

Understanding dump devices

sysdumpdev

Article Number: 287 | Rating: Unrated | Last Updated: Mon, Jul 9, 2018 12:45 PM

If your system crashes due to an unexpected event, it core dumps. In fact a core dump can occur without a crash. However, for this article I assume that the system goes down due to a fatal event or via a user's forced action. The dump contains contents of the memory up to the point of the crash. By its very nature a crash happens unexpectedly, therefore it is up to the system administrator to prepare for the event in advance for when it happens. You can tell if a crash has happened because your system has re-booted, and there are entries in you error log with the label: SYSDUMP.

For this demonstration, I am using AIX 7.1. However the principals I discuss apply to AIX 5.3 and 6.1 as well.

Prepare, prepare

To prepare for the unexpected system crash you need to make sure you have a dump device logical volume (LV) where the dump is placed when the system comes back up. However, if that dump device is not available, then a secondary dump device should be assigned for the placement of the dump. It may be the case that one does not care about when the system crashes and thus is not interested in the dump file for further investigation. This is entirely up to the owner of the system. But, beware it is good practice and a requirement to have a primary dump device present in your rootvg for the system to operate correctly. The dump device can be mirrored, but IBM AIX support throws caution to this. This is because a crash maybe mirrored or sync related and thus invalidates the mirroring on the dump device. In certain circumstances, the dump file could be copied to only one of the copies of the mirrored dump device, that resides on the mirrored disks. It may be the case that only half the copy of the dump file is recovered when the system is restarted. A good practice is to have the primary dump device on one disk, un-mirrored, and the secondary device on the other disk, un-mirrored. However, I have found it is common to mirror the rootvg dump device. The second dump device can either be within rootvg or outside of rootvg, as long as it is not on a paging space, or an external device, like a tape device for example.

Dump devices

Traditionally the default dump device for system dumps was: /dev/hd6 (paging space) and still is on a lot of systems. If there is not enough space to copy over the dump file after a crash, then the system administrator is prompted upon restart to copy the dump file over to some removable media , like a tape or DVD. This can be time consuming and it is sometimes the case that you want to get your system back up quickly. I can sympathise with system administrators who just ignore the prompt to get the system back up due to business pressure, thus deleting the dump, so then one does not know why it crashed in the first place. If you do not have enough space on your dump device to copy the dump, then during the start-up process, the copydumpmenu menu utility is invoked to give the system administrator the opportunity to copy the dump to a removable media, for example to a tape device if present. The copydumpmenu utility can also be called from the command line when the system is up. The copy directory by default is /var/adm/ras with the file-name:vmcore.<X>.BZ , where X is a sequence number. The dump file is a BZ (BZIP) and not a Z compressed file format.

The snap command can be used to gather information about the dump file, be-sure to include the -D flag, it gathers the information from the primary dump device.

With systems now having more memory available, this has provided more flexibility as to where the primary dump device could be placed. Typically, for systems with over 4 GB of memory there is now a dedicated dump device, called: lg_dumplv

```
1                                     # lsvg -l rootvg |grep sysdump
2                                     lg_dumplv sysdump 8 8  open/syncd N/A
```

Using the sysdumpdev command, one can determine what devices are used for the system dumps.

The following output shows a system using AIX 7.1 having the lg_dumplv as its primary dump device:

```
1
2
3
4
5
6
7
8
# sysdumpdev -l
primary      /dev/lq_dumplv
secondary    /dev/sysdumpnull
copy directory /var/adm/ras
forced copy flag  TRUE
always allow dump  TRUE
dump compression  ON
type of dump     traditional
```

Looking more closely at the above output fields. Notice that an extra field is now present for AIX 6.1 onwards: type of dump. Currently set to traditional, here you can have it set at (firmware) fw-assisted, if your hardware supports it. For the secondary field, there is no dump device. This is denoted by using the sysdumpnull device. This means all system dumps are lost if it goes to that device. The copy directory is /var/adm/ras, this is where the system dump will be copied to , for either further examination, or to be copied off to go to IBM support. Note that 'always allow dump' is set to true, this must be the case if a dump is to be successfully initiated. Dump compression is on by default.

Common settings using sysdumpdev are:

- To change the primary device use: `sysdumpdev -P -p <device_name>`
- To change the secondary device use: `sysdumpdev -P -s <device_name>`
- To change the copy directory use: `sysdumpdev -D <path_name>`
- To change the always dump condition use: `sysdumpdev -k` for false, `sysdumpdev -K` for true
- To change the type of dump use: `sysdumpdev -t <fw-assisted | traditional>`

User-controlled system dump

To initiate a dump, (which reboots the system as part of its process) use the sysdumpstart command, the following command uses the primary device to place its dump:

```
1
# sysdumpstart -p
```

As this process is initiated, the system LED panel or HMC screen, on my Power 5 box displays 00c2. This indicates that the dump is in progress. Upon the restart of bootsystem, the error log could contain the following entries:

```
1
# errpt lmore
2
IDENTIFIER TIMESTAMP T C
3 RESOURCE_NAME DESCRIPTION
4 A6DF45AA 1027180611 I O RMCdaemon The
daemon is started.
5
67145A39 1027180411 U S SYSDUMP
6 SYSTEM DUMP
7
F48137AC 1027180411 U O minidump
COMPRESSED MINIMAL DUMP
A6DF45AA 1027180411 I O RMCdaemon The
daemon is started.
9DBCfDEE 1027180511 T O errdemon
```

ERROR LOGGING TURNED ON

Further investigation of the error report states:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

Type: UNKN

WPAR: Global

Resource Name: SYSDUMP

Description

SYSTEM DUMP

Probable Causes

UNEXPECTED SYSTEM HALT

User Causes

SYSTEM DUMP REQUESTED BY USER

Recommended Actions

16

PERFORM PROBLEM DETERMINATION
PROCEDURES

17

18

Failure Causes

19

UNEXPECTED SYSTEM HALT

20

Recommended Actions

21

PERFORM PROBLEM DETERMINATION
PROCEDURES

22

23

24

Detail Data

25

DUMP DEVICE

26

/dev/lg_dumplv

27

DUMP SIZE

28

63894528

29

TIME

30

Thu Oct 27 18:02:28 2011

31

DUMP TYPE (1 = PRIMARY, 2 = SECONDARY)

32

1

DUMP STATUS

0

Looking at the above output, we know the dump went to the primary dump device.

Using the following sysdumpdev command also confirms the dump took place, on the primary device. Information on the date, size, device name and if the dump was successful is also displayed:

```
1
2
3
4
5
6
7
8
9
10
11
12
# sysdumpdev -L
0453-039
Device name: /dev/lg_dumplv
Major device number: 10
Minor device number: 16
Size: 63894528 bytes
Uncompressed Size: 498002880 bytes
Date/Time: Thu Oct 27 18:02:28 BST 2011
Dump status: 0
Type of dump: traditional
dump completed successfully
```

The following will also inform you of the latest system dump, its size and location:

```
1
                                     # sysdumpdev -z
2
                                     63894528 /dev/lg_dumplv
```

The compressed dump is now on the LV lg_dumplv. The dump was not copied across to the copy directory when issuing a user initiated dump. To copy the most recent system dump from a system dump device to a directory, use the savecore command. For example, to copy the dump to the directory /var/adm/ras. I could use:

```
1
                                     # savecore -d /var/adm/ras
2
                                     vmcore.0.BZ
```

If you need to uncompress the file use the dmpuncompress utility. The format of the command is:

```
1
                                     dmpuncompress < filename>
```

After uncompressing, the dump file is now ready for further investigation using kdb or for transfer to


```
63894528
```

```
Date of last dump  
18-02-28 B>
```

```
Thu Oct 27
```

The fields are populated with the current dump that is on the primary dump device. This is the default setting, after the copy, the dump file is present in: /var/adm/ras:

```
1  
# ls -l dump_file_copy.BZ  
2  
-rw-r--r-- 1 root system 63894528 Oct 27  
18:15 dump_file_copy.BZ
```

After a dump has occurred there may well be a minidump generated as well. Contained in the errorlog output listing earlier in the article, there was an entry for:

```
1  
F48137AC 1027180411 U O minidump  
COMPRESSED MINIMAL DUMP
```

The minidump is a small compress dump that will be present in: /var/adm/ras. This file contains a snapshot of the system when the system was dumped or crashed. This file can be used for diagnosing if

the main dump is not present, due to the dump being removed or not captured.

Creating a secondary device

Earlier in this demonstration, in the 'sysdumpdev -l' output, the secondary dump device was set to /dev/sysdumpnull. This means if a dump goes to this secondary device it will be lost. It behaves much like the NULL device, everything that goes to it, goes straight into the dustbin. I will now create a secondary device and change the sysdumpdev attributes to reflect this new change. So, I can be sure that if my first dump device is unavailable, the dump goes to the secondary device.

In this demonstration, the primary device uses eight logical partitions (as shown in earlier output). So that is the amount I will create for the secondary device. However, I will first go over the actions required to size a dump device.

First we need to know the potential size of the dump AIX would generate, then using that number as a base to create the device. Using the sysdumpdev command with the 'e' option, will do a best guess of the size required. It is best to run this when the system is in normal use and not idle:

```
1                                     # sysdumpdev -e
2                                     0453-041 Estimated dump size in bytes: 282486374
```

Please note, if compressed is set on, the number of bytes returned by sysdumpdev is for a compressed dump and not an uncompressed file size. The previous command returns 282486374 bytes. For ease of use, lets convert that number to MB:

```
1                                     # expr 282486374 / 1024 / 1024
2                                     269
```

Next, add on approx 50%, (about 135 MB) to allow for a crash size if the system is overloaded, which bring us to a size of 404MB. This is the figure I will aim for at the minimum when creating the dump device. Please note also, that the file-system it will be copied to should have at least that amount of space free or the copy will fail.

First, make sure the primary device, lg_dumplv, is not mirrored in rootvg and is only residing on one disk. The secondary disk can then be placed on the other disk. From the following output, we can determine that there is only one copy of lg_dumplv:

```
1
2
3
4
5
6
7
8
# lsvg -l rootvg
rootvg:
LV NAME      TYPE  LPs  PPs  PVs
LV STATE    MOUNT POINT
hd5          boot  1    2    2  closed/syncd
N/A
...
...
livedump     jfs2  2    4    2
open/syncd  /var/adm/ras/livedump
lg_dumplv    sysdump  8    8    1
open/syncd  N/A
```

Next, query the LV `lg_dumplv` to see what disk or disks it resides on. From the following output we can see that the `lg_dumplv` only resides on `hdisk0`. So all is good. Now the secondary device can be created and will reside on the rootvg disk: `hdisk1`.

```
1
2                                     # lslv -m lg_dumplv
3
4                                     lg_dumplv:N/A
5
6                                     LP  PP1 PV1          PP2 PV2          PP3
7                                     PV3
8
9                                     0001 0008 hdisk0
10                                    0002 0009 hdisk0
11                                    0003 0010 hdisk0
12                                    0004 0011 hdisk0
13                                    0005 0012 hdisk0
14                                    0006 0013 hdisk0
15                                    0007 0014 hdisk0
16                                    0008 0015 hdisk0
```

To determine how many logical partitions (LP) to use to create the secondary device query the rootvg volume group and note the PP size. In the following output it is 128MB in size.

```
1
2                                     # lsvg rootvg
```

```
2          VOLUME GROUP:   rootvg           VG
          IDENTIFIER: 00c23bed00004c00000
3
          0013142b3b106
4
          VG STATE:      active           PP SIZE:
5          128 megabyte(s)
6          VG PERMISSION: read/write      TOTAL
          PPs:   270 (34560 megabyte
7
          ...
          ...
```

So to create a LV of at least 404 MB, I would need four partitions, (this would be a LV size of 512 MB). The command to create the LV is `mklv`. The basic format for a system dump type using the `mklv` command is:

```
1
          mklv -t sysdump -y <LV name> < volume group> <
          number of LP's> <hdisk to reside on>
```

Assume the following:

- LV is called `lg_dumplv2`
- It resides on `hdisk1`
- It is created with a size of 4 partitions

The following command could be run to create the LV:

```
1 # mklv -t sysdump -y lg_dumplv2 rootvg 4 hdisk1
```

However as discussed earlier, the secondary device in this demonstration is created with the same amount of partitions as the current primary device, which is 8. The following command achieves this, with the hdisk and LV name the same as just ran in the previous mklv command:

```
1 # mklv -t sysdump -y lg_dumplv2 rootvg 8 hdisk1
```

First, confirm that it has indeed been created on hdisk1, by querying the LV lg_dumplv2:

```
1 # lslv -m lg_dumplv2
2 lg_dumplv2:N/A
3 LP PP1 PV1 PP2 PV2 PP3
4 PV3
5 0001 0003 hdisk1
6 0002 0004 hdisk1
7 0003 0005 hdisk1
```

```
8          0004 0006 hdisk1
9          0005 0007 hdisk1
10         0006 0008 hdisk1
11         0007 0009 hdisk1
          0008 0010 hdisk1
```

Though the LV is now created it is not active, it is in a closed state, this can be seen by viewing the LV's contained in rootvg:

```
1          # lsvg -l rootvg
2
3          rootvg:
4
5          LV NAME          TYPE    LPs    PPs    PVs
6          LV STATE        MOUNT POINT
7
8          hd5             boot    1      2      2    closed/syncd
9          N/A
10
11         hd8             jfs2log 1      2      2    open/syncd
12         N/A
13
14         ...
15
16         ...
17
18         lg_dumplv       sysdump 8      8      1
19         open/syncd    N/A
```



```
lg_dumplv2      sysdump  8    8    1
closed/syncd  N/A
```

The next task is to activate it, this is done by assigning it as the secondary device using the sysdumpdev command as described earlier, like so:

```
1
2
3
4
5
6
7
8
# sysdumpdev -Ps /dev/lg_dumplv2
primary      /dev/lg_dumplv
secondary    /dev/lg_dumplv2
copy directory /var/adm/ras
forced copy flag  TRUE
always allow dump  TRUE
dump compression  ON
type of dump     traditional
```

Next review rootvg, to see if it is active:

```
1
2
# lsvg -l rootvg
```

```
3          rootvg:
4          LV NAME      TYPE    LPs   PPs   PVs
5          LV STATE    MOUNT POINT
6          hd5          boot    1     2     2   closed/syncd
7          N/A
8          hd8          jfs2log 1     2     2   open/syncd
9          N/A
          ...
          ...
          lg_dumplv     sysdump 8     8     1
          open/syncd   N/A
          lg_dumplv2    sysdump 8     8     1
          open/syncd   N/A
```

All looks good, now test it by initiating a system dump to the secondary device:

```
1          # sysdumpstart -s
```

After the reboot, confirm the system dump went to the secondary device, by querying sysdumpdev, to see where the latest system dump resides:

```
1
2 # sysdumpdev -L
3
4 0453-039
5
6 Device name: /dev/lg_dumplv2
7
8 Major device number: 10
9
10 Minor device number: 18
11
12 Size: 64955392 bytes
13
14 Uncompressed Size: 502517142 bytes
15
16 Date/Time: Thu Oct 27 18:19:37 BST 2011
17
18 Dump status: 0
19
20 Type of dump: traditional
21
22 dump completed successfully
```

As can be seen from the previous output, the dump did go to the secondary device.

One can now use the savecore command, to copy the most recent dump across to a directory either for investigation or in readiness to be moved off the system.

```
1
2 # savecore -d /var/adm/ras
```

Conclusion

If your system crashes, you will want to have a record of the events up to the crash. Having dump devices to collect this information enables you to be on a good footing when logging a call with IBM, as you will have a record of the events prior to the crash.

Posted - Mon, Jul 9, 2018 12:45 PM. This article has been viewed 4776 times.

Online URL: <http://kb.ictbanking.net/article.php?id=287>