OsmoTRX - Bug #3250

VTY: SIGABRT: Assert failed 0 trx_vty.c:506

05/09/2018 08:06 AM - fixeria

Status:	Resolved	**Start date:**	05/09/2018
Priority:	Normal	**Due date:**	
Assignee:	fixeria	**% Done:**	100%
Category:			
Target version:			
Spec Reference:			

Description

It seems that not all VTY configuration nodes are registered correctly.

```
$ telnet localhost 4237
OsmoTRX> en
OsmoTRX> configure terminal
OsmoTRX(config)# ctrl
OsmoTRX(config-ctrl)# exit or end
```

Result:

```
-- Transceiver active with 2 channel(s)

...

Assert failed 0 trx_vty.c:506
backtrace() returned 10 addresses
osmo-trx-uhd() [0x43785a]
/usr/local/lib/libosmovty.so.4(vty_go_parent+0x81) [0x7fe36b2227e1]
/usr/local/lib/libosmovty.so.4(+0xce3b) [0x7fe36b223e3b]
/usr/local/lib/libosmovty.so.4(vty_read+0x496) [0x7fe36b224af6]
/usr/local/lib/libosmovty.so.4(+0xfa89) [0x7fe36b226a89]
/usr/local/lib/libosmocore.so.10(osmo_select_main+0x18f) [0x7fe36affb1ef]
osmo-trx-uhd() [0x407fe7]
/lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xf5) [0x7fe3696dbf45]
osmo-trx-uhd() [0x40a8e9]
signal 6 received

...

Stopped reason: SIGABRT
0x00007ffff5657c37 in __GI_raise (sig=sig@entry=0x6) at ../nptl/sysdeps/unix/sysv/linux/raise.c:56
56      ../nptl/sysdeps/unix/sysv/linux/raise.c: No such file or directory.

gdb-peda$ bt
#0  0x00007ffff5657c37 in __GI_raise (sig=sig@entry=0x6) at ../nptl/sysdeps/unix/sysv/linux/raise.
c:56
#1  0x00007ffff565b028 in __GI_abort () at abort.c:89
#2  0x000000000043785f in trx_vty_go_parent (vty=<optimized out>) at trx_vty.c:506
#3  0x00007ffff71897e1 in vty_go_parent (vty=0x1059ea0) at command.c:2147
#4  0x00007ffff71898c2 in config_exit (self=<optimized out>, vty=<optimized out>, argc=<optimized
out>, argv=<optimized out>) at command.c:2632
#5  0x00007ffff71871c7 in cmd_execute_command_real (vline=vline@entry=0x105b6b0, vty=vty@entry=0x1
059ea0, cmd=cmd@entry=0x0) at command.c:2275
#6  0x00007ffff7189a34 in cmd_execute_command (vline=vline@entry=0x105b6b0, vty=vty@entry=0x1059ea
0, cmd=cmd@entry=0x0, vtysh=vtysh@entry=0x0)
    at command.c:2308
#7  0x00007ffff718bc19 in vty_command (buf=<optimized out>, vty=0x1059ea0) at vty.c:420
#8  vty_execute (vty=0x1059ea0) at vty.c:684
#9  vty_read (vty=<optimized out>) at vty.c:1426
#10 0x00007ffff718da89 in client_data (fd=0x1059df8, what=0x1) at telnet_interface.c:135
#11 0x00007ffff6f621ef in osmo_fd_disp_fds (_eset=0x7fffffffe030, _wset=0x7fffffffdfb0, _rset=0x7f
```

```
fffffdf30) at select.c:216
#12 osmo_select_main (polling=0x0) at select.c:256
#13 0x0000000000407fe7 in main (argc=argc@entry=0x3, argv=argv@entry=0x7fffffffe5a8) at osmo-trx.c
pp:568
#14 0x00007ffff5642f45 in __libc_start_main (main=0x406eb0 <main(int, char**)>, argc=0x3, argv=0x7
fffffffe5a8, init=<optimized out>,
    fini=<optimized out>, rtld_fini=<optimized out>, stack_end=0x7fffffffe598) at libc-start.c:287
#15 0x000000000040a8e9 in _start ()
```

trx_vty_go_parent():

```
static int trx_vty_go_parent(struct vty *vty)
{
        switch (vty->node) {
        case TRX_NODE:
                vty->node = CONFIG_NODE;
                vty->index = NULL;
                vty->index_sub = NULL;
                break;
        case CHAN_NODE:
                vty->node = TRX_NODE;
                vty->index = NULL;
                vty->index_sub = NULL;
                break;
        default:
                OSMO_ASSERT(0);
        }

        return vty->node;
}
```

History

#1 - 05/09/2018 08:59 AM - fixeria

- Status changed from New to Feedback

- % Done changed from 40 to 90

See https://gerrit.osmocom.org/8083

#2 - 05/09/2018 09:05 AM - laforge

thanks.

btw: It would be great to have a VTY unit test for those kind of bugs, see e.g. osmo-bsc/tests/vty_test_runner.py

In general, not all osmocom components have related testing yet, and it's a good way to ensure various VTY features work and continue to work.

```
> static int trx_vty_go_parent(struct vty *vty)
> {
>         switch (vty->node) {
>         case TRX_NODE:
>                 vty->node = CONFIG_NODE;
>                 vty->index = NULL;
>                 vty->index_sub = NULL;
>                 break;
>         case CHAN_NODE:
>                 vty->node = TRX_NODE;
>                 vty->index = NULL;
>                 vty->index_sub = NULL;
>                 break;
>         default:
>                 OSMO_ASSERT(0);
>         }
>
>         return vty->node;
> }
>
```

I think this assert is wrong, as there are other nodes provided by the library itself.

neels is probably the person who has worked most with the VTY code in recent years

#3 - 05/09/2018 12:23 PM - neels

the vty_go_parent() gets called for all nodes, not only for those that are actively defined by the caller.
The VIEW_NODE and ENABLE_NODE don't get passed to the go_parent_cb, but as you see the CTRL_NODE defined in libosmocore does.
So the OSMO_ASSERT(0) is wrong.

TLDR: we can just do this in trx_vty_go_parent() and it should all work out fine:

```
vty->index = NULL;
vty->index_sub = NULL;
return 0;
```

Explained: the vty_go_parent() was once responsible to accurately reflect the tree structure of node IDs upon node exit.
But I changed that some time ago, so that we record the parent ancestry, and now automatically go back to the parent node, overriding whatever go_parent does.
For some gritty details however there still needs to be **some** go-parent defined, a patch to make that unnecessary is still waiting somewhere on a branch.
The idea now is that the go_parent is optional: if you want some action to take place upon exiting a node,
like triggering re-init of some component based on the new values, you can do that in the go_parent_cb,
but you can also just do nothing. At the moment it is still necessary to have a function defined though, even if it does nothing.
vty_go_parent() completely ignores whatever node id the trx_vty_go_parent() has returned and will go to the correct parent.

The vty->index is not automatically set to what the parent had yet, that is probably also a patch still waiting on a branch somewhere.
So if vty->index should point to a specific object, that still is the responsibility of the go_parent_cb.
(As soon as we do that implicitly, almost all go_parent_cb around anywhere become obsolete, except the ones that trigger some init action.)

#4 - 05/09/2018 12:58 PM - fixeria

- Status changed from Feedback to Resolved

- % Done changed from 90 to 100

Merged.