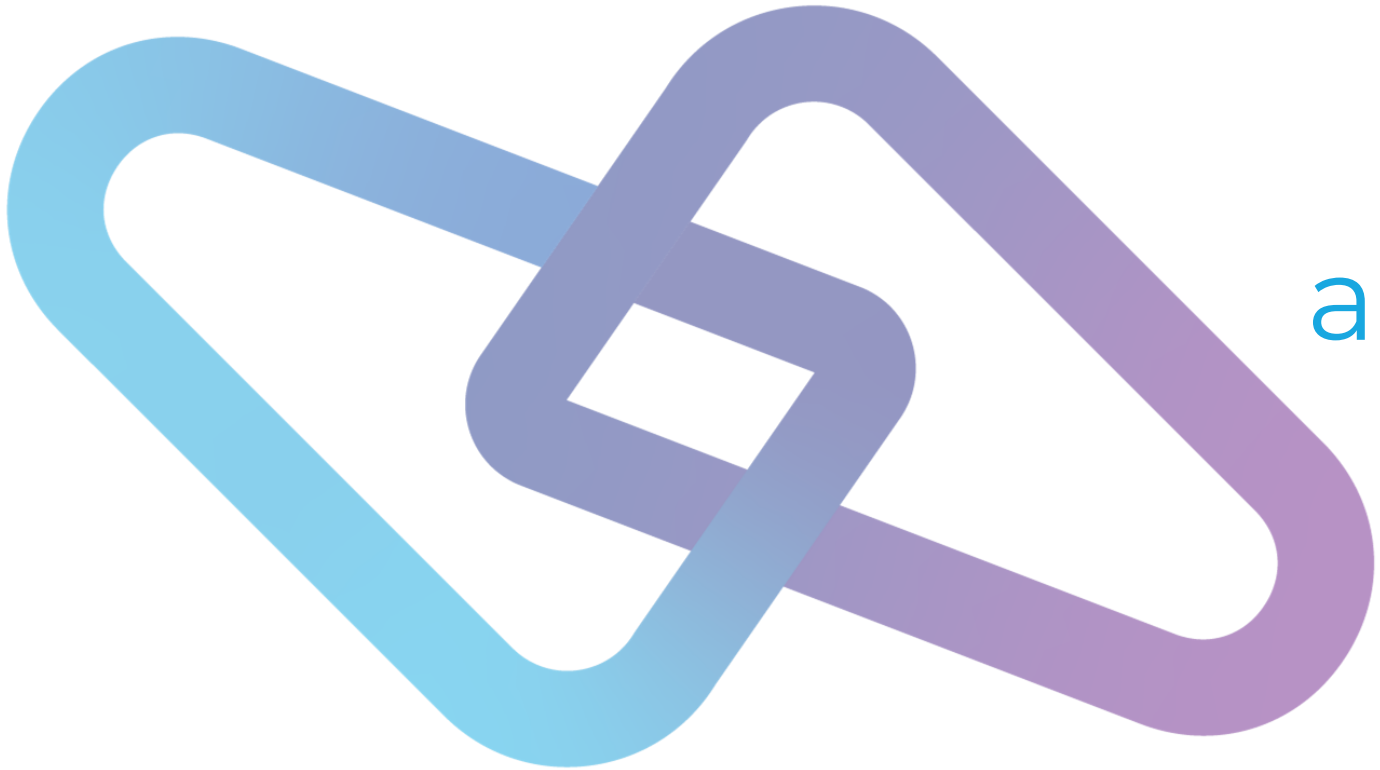


Mastering tracing and debugging via SDAID

Vasilisa Suvorova
June 2024

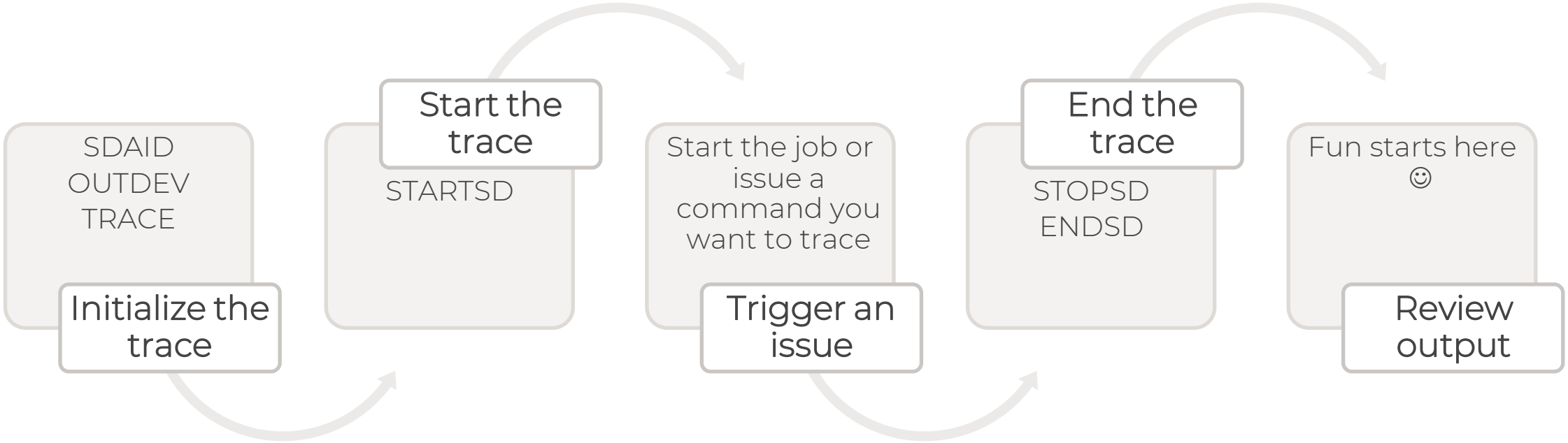
Agenda

- Why do we need SDAID
- An overview of debugging process
- Tracing to different output devices – examples
- SDAID vs CP TRACE
- Recent updates and known issues
- Future improvements under consideration
- Q&A

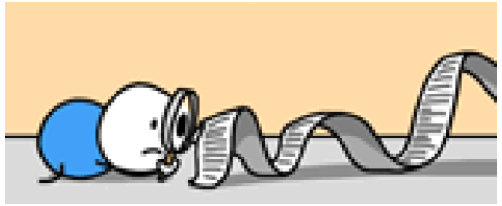
Why do we need SDAID

- To track events causing error
- To isolate the problem
- To track memory leaks
- To understand the code logic
- To track I/O operations

Overview



What events can be traced



Instruction	Successful branch	Storage alteration	EXTERNAL interrupts
GETVIS/FREVIS	LOCK/UNLOCK	PGMLOAD	SVC
I/O	VTAMIO	...and many more	

Where does SDAID write to

- Nice option if your system is running under z/VM

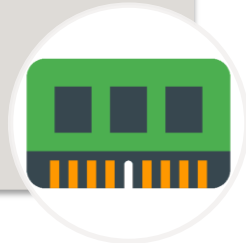
Printer

- No VTAPE support!
- DOSVSDMP utility needs to be used to print data from tape

Tape

- VTAPE needs to be used to dump buffer to VSAM
- DOSVSDMP utility needs to be used to print data from VTAPE

Buffer



Defining the Area to be Traced

You define the area for which processing is traced by using the AREA or JOBNAME definition. Either AREA or JOBNAME can be specified in the TRACE statement, but not both.

- AREA
  - Partition_id Specifies the partition to be observed like BG, F3, Y1
  - SUP The activities of the supervisor are to be traced
  - ALL Specifies that the activities of the entire VSEn system are to be traced
- JOBNAME
  - Jobname Specifies the name of the VSEn/POWER job to be traced
    - ➢ Jobnum Specifies the VSEn/POWER-defined job number

Examples:

```
TR BR  JOBNAME=MYJOB
TR SVC=15  AREA=BG
```

Defining the Storage to be Traced

In addition, you may use one of the following storage area definitions for the AREA and JOBNAME definitions:

- OFFSET — Limits tracing to a certain address range via offsets relative to the partition start address (or phase loadpoint)
- ADDRESS — Limits tracing to a certain address range within the storage allocated to VSEn
- PHASE — Limits tracing to the specified phase
- LTA — Defines that the Logical transient area is specified as tracing range

Examples:

```
TR SVC=* AREA=F3 ADDR=600000:600200
TR STOR AREA=BG PHASE=TEST2 OFF=100:200
```

Defining Additional Trace Output: OUTPUT Definition

In addition to the basic trace event record, you may specify additional trace output which is recorded together with the trace event records.

The following are the most important additional trace output definitions:

- **TOD** Time-of-Day clock
- **GReg** General purpose and access registers (if available)
- **CReg** Control registers
- **FReg** Floating point registers
- **DUMP** Virtual storage
- **CCW** CCWs, IRB (TRACE=IO, SSCH, or VTAMIO only)
- **CCB** CCB or IORB (TRACE=IO, SSCH, or VTAMIO only)

Examples:

```
TR  INSTR=* ADDR=5000:5010 OUTP=(DUMP ADDR=4000:4080)
TR  IO  Unit=01E  OUTP=(CCB CCW TOD)
```

Defining the Trace Options: OPTION Definition

The following trace option definitions are available:

- NOJCL — Suppresses tracing of Job Control phases.
- Halt — Put the system into a wait state when defined trace event occurs
- Terminate — Allows to terminate SDAID output at the occurrence of a specified event. You may start the trace output again if you issue the STOPSD/STARTSD commands.
- OCCurence=y:z — Specifies the number of associated events to be traced.
- SUPervisor — Traces a code segment within the supervisor, It allows to trace supervisor routines while they are working for a user partition. You may use OPT=SUP if you specify AREA=Partition_id or JOBNAME=jobname

Trace definitions: more examples

```
TRACE BR ADDR=0020BE10:00243162
TRACE INST=BR JOBNAME=MYTSTJOB OPT=(OC=1:20)
TRACE INST=(D2 92) PHASE=MYTEST OUTP=GR
TRACE INST=* AREA=F3 ADD=5000:5200 OPTION=SUP
TRACE GETVIS=PAR AREA=BG OUTP=(DUMP REG=1:8, GR)
TRACE SVC=(3D 3E) AREA=BG OUTP=(DUMP PTR=5:20 DMP=0:16, GR)
TRACE IO AREA=BG UNIT=241 OUTP=(CCB CCWD=80)
```

SVA

Tracing in specified range if only supervisor code is executed from F3

Be careful – if memory is not available SDAID will silently stop due to the program check
(this is a bad example that may fail)

PTR=REG:Offset DMP=Offset:Length

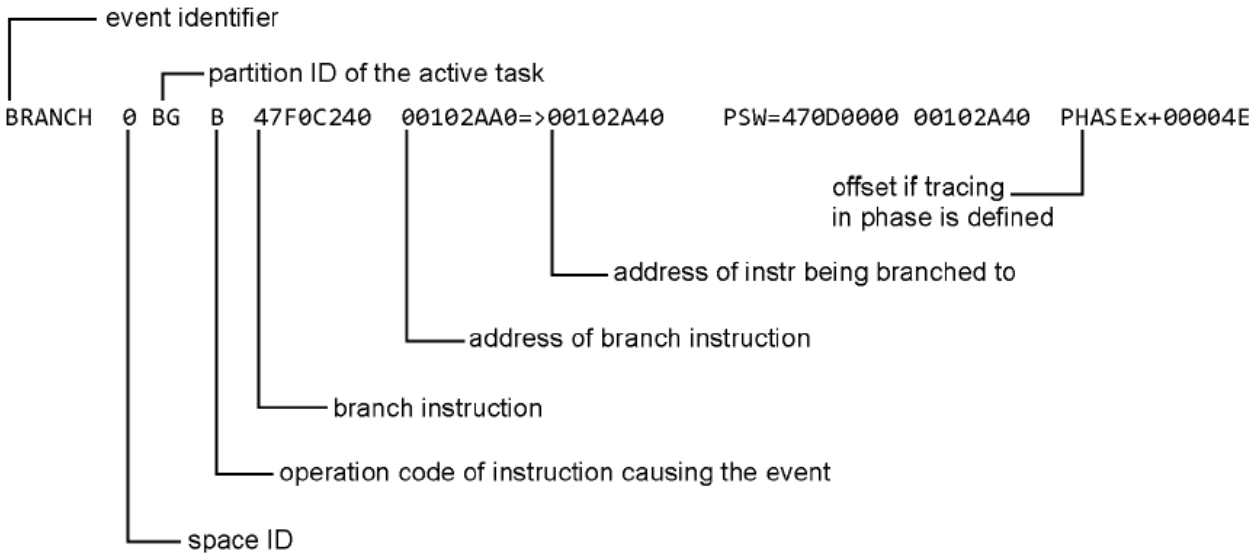
SDAID example – trace initialization via JCL

```
* $$ JOB JNM=SDAID,CLASS=6,DISP=D
* $$ LST CLASS=V,DISP=D,DEST=(,VASILIS)
* $$ PUN CLASS=V,DISP=D,DEST=(,VASILIS)
// JOB SDAID    VASILIS
* CP DEF PRT 02E                          <- define a printer
* CP SPOOL 02E VASILIS                    <- direct output to your user
* ****
* To Stop SDAID issue ENDSD command
* ****
// EXEC SDAID                             <- start trace initialization
OUTDEV P=02E                              <- output to printer
TRACE BR PHASE=$IJBCSIO OUTP=GR           <- trace Branches
/*
// EXEC DTRIATTN,SIZE=AUTO,PARM='STARTSD'  <- start tracing
/*
/&
* $$ EOJ
```

02E should be defined to your VSEn in Hardware Configuration table

SDAID example – output (on VM RDR)

```
TRACE BR PHASE=$IJBCSIO OUTP=GR
BRANCH S  AR  C00  BASSM  0CEF          0005DF46=>0A4346A8    PSW=440C1000 8A4346A8
GR 0-7   00000001 0025FF8C 00000001 00000000 00000010 0006F804 00000001 0006C3B6
   8-F   00064AD0 0AFF4000 0006C1F0 00014CE0 A005DAB0 0AFF4010 8005DF48 8A4346A8
BRANCH S  AR  C00  BRC    A7F4000C       0A4346A8=>0A4346C0    PSW=440C1000 8A4346C0 $IJBCSIO+000280
GR 0-7   00000001 0025FF8C 00000001 00000000 00000010 0006F804 00000001 0006C3B6
   8-F   00064AD0 0AFF4000 0006C1F0 00014CE0 A005DAB0 0AFF4010 8005DF48 8A4346A8
...
```



Tracing to real tape

- Initialize a trace in Direct input mode (via **AR commands**) – see next slide
- OUTDEV TAPE=cuu
- Mount a tape to PART=SYSTEM
- Start/Stop the trace as usual
- Use EXEC DOSVSDMP or FP 467 to print trace from tape
- See the output in LIST queue (or via Navigator):

```
06/28/2023
PRINTOUT OF SDAID TAPE. DUMP FILE: 1
BRANCH S  AR  C00  BZ     4780A2AA      20EE9272=>20EE92CA    PSW=450C0000 A0EE
BRANCH S  AR  C00  BZ     4780A304      20EE92CE=>20EE9324    PSW=450C0000 A0EE
BRANCH S  AR  C00  BZ     4780A33C      20EE9332=>20EE935C    PSW=450C0000 A0EE
BRANCH S  AR  C00  BZ     4780A372      20EE9384=>20EE9392    PSW=450C0000 A0EE
BRANCH S  AR  C00  BZ     4780A384      20EE9396=>20EE93A4    PSW=450C0000 A0EE
BRANCH S  AR  C00  BAS    4DF0B9D4      20EE93D6=>20EEA9F4    PSW=450C2000 A0EE
BRANCH S  AR  C00  BZ     4780B9EA      20EEA9FE=>20EEAA0A    PSW=450C0000 A0EE
BRANCH S  AR  C00  BNP    47D0BA2E      20EEAA2E=>20EEAA4E    PSW=450C1000 A0EE
```

Tracing to real tape – example (AR)

```
LIBSERV MOUNT,VOL=V30260/W,PART=SYSTEM,UNIT=22A
AR 0015 1I40I  READY
AR 0024 AOMAP00I LIBRARY INFORMATION CUU=022A,  LIB=02433903 ,CLUSTER=00
AR 0024 AOMAP20I MOUNT COMPLETE VOLID=V30260, VISION=V30260, CAT=INSERT
AR 0015 1YH2I MOUNT    FINISHED FOR UNIT 22A
sdaid
AR 0015 4C05I PROCESSING OF SDAID   COMMAND SUCCESSFUL
AR 0015 1I40I  READY
OUTDEV TAPE=22A
AR 0015 4C05I PROCESSING OF OUTDEV  COMMAND SUCCESSFUL
AR 0015 1I40I  READY
TRACE BR PHASE=$IJBSPDT
AR 0015 4C05I PROCESSING OF TRACE   COMMAND SUCCESSFUL
AR 0015 1I40I  READY
ready
AR 0015 4C05I PROCESSING OF READY   COMMAND SUCCESSFUL
AR 0015 1I40I  READY
startsd
AR 0015 4C36I SDAID SETS OFF THE PSEUDO PAGE FAULT PORTION
AR 0015 4C05I PROCESSING OF STARTSD COMMAND SUCCESSFUL
AR 0015 1I40I  READY
```

Tracing to the memory buffer

- OUTDEV BUFFER=nnn (up to 256K)
- Initialize VSAM dataset for VTAPE (SKVTAPE in ICCF lib #59)
- Set up the trace
- Start a VTAPE and DUMP the buffers to VSAM dataset
- Print the trace from VTAPE via DOSVSDMP
- See article with example here:

https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/jens-remus1/2021/02/18/how-to-perform-a-sdaid-trace-to-buffer-in-zvse

Things to keep in mind

- You can set up to 10 traces per session (depending on trace type).
- Performance degradation under tracing is expected - may not work for timing issues
  - Be especially careful when defining more then one trace per session. The following trace setup has a very high impact on the performance:

    TRACE STOR ADDR=47C:47C

    TRACE INST=* ADDR=21FFFF0:21FFFF0
- Be careful when tracing to buffers, ENDSD command releases all resources that were used during the session (including buffers).
- For more info, please refer to the *“VSEn 6.3 Diagnosis Tools”* guide.

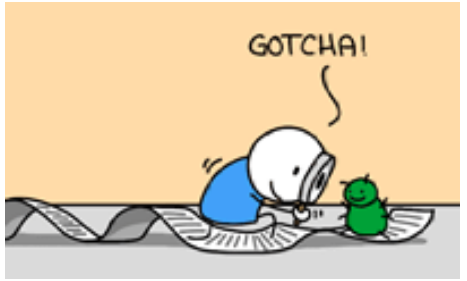
SDAID vs CP Trace

SDAID

- Great for tracing
- Interactive debugging is possible with HALT option
- PHASE name or partition can be specified
- GETVIS or LOCK traces are available with more into in a trace record
- Tracing with extra display to RDR/Tape

CP TRACE

- Great for interactive debugging
- Tracing is possible with RUN PRINT options (or console can be “recorded”)
- No info about guest jobs/partitions – address range required
- Only SVC trace is available



Recent and upcoming updates

VA00138 (VP00132) - out

- INSTR trace prints ?????? for instructions introduced with IBM System z9 and later.

VA00148 (VP00147) – coming soon

- The Storage alteration trace may skip or incorrectly process events caused by the instructions, introduced with IBM System z9 and later. The following facilities are addressed in this PTF:
  - Execute-extensions
  - General-instructions-extension
  - Long displacement

VA00161 (VP00163) – coming soon

- SDAID loops when processing Storage alteration trace with OUTDEV P= if trace catching EX/EXRL instructions.

Full instruction set support (z9+) for all traces – in to do list.

Future improvements under consideration

- Better recovery instead of a silent stop on program check.
- Enhance the SDAID batch utility to allow the use of tape library devices assigned to system (PART=SYSTEM).
- Tapeless SDAID support (e.g. SDAID on disk)

Your ideas are very welcome!

Good luck!