Forum answers on libnetconf

Sunday, September 25, 2016 7:34 PM

What all functionalities should be implemented in a application to make it NETCONF compliance ?

From <https://groups.google.com/forum/#!topic/libnetconf/vOgC56XgfGo>

to find exact answers you should look into the relevant RFCs (mainly RFC 6241). For example, the startup or candidate datastore are capabilities, which do not have to be supported. Also, for any standard RPC defined in the NETCONF RFC that you will not support I suggest replying with the operation-not-supported reply. Finally, all I can understand when saying something is "YANG compliant" is that it uses valid YANG modules to describe datastores. YANG is simply a modeling language, not a protocol.

Answer to above question by the author of libnetconf is as below:

simple answer is no, your application will not be NETCONF compliant. Missing startup and candidate is not the problem. And libnetconf actually allows the server to set what capabilities (and the startu and candidate datastores are actually capabilities) are supported. This way you can switch off all of the capabilities. However, the NETCONF implementation must support the base NETCONF functionali which includes those base rpcs such as get-config or edit-config. So if you will not support these, the implementation will not be NETCONF compliant.

One note to your architecture. The common way how to connect libnetconf server (e.g. netopeer-server) with an application is a transAPI module. However, there is another way - the custom datastore implementation. libnetconf supports 3 types of datastores: 1) file - the common type you know from netopeer, data are stored in XML files, ***2) empty - for data models without configuration data (only status), it is used for some internal data models such as ietf-netconf-monitoring; and*** *3) custom - server provides the set of callback functions that actually implements the datastore and these function interconnect the NETCONF server with an existing database or any other data store. But such NETCONF datastore implementation still should implement e.g. edit config and other base NETCONF operations. I know that CZ.NIC uses this way in their Turris project [1] (they maps NETCONF to OpenWRT's UCI) and I'm actually implementing this in OF-CONFIG implementation for Open vSwitch [2]. I believe that this way of integrating NETCONF with a current system would fit to your architecture, try to think about it.*

From <https://groups.google.com/forum/#!topic/libnetconf/vOgC56XgfGo>

How to fill running data-store with transapi_init()?

From <https://groups.google.com/forum/#!topic/libnetconf/jB-8INBPs8c>

this is how it works. There are three datastores in Netopeer, running, startup and candidate, we don't care about the last one now. Let's say a transAPI module is being initialized. You start with the running config (presumably reflecting the current state of the system, if nothing changed since terminating the server). Then you are given the chance to check whether this really is the case, if there weren't some changes, in transapi_init() function. There you're supposed to detect the current state of the system an build the corresponding XML configuration. After that libnetconf considers the running configuration to match the one on the device. So, it finally copies the startup configuration to the running and calls the appropriate callbacks. This way it should be possible to fully control the module start and not call callbacks on invalid changes. I ho this made it clear for you.

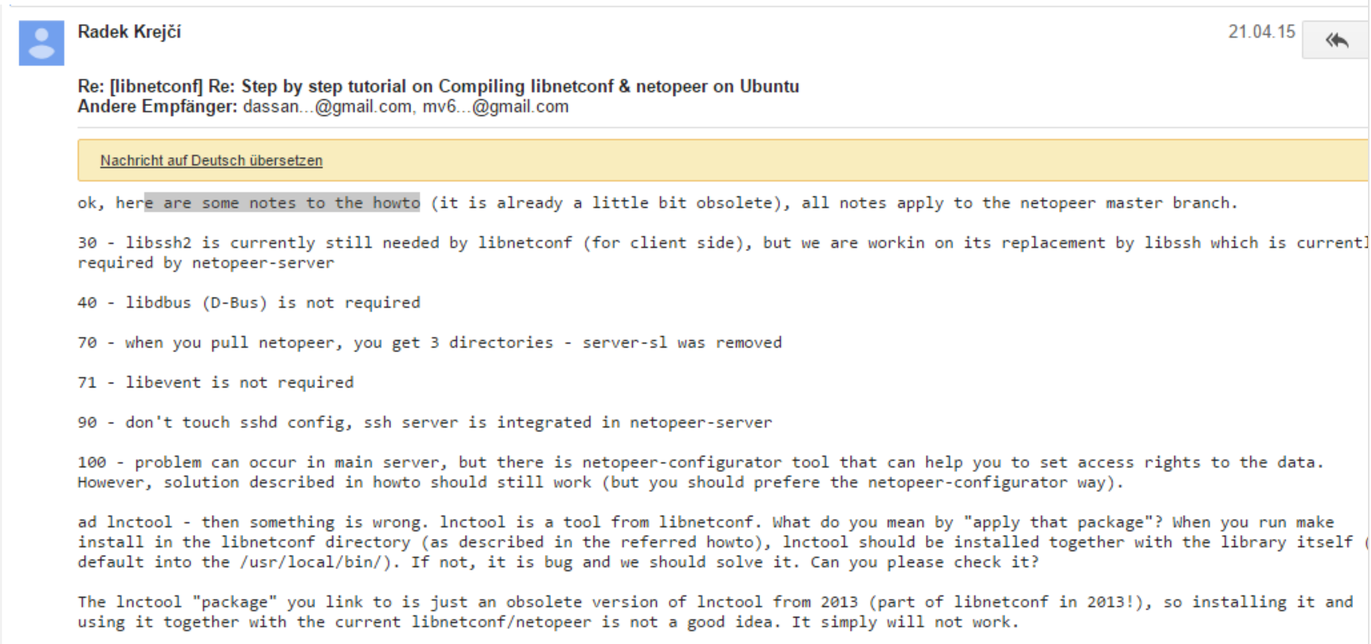
Step by step tutorial on Compiling libnetconf & netopeer on Ubuntu

From <https://groups.google.com/forum/#!searchin/libnetconf/libxml2/libnetconf/sVaj8tYYZC0/U3aN6h3SgfoJ>

how to generate code using lnctool ?

From <https://groups.google.com/forum/#!msg/libnetconf/wvK_39B8-w8/U9JHCmAmKkUJ>

Additionally below link might come in handy as well:
https://wiki.opendaylight.org/view/OpenDaylight_Controller:Config:Examples:Netconf:Manual_netopeer_installation



Radek Krejčí 21.04.15

Re: [libnetconf] Re: Step by step tutorial on Compiling libnetconf & netopeer on Ubuntu
Andere Empfänger: dassan...@gmail.com, mv6...@gmail.com

ok, here are some notes to the howto (it is already a little bit obsolete), all notes apply to the netopeer master branch.

30 - libssh2 is currently still needed by libnetconf (for client side), but we are workin on its replacement by libssh which is currentl required by netopeer-server

40 - libdbus (D-Bus) is not required

70 - when you pull netopeer, you get 3 directories - server-sl was removed

71 - libevent is not required

90 - don't touch sshd config, ssh server is integrated in netopeer-server

100 - problem can occur in main server, but there is netopeer-configurator tool that can help you to set access rights to the data. However, solution described in howto should still work (but you should prefere the netopeer-configurator way).

ad lnctool - then something is wrong. lnctool is a tool from libnetconf. What do you mean by "apply that package"? When you run make install in the libnetconf directory (as described in the referred howto), lnctool should be installed together with the library itself (default into the /usr/local/bin/). If not, it is bug and we should solve it. Can you please check it?

The lnctool "package" you link to is just an obsolete version of lnctool from 2013 (part of libnetconf in 2013!), so installing it and using it together with the current libnetconf/netopeer is not a good idea. It simply will not work.