE type. The second type of record is the $WAIT data. There is one of these for every $WAIT that is issued. Under each $WAIT entry is a $POST entry. This lists the type of $POST that woke a PCE out of a particular $WAIT.

Because the amount of data for all PCE types can be very large, it is possible to filter the display based on PCENAME=. This lists data for all PCE types whose name match the pattern passed. For example,

- $D PERFDATA(PCESTAT),PCENAME=CKPT only lists the CKPT PCE.
- $D PERFDATA(PCESTAT),PCENAME=NET* list all network type PCEs.

NOTE: a PCE is only displayed if there is non-zero data to display. '*' and '?' can be used for pattern matching on PCENAME=.

The same data shown above for CPUSTATs is also displayed for PCE level records. In addition, the following data is displayed for WAIT level records:

WAIT=	The parameters passed on a $WAIT call for a PCE of this type. • If XECB= was specified on the $WAIT call, then 'XECB' will appear after event or resource $WAITed on. • If INHIBIT=NO was specified on the $WAIT call, then 'INHIBIT=NO' will also appear on this line. • If an unknown (or installation defined) resource type or event is $WAITed on, then WAITTYPE will display as RESxx and EVENTxx respectively. The xx is the value that the resource/event would be equated to in $HASPEQU.
MOD=	Module where the $WAIT was issued
SEQ=	Sequence number on the last card of the $WAIT macro
COUNT=	Count of how many times a PCE of this type $WAIT at this location (if a PCE is currently $WAITING at this location, it is not represented in this count. The count and the wait time are only updated when a PCE is posted out of a $WAIT).
AVGWAIT=	Average time PCE of this type has $WAITed for this $WAIT. The $WAIT that a PCE is currently $WAITing on is NOT included in these numbers.
CMOD=	Name of the calling JES2 module which issued the $QSUSE macro. Added in OS/390 Rel.8.
CSEQ=	Sequence number on the $QSUSE macro causing this $WAIT to be called. Added in OS/390 Rel.8.

The following data is displayed for POST level records:

POST=	Type of $POST that woke the PCE up from the previously listed $WAIT. This will display 'RESOURCE' for any resource $POST, 'XECB' for MVS posting an XECB, $$POST for posts of an element via $$POST, 'FORCE' is the PCE was posted FORCE, or the EVENT(s) specified on the $POST if this was a event post.
COUNT=	Count of how often this post type woke the a PCE of this type up at this $WAIT.
AVGWAIT=	Average time a PCE of this type was $WAITing when it was dispatched after a $POST of this type.

$QSUSE Statistics

The $QSUSE macros are listed in order of usage (highest used macro first): (QSUSE was added in JES2 SP520.) The following is a sample display of the data:

```
$HASP660 $DPERFDATA(QSUSE)
$HASP660 $QSUSE UTILIZATION STATISTICS - INTERVAL=139:50:08.119825,
$HASP660 MOD=HASPJQS,SEQ=03330000,COUNT=229,AVGWAIT=0.329105,
$HASP660 MOD=HASPXEQ,SEQ=03510000,COUNT=156,AVGWAIT=0.272880,
$HASP660 MOD=HASPTRAK,SEQ=04479900,COUNT=93,AVGWAIT=0.924224,
$HASP660 MOD=HASPSXIT,SEQ=12574100,COUNT=74,AVGWAIT=0.277919,
$HASP660 MOD=HASPHOPE,SEQ=11950000,COUNT=70,AVGWAIT=0.939371,
$HASP660 MOD=HASPJQS,SEQ=46496600,COUNT=67,AVGWAIT=0.909164,
$HASP660 MOD=HASPTRAK,SEQ=05025900,COUNT=46,AVGWAIT=0.907750,
```

```
$HASP660 MOD=HASPSPIN,SEQ=05463000,COUNT=46,AVGWAIT=0.292662,
$HASP660 MOD=HASPNJT,SEQ=17172000,COUNT=29,AVGWAIT=0.367108,
$HASP660 MOD=HASPRDR,SEQ=56718400,COUNT=28,AVGWAIT=0.705652,
. . .
```

In the example, the first line shows the elapsed time (INTERVAL=) that the JES2 main task was active since JES2 was started or since the last $T PERFDATA(QSUSE),RESET was issued. For each occurance of $QSUSE macro issued, the following information is displayed for all PCEs:

MOD=	Module name that contains the $QSUSE macro instruction
SEQ=	Sequence number (col 73-80) of the last line of the macro
COUNT=	Number of times the listed $QSUSE macro $WAITed
AVGWAIT=	Average $WAIT time (in seconds).

WLM Sampling Data

Workload manager (WLM) sampling data is tracked for the WLM management of batch initiators. (SAMPDATA) was added with OS/390 Version 2 Release 4.)

```
$HASP660 SERVICE CLASSES KNOWN TO JES2:
$HASP660 SRVCLASS(HOTPRIME)=(TOKEN=0F4A8000,REGISTERED,SYSTEMS=(AQFT,AQTS)),
$HASP660 SRVCLASS(NRPRIME)=(TOKEN=064A8000,REGISTERED,SYSTEMS=(AQFT,AQTS)),
 ...
$HASP660 SERVICE CLASS SAMPLING DATA:
$HASP660 SRVCLASS(15)=(SYS_QUEUE=0,SYS_INEL=18,SYS_LIMIT=0,
$HASP660 LOCAL_QUEUE=0,LOCAL_INEL=18),
$HASP660 SRVCLASS(16)=(SYS_QUEUE=0,SYS_INEL=2,SYS_LIMIT=0,
$HASP660 LOCAL_QUEUE=0,LOCAL_INEL=2)
$HASP660 REPORT CLASS SAMPLING DATA:
 . . . .
```

SERVICE CLASSES KNOWN TO JES2: In the above example, the first few lines show the Service Classes known to JES2:

SRVCLASS( )	Service Class Name
TOKEN=	Classification token which can be mapped to a service class index and report class index.
REGISTERED, SYSTEMS=	List of systems in the JESplex which have this service class registered.

SERVICE CLASS SAMPLING DATA: For each of the Service Classes for which there was a sample, the following data is passed to WLM by JES2 during each checkpoint cycle and displayed here:

SRVCLASS(nn)	Index into the service class
SYS_QUEUE=	Sysplex queue delay - Jobs eligible to be initiated somewhere in the JESplex.
SYS_INEL=	Sysplex ineligible - Jobs not eligible to be initiated on any system in the JESplex. This includes operator held jobs, jobs held because of duplicate name, jobs busy on a JES transmitter (offload and keep).
SYS_LIMIT=	Sysplex limited - Jobs are not eligible because the JES job queue(s) that are related to these jobs have reached their limit of executing jobs. See limit on the JES class initialization statement.
LOCAL_QUEUE=	Local queue delay - Jobs eligible to be initiated on this system.
LOCAL_INEL=	Local ineligible - Jobs not eligible to be initiated on this system. This includes operator held jobs, jobs held because of duplicate name, jobs busy on a JES transmitter (offload and keep).

REPORT CLASS SAMPLING DATA: Similar to the Service Class sampling data shown above, the same data is also listed for Report Classes, at the JESplex level. (SYS_QUEUE=, SYS_INEL=,and SYS_LIMIT=)

Note that it is possible for two or more service classes to map into a single report class.

Subtask Statistics

$D PERFDATA(SUBTSTAT) was added with APAR OW55693. The following is a sample display of the data. There is one entry for each JES2 subtask. The data is displayed in subtask routine name order (alphabetical).

```
$HASP660 $DPERFDATA(SUBTSTAT)
$HASP660 SUBTASK STATISTICS - INTERVAL=497:38:13.858921,
$HASP660 ROUTINE=$RACROUT,COUNT=998,AVG_QUEUE_TIME=0.000574,
$HASP660 AVG_RUN_TIME=0.000019,AVG_CPU_TIME=0.000017,
$HASP660 ROUTINE=FSMGCR,COUNT=3,AVG_QUEUE_TIME=0.000035,
$HASP660 AVG_RUN_TIME=0.002200,AVG_CPU_TIME=0.001506,
$HASP660 ROUTINE=FULLPATH,COUNT=14,AVG_QUEUE_TIME=0.000143,
$HASP660 AVG_RUN_TIME=0.288402,AVG_CPU_TIME=0.263313,
$HASP660 ROUTINE=JOBVALM,COUNT=1001,AVG_QUEUE_TIME=0.000038,
$HASP660 AVG_RUN_TIME=0.001632,AVG_CPU_TIME=0.000222,
$HASP660 ROUTINE=PSAFSCAN,COUNT=1774,AVG_QUEUE_TIME=0.000199,
$HASP660 AVG_RUN_TIME=0.000153,AVG_CPU_TIME=0.000065,
$HASP660 ROUTINE=RPDBSEC,COUNT=1001,AVG_QUEUE_TIME=0.000101,
$HASP660 AVG_RUN_TIME=0.000107,AVG_CPU_TIME=0.000103,
$HASP660 ROUTINE=XINITST,COUNT=16,AVG_QUEUE_TIME=0.000581,
$HASP660 AVG_RUN_TIME=0.004094,AVG_CPU_TIME=0.002164
```

Usage Tips:

Resetting Performance Statistics

The CKPTSTAT, CPUSTAT, EVENT, PCESTAT and QSUSE statistics start accumulating as soon as JES2 is initialized. However, it is possible to reset the counts/times to zero via a $TPERFDATA(<stat-type>),RESET command. The subscript is the information you wish to reset. $T PERFDATA(QSUSE) resets the $QSUSE information. Since the initialization statistics are only set during JES2 initialization, they are not reset. CPUSTAT is reset when PCESTAT is reset. SAMPDATA is a snapshot of the current sampling data and can be considered to be constantly being reset.

Displaying Excessive Lines:

To circumvent the CONDEF DISPMAX limitations, specify "L=Z" to send the output to the "out of line" display area. This avoids the $HASP003, RC=(57) error message and termination of the display. (This is no longer necessary in OS/390 Release 7 because the PERFDATA is exempt from DISPMAX.)

Capturing JES2 Subtask CPU Time:

The 'D A,JES2' command captures the JES2 address space Elapsed and CPU times accumulated. This can be used in conjunction with the $D PERFDATA(CPUSTAT) command. The JES2 subtask times can be obtained by subtracting out the PCE (JES2 main task) times from the change in times displayed by the IEE115I messages (output from the "D A,JES2" command.)

Coordinating Statistics from Multiple Members:

To get PERFDATA statistics for the same period on all systems in the MAS, issue these commands using the MVS ROute command to synchronize on all members.

Sample Usage Scenario:

At the start:

```
 RO *all,$T PERFDATA(*),RESET   < to reset the other perfdata stats
 RO *all,D A,JES2               < to get JES2 AS Elapsed & CPU times - base #s
```

After 10 minutes:

```
 RO *all,D A,JES2               < to get JES2 AS ET & CT updated #s
 RO *all,$D PERFDATA(*),L=Z     < to get updated statistics for the interval
```