

libosmcore - Bug #71

Log target of everything is broken

02/19/2016 10:47 PM - zecke2

Status: Resolved	Start date:
Priority: High	Due date:
Assignee: neels	% Done: 100%
Category:	
Target version:	
Spec Reference:	
Description Setting logging level all everything does not show debug messages. There must be an issue inside the osmo_vlogp function. This is in libosmcore.	
Related issues: Related to libosmcore - Bug #2836: log filter doesn't work for stderr when g... Resolved 01/18/2018 Related to libosmcore - Bug #3409: log level all is broken Resolved 07/20/2018	

History

#1 - 02/21/2016 04:46 PM - laforge

- Project changed from OpenBSC to libosmcore
- Category deleted (OpenBSC)

#2 - 11/09/2016 10:20 AM - laforge

- Assignee deleted (laforge)

#3 - 01/12/2017 01:20 PM - laforge

- Assignee set to msuraev

#4 - 01/13/2017 12:52 PM - msuraev

- Status changed from New to In Progress

Tested Gerrit # 1580 using following logging configuration:

```
log stderr
logging color 1
logging timestamp 0
logging level all everything
```

When running without debug mask DEBUGP messages are not printed.
When running with debug mask (.. -d DOML) then corresponding DEBUGP are printed.

It's the same if I use "logging level oml everything" instead.

Have I understood it correctly that with this configuration DEBUGP should be printed even without specifying the mask in command-line?

#5 - 01/17/2017 03:02 PM - msuraev

- Status changed from *In Progress* to *Stalled*

#6 - 02/07/2017 04:31 PM - msuraev

- % Done changed from 0 to 50

Fix is available in gerrit 1582.

#7 - 02/21/2017 10:58 AM - neels

I think it would be best to deprecate the 'logging level all everything' and fix it by adding a different command, to avoid situations where users have this in their conf file, not being aware of it, and suddenly the log explodes because the meaning of it changed (even though it changed to actually doing what it says). I happen to have it in my config files in my test setups, no idea why, and would need to adjust that "out of the blue". So if we really need this functionality, let's make it a different command and print a "% this command no longer exists" for the old one?

#8 - 02/23/2017 02:43 PM - laforge

see <https://gerrit.osmocom.org/#/c/1582/> for current patches related to this.

#9 - 02/23/2017 02:45 PM - laforge

msuraev wrote:

Have I understood it correctly that with this configuration DEBUGP should be printed even without specifying the mask in command-line?

the log mask on the command line is nothing but a legacy way of configuring the stderr logging via vty/config file. I wasn't even aware somebody still uses it and I would remove the command line options altogether. However, some people seem to prefer that way.

So whether you specify something as command line log mask or as "log stderr" in the config file should never make any difference.

#10 - 02/23/2017 02:53 PM - laforge

neels wrote:

I think it would be best to deprecate the 'logging level all everything' and fix it by adding a different command, to avoid situations where users have this in their conf file, not being aware of it, and suddenly the log explodes because the meaning of it changed (even though it changed to actually doing what it says). I happen to have it in my config files in my test setups, no idea why, and would need to adjust that "out of the blue". So if we really need this functionality, let's make it a different command and print a "% this command no longer exists" for the old one?

Well, it has its advantages and disadvantages. We shouldn't have to apologize to users if we fix bugs, even if the bugs have existed for years. Do you have a good proposal as to what this new command should be called?

#11 - 02/24/2017 01:48 PM - neels

BTW, a current workaround to get everything logged is simply

```
logging level all debug
```

#12 - 02/27/2017 11:53 AM - msuraev

to summarize, I can do following:

- make 'everything' alias to debug and print deprecation warning if it's used in the config
- make 'all' act as globing which sets default log level for any category
- double-check that any subsequent per-category log level overrides it
- double-check that it works regardless of cli parameters

Am I missing anything?

#13 - 07/06/2017 03:25 PM - msuraev

Gerrit 3148 has been sent for review. If you think that deprecating "everything" is the right way than +2 it, otherwise I'll try to fix gerrit 1582 to make "everything" alias to "debug".

#14 - 07/11/2017 04:41 PM - msuraev

- Status changed from Stalled to In Progress

Gerrit 3148 has been merged.

#15 - 07/21/2017 02:42 PM - msuraev

- Status changed from In Progress to Stalled

#16 - 10/16/2017 12:14 PM - msuraev

It's been reported on ML that "logging level all everything" somehow makes it possible to skip logging for certain subsystems while "logging level all debug" doesn't. Need more info to reproduce this properly.

#17 - 10/17/2017 02:13 AM - keith

My analysis posted to the ML: (simplified)

Using log facilities MNCC and DCC, connecting and disconnecting osmo-sip-connector will log messages at NOTICE on both those facilities.

I'm bringing up a fresh nitb with empty config file and database. I will use some logging config commands and then connect and disconnect the MNCC socket.

```
OpenBSC> enable
OpenBSC# logging enable
OpenBSC# logging level mncc fatal
OpenBSC# logging level cc fatal
```

```
OpenBSC# logging level all debug
OpenBSC# logging filter all 1
OpenBSC# <0006> mncc_sock.c:271 MNCC Socket has connection
with external call control application
<0006> mncc_sock.c:84 MNCC Socket has LOST connection
<0001> gsm_04_08.c:469 Clearing all currently active
transactions!!!
```

Here the specific per-subsystem levels were not respected, rather they were over-ridden by the logging all directive.

Change logging all to "everything":

```
OpenBSC# logging level all everything
```

Change Logging MNCC to info:

```
OpenBSC# logging level mncc info
```

```
OpenBSC# <0006> mncc_sock.c:271 MNCC Socket has connection
with external call control application
<0006> mncc_sock.c:84 MNCC Socket has LOST connection
```

Now the subsystem levels were respected, as CC was level FATAL and MNCC was at level INFO

This is my desired behaviour.

```
OpenBSC# logging level cc info
OpenBSC# logging level mncc fatal
OpenBSC# <0001> gsm_04_08.c:469 Clearing all currently
active transactions!!!
```

Reversed. Levels still respected, good.

```
OpenBSC# logging level all debug
OpenBSC# <0006> mncc_sock.c:271 MNCC Socket has connection
with external call control application
<0006> mncc_sock.c:84 MNCC Socket has LOST connection
<0001> gsm_04_08.c:469 Clearing all currently active
transactions!!!
```

Again, Now **everything** is logging at DEBUG

Just to really really drive the point home:

```
OpenBSC# logging level all notice
OpenBSC# logging level cc fatal
OpenBSC# logging level mncc fatal
OpenBSC# <0006> mncc_sock.c:271 MNCC Socket has connection
with external call control application
<0006> mncc_sock.c:84 MNCC Socket has LOST connection
<0001> gsm_04_08.c:469 Clearing all currently active
transactions!!!
```

It is impossible to silence a subsystem when logging level all [anything other than everything] has been issued.

I am only able to achieve this with logging level all everything - This is the key to respecting the per-subsystem levels.

It is impossible to have:

```
logging level X on Facility 1
and
logging level Y on facility 2
unless
```

logging level all everything has been issued.

(otherwise one gets level Z on all facilities after logging level all Z

#18 - 10/17/2017 02:35 AM - keith

msuraev wrote:

Need more info to reproduce this properly.

Do you somehow observe different behaviour to what I describe?

With current libosmocore,
do you know a sequence of logging vty commands that will display SMPP and only SMPP at level DEBUG?

#19 - 10/17/2017 02:44 AM - keith

I have sent <https://gerrit.osmocom.org/#/c/4313/> for review,
This is the patch I am using currently to get the behaviour I desire.

#20 - 10/17/2017 10:08 AM - msuraev

Frankly, I think having "special" categories ("all" vs "cc", "smpp" etc) and levels ("everything" vs "notice", "error" etc) is a counter-intuitive hack. Which is undocumented as well. We should use separate, explicit commands for things like setting default log level for categories. For example "logging default-level notice". I'll follow-up with patches implementing it once I have time - don't hesitate to submit your version though. Or propose entirely different ideas.

#21 - 10/17/2017 12:00 PM - laforge

On Tue, Oct 17, 2017 at 10:08:20AM +0000, msuraev [REDMINE] wrote:

Frankly, I think having "special" categories ("all" vs "cc", "smpp" etc) and levels ("everything" vs "notice", "error" etc) is a counter-intuitive hack.

ACK.

Which is undocumented as well.

That's unfortunate.

We should use separate, explicit commands for things like setting default log level for categories. For example "logging default-level notice".

if we want such a feature at all, I agree.

#22 - 10/17/2017 01:44 PM - keith

I think the feature of seeing facility X at level debug while setting facility Y at level fatal is a good feature to have.

For the immediate, if it does not step on anybody's toes, I would appreciate gerrit #4313 as it allows me to see the logging i need at any given time from the VTY.

#23 - 10/17/2017 03:38 PM - laforge

On Tue, Oct 17, 2017 at 01:44:05PM +0000, keith [REDMINE] wrote:

Issue [#71](#) has been updated by keith.

I think the feature of seeing facility X at level debug while setting facility Y at level fatal is a good feature to have.

This is what we always had, and what we will always have. But this doesn't involve any "logging level all everything" commands. I've never used it in my close to 10 years of OpenBSC work, and I always have various different log levels for different sub-systems. You don't need "everything", nor do you need "all" for that. In fact, I would argue (like Max has recently argued, if I understand him correctly), that those additional "features" are very hard to understand, not documented and conceptually wrong, as "all" is not a sub-system/category (contrary to "mm" or "rsI") and "everything" is not a log level (contrary to "debug").

For the immediate, if it does not step on anybody's toes, I would appreciate gerrit #4313 as it allows me to see the logging i need at any given time from the VTY.

I still don't get it, sorry :/

#24 - 10/17/2017 05:42 PM - keith

laforge wrote:

I've never used it in my close to 10 years of OpenBSC work,

Well I can't argue with that.

But..

If you connect to a vty, enable logging and issue show logging vty, the first line you will see is

Global Loglevel: EVERYTHING

and I always have various different log levels for different sub-systems. You don't need "everything", nor do you need "all" for that.

I think in fact that you do "need" everything, as it is the default state.

I had a habit of issue a log level all debug command and you are saying you never use it.

So I see now where the confusion comes from.

I think what happens is that once you issue **log level all debug**, you cannot get *BACK* to the initial state where individual facility log levels are respected (let me call it "EVERYTHING") without being all to issue **logging level all everything**

#25 - 10/17/2017 07:12 PM - laforge

On Tue, Oct 17, 2017 at 05:42:24PM +0000, keith [REDMINE] wrote:

I think in fact that you do "need" everything, as it is the default state.

I had a habit of issue a log level all debug command and you are saying you never use it.

So I see now where the confusion comes from.

Thanks for explaining.

I think what happens is that once you issue **log level all debug**, you cannot get *BACK* to the initial state where individual facility log levels are respected (let me call it "EVERYTHING") without being all to issue **logging level all everything**

So if you're saying the problem is that you cannot get back to "everything" after using "logging level all \$FOO", then the logical consequence to me would be to remove "logging level all", and/or simply ignore (with a nice notice) any "logging level all" command entered. At that point, the "everything" behavior would be persistent? Would that work for you?

I think the notion of "logging level all ..." is hard to understand to

begin with. I have not seen such a "global level restriction that overrides all the per-subsystem restrictions" in any other logging framework. And if it exists, I believe it shouldn't be "logging level " like we had so far. The "logging level" configures a per-subsystem log-level. Having a global "mask" or "filter" should have a different syntax to set it apart as it's not orthogonal what the other (real) "logging level" commands do.

Yes, I know, it sucks to change things so many years later, but then I also don't think we should stick with tradition if there is a general feeling that the historic approach is confusing and not well understood, even within the "core community" of the project.

#26 - 10/21/2017 06:27 PM - laforge

- Assignee changed from msuraev to laforge

#27 - 11/22/2017 01:32 PM - pespin

Running latest master today, debugging some stuff I added a "log level all debug" to my config file to get all possible output into stderr:

```
log stderr
  logging color 1
  logging print extended-timestamp 1
  logging print category 1
  logging level all debug
```

After a while of not understanding why I was getting some messages but not all of them, I entered with telnet and checked "show running-config":

```
log stderr
  logging filter all 1
  logging color 1
  logging print category 1
  logging print extended-timestamp 1
  logging level all debug
  logging level rsl info
  logging level oml info
  logging level rll notice
  logging level rr notice
  logging level meas notice
  logging level pag info
  logging level llc info
  logging level llp info
  logging level dsp debug
  logging level pcu notice
  logging level ho notice
  logging level trx notice
  logging level loop notice
  logging level abis notice
  logging level rtp notice
  logging level sum notice
  logging level lglobal notice
  logging level llapd notice
  logging level linp notice
  logging level lmux notice
  logging level lmi notice
  logging level lmib notice
```



```
logging level lsms notice
logging level lctrl notice
logging level lgtp notice
logging level lstats notice
logging level lgsup notice
logging level loap notice
logging level lss7 notice
logging level lsccp notice
logging level lsua notice
logging level lm3ua notice
logging level lmgcp notice
```

So, it seems my all debug is overridden automatically in lots of categories with "notice" level after my "all debug" line. Then I get into an scenario in which the following is printed:

```
LOGP(DPCU, LOGL_INFO, "Sending RACH indication: qta=%d, ra=%d, "
"fn=%d\n", qta, ra, fn);
```

But not this one (immediately before in the code calling the previous message):

```
LOGP(DL1P, LOGL_INFO, "RACH for packet access (toa=%d, ra=%d)\n",
rach_ind->acc_delay, rach_ind->ra);
```

Which is really strange, because according to log levels it should be the opposite, like the first one not being printed (pcu cfg is NOTICE) and the second one being printed (l1p cfg is INFO).

If instead of using "logging level all debug", I take the output from the show-running-config and I add them to my cfg file modifying the lines to have them all with "debug" level, then everything works fine (and both messages are printed).

#28 - 01/18/2018 12:51 PM - msuraev

- Related to Bug #2836: log filter doesn't work for stderr when gsmtap is configured added

#29 - 08/08/2018 09:46 AM - laforge

- Assignee changed from laforge to pespun

#30 - 08/28/2018 05:17 PM - pespun

- Related to Bug #3409: log level all is broken added

#31 - 08/28/2018 05:18 PM - pespun

- Status changed from Stalled to Feedback

See patches submitted in gerrit (you can find links in [#3409](#)).

#32 - 09/12/2018 12:10 PM - pespun

- Assignee changed from pespun to neels

#33 - 09/12/2018 01:26 PM - neels

- Status changed from Feedback to In Progress

- % Done changed from 50 to 90

A related set of patches is up on gerrit, after the discussion on openbsc@ has seemingly concluded.

<http://lists.osmocom.org/pipermail/openbsc/2018-September/012178.html>

<https://gerrit.osmocom.org/10883>

<https://gerrit.osmocom.org/10884>

<https://gerrit.osmocom.org/10885>

<https://gerrit.osmocom.org/10886>

<https://gerrit.osmocom.org/10887>

<https://gerrit.osmocom.org/10888>

<https://gerrit.osmocom.org/10889>

<https://gerrit.osmocom.org/10890>

#34 - 09/30/2018 08:38 AM - laforge

[neels](#), what's missing here?

#35 - 10/02/2018 02:09 PM - neels

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

To summarize, 'everything' was either a misnomer or an implementation slip up. Either way it always meant "switch off a global logging clamp" that was before set by 'logging level all LEVEL'.

Now we have 'logging level force-all LEVEL' and 'no logging level force-all' as less confusing names for the same functionality.

The old 'logging level all LEVEL' and 'logging level all everything' are deprecated aliases for the same, i.e. they now again work exactly like they always did.

All patches merged.