

lquerypv as hex viewer

Author : aklyachkin

Categories : [Power Systems](#)

Tagged as : [AIX](#), [hex](#), [lquerypvLVM](#)

Date : 07.02.2016

lquerypv is an undocumented "midlevel" LVM command in AIX. Usually you don't need it and never use it, but it is used by many "high-level" LVM commands to get LVM information from a physical volume.

If you run the command without any parameters, you'll get a short usage information:

```
# lquerypv
```

```
0516-108 lquerypv: PV identifier not entered. Usage: lquerypv [-h /dev/PVname] | [-p PVID [-g VGid | -N PVname] [-scPnaDdAtY]]
```

The first way to use the command (lquerypv -h /dev/PVname) shows dump of the first 256 bytes of the selected disk:

```
# lquerypv -h /dev/hdisk0
```

```

00000000  C9C2D4C1 00000000 00000000 00000000 |.....| 00000010
00000000 00000000 00000000 00000000 |.....| 00000020 00000000
0 00000000 00000000 00000000 |.....| 00000030 00000000 00000000
00000000 00000000 |.....| 00000040 00000000 00000000 00000000
00000000 |.....| 00000050 00000000 00000000 00000000 00000000
|.....| 00000060 00000000 00000000 00000000 00000000 |.....
.....| 00000070 00000000 00000000 00000000 00000000 |.....|
00000080  0000313A BB567908
00000000 00000000 |..1:.Vy.....| 00000090 00000000 00000000 00000000
00000000 |.....| 000000A0 00000000 00000000 00000000 00000000
|.....| 000000B0 00000000 00000000 00000000 00000000 |.....
.....| 000000C0 00000000 00000000 00000000 00000000 |.....|
000000D0 00000000 00000000 00000000 00000000 |.....| 000000E0
00000000 00000000 00000000 00000000 |.....| 000000F0 00000000
00000000 00000000 00000000 |.....|

```

What you see as marked in the output is PVID (ID of the physical volume), usually seen in lspv output. If you don't have PVID in lspv output, or you have some unexpected PVID, or you have some strange errors, trying to import a volume group, you can check the real PVID on the physical volume, using the command above.

Now we're coming to something not obvious. I think, one the most important UNIX concepts is "everything is a file". If it is so, we can use any file name instead of PV name as the parameter to lquerypv. Let's check:

```
# lquerypv -h /etc/hosts
```

```
00000000 23204942 4D5F5052 4F4C4F47 5F424547 |# IBM_PROLOG_BEG| 00000010
494E5F54 4147200A 23205468 69732069 |IN_TAG .# This i| 00000020 7320616
E 20617574 6F6D6174 6963616C |s an automatical| 00000030 6C792067 656E6572
61746564 2070726F |ly generated pro| 00000040 6C6F672E 200A2320 200A2320
626F7336 |log. .# .# bos6| 00000050 31442073 72632F62 6F732F75 73722F73
|1D src/bos/usr/s| 00000060 62696E2F 6E657473 74617274 2F686F73 |bin/netst
art/hos| 00000070 74732031 2E32200A 2320200A 23204C69 |ts 1.2 .# .# Li|
00000080 63656E73 6564204D 61746572 69616C73 |censed Materials| 00000090
202D2050 726F7065 72747920 6F662049 | - Property of I| 000000A0 424D200A
2320200A 2320434F 50595249 |BM .# .# COPYRI| 000000B0 47485420 496E7465 7
26E6174 696F6E61 |GHT Internationa| 000000C0 6C204275 73696E65 7373204D 61
636869 |l Business Machi| 000000D0 6E657320 436F7270 2E203139 38352C31 |n
es Corp. 1985,1| 000000E0 39383920 0A232041 6C6C2052 69676874 |989 .# All
Right| 000000F0 73205265 73657276 6564200A 2320200A |s Reserved .# .|
```

and it works.

Not always useful to look at the first 256 bytes of the file. How can we look at the next 256 bytes of the file? There is another undocumented option - we can specify, at which point we want to start looking at a file. But the address must be in hexadecimal form. To show the next 256 bytes we can run the following command:

```
# lquerypv -h /etc/hosts 100
```

```
00000100 23205553 20476F76 65726E6D 656E7420 |# US Government | 00000110
55736572 73205265 73747269 63746564 |Users Restricted| 00000120 2052696
7 68747320 2D205573 652C2064 | Rights - Use, d| 00000130 75706C69 63617469
6F6E206F 72200A23 |uplication or .#| 00000140 20646973 636C6F73 75726520
72657374 | disclosure rest| 00000150 72696374 65642062 79204753 41204144
|ricted by GSA AD| 00000160 50205363 68656475 6C652043 6F6E7472 |P Schedul
e Contr| 00000170 61637420 77697468 2049424D 20436F72 |act with IBM Cor|
00000180 702E200A 2320200A 23204028 23293437 |p. .# .# @(# )47| 00000190
09312E32 20207372 632F626F 732F7573 |.1.2 src/bos/us| 000001A0 722F7362
696E2F6E 65747374 6172742F |r/sbin/netstart/| 000001B0 686F7374 732C2063 6
D646E65 742C2062 |hosts, cmdnet, b| 000001C0 6F733631 442C2064 32303037 5F
343941 |os61D, d2007_49A| 000001D0 32203130 2F312F30 37203133 3A35373A |2
10/1/07 13:57:| 000001E0 35320A23 2049424D 5F50524F 4C4F475F |52.# IBM_PR
OLOG_| 000001F0 454E445F 54414720 0A23200A 2320434F |END_TAG .# .# CO|
```

So we can look at the whole file, starting every time 256 bytes further. Nice attempt, but I would like to see a little bit more, than 256 bytes. The second undocumented argument let us specify how many bytes (in hex) we would like to see. E.g., to show first 512 bytes of the file we specify 0 as a starting point and 200 as a length of the excerpt.

```
# lquerypv -h /etc/hosts 0 200
```

```
00000000 23204942 4D5F5052 4F4C4F47 5F424547 |# IBM_PROLOG_BEG| 00000010
494E5F54 4147200A 23205468 69732069 |IN_TAG .# This i| 00000020 7320616
E 20617574 6F6D6174 6963616C |s an automatical| 00000030 6C792067 656E6572
```

```

61746564 2070726F |ly generated pro| 00000040 6C6F672E 200A2320 200A2320
626F7336 |log. .# .# bos6| 00000050 31442073 72632F62 6F732F75 73722F73
|1D src/bos/usr/s| 00000060 62696E2F 6E657473 74617274 2F686F73 |bin/netst
art/hos| 00000070 74732031 2E32200A 2320200A 23204C69 |ts 1.2 .# .# Li|
00000080 63656E73 6564204D 61746572 69616C73 |censed Materials| 00000090
202D2050 726F7065 72747920 6F662049 | - Property of I| 000000A0 424D200A
2320200A 2320434F 50595249 |BM .# .# COPYRI| 000000B0 47485420 496E7465 7
26E6174 696F6E61 |GHT Internationa| 000000C0 6C204275 73696E65 7373204D 61
636869 |l Business Machi| 000000D0 6E657320 436F7270 2E203139 38352C31 |n
es Corp. 1985,1| 000000E0 39383920 0A232041 6C6C2052 69676874 |989 .# All
Right| 000000F0 73205265 73657276 6564200A 2320200A |s Reserved .# .| 00
000100 23205553 20476F76 65726E6D 656E7420 |# US Government | 00000110 5
5736572 73205265 73747269 63746564 |Users Restricted| 00000120 20526967 68
747320 2D205573 652C2064 | Rights - Use, d| 00000130 75706C69 63617469 6F6
E206F 72200A23 |uplication or .#| 00000140 20646973 636C6F73 75726520 7265
7374 | disclosure rest| 00000150 72696374 65642062 79204753 41204144 |ric
ted by GSA AD| 00000160 50205363 68656475 6C652043 6F6E7472 |P Schedule Co
ntr| 00000170 61637420 77697468 2049424D 20436F72 |act with IBM Cor| 0000
0180 702E200A 2320200A 23204028 23293437 |p. .# .# @(#)47| 00000190 093
12E32 20207372 632F626F 732F7573 |.1.2 src/bos/us| 000001A0 722F7362 696E
2F6E 65747374 6172742F |r/sbin/netstart/| 000001B0 686F7374 732C2063 6D646
E65 742C2062 |hosts, cmdnet, b| 000001C0 6F733631 442C2064 32303037 5F3439
41 |os61D, d2007_49A| 000001D0 32203130 2F312F30 37203133 3A35373A |2 10/
1/07 13:57:| 000001E0 35320A23 2049424D 5F50524F 4C4F475F |52.# IBM_PROLOG
_| 000001F0 454E445F 54414720 0A23200A 2320434F |END_TAG .# .# CO|

```

Now using this secret knowledge and KornShell capabilities, we can write a simple oneliner to dump a file in hex:

```
# hd() { [ -f "$1" ] && lquerypv -h "$1" 0 $(printf "%0x" $(ls -l "$1" 2>/dev/
null | awk '{print $5}')) || print -u2 "File $1 not found" ; }
```

```
# hd /etc/hosts
```

```

00000000 23204942 4D5F5052 4F4C4F47 5F424547 |# IBM_PROLOG_BEG| 00000010
494E5F54 4147200A 23205468 69732069 |IN_TAG .# This i| 00000020 7320616
E 20617574 6F6D6174 6963616C |s an automatical| 00000030 6C792067 656E6572
61746564 2070726F |ly generated pro| ...

```

P.S. If you don't like to write scripts and use undocumented commands, there is a documented hex viewer in AIX - od -cx.