NVIDIA DOCA Comm Channel Admin Tool
# Table of contents

- Introduction
- Prerequisites
- Description and Execution
  - Sample Output from BlueField Arm
  - Sample Output from x86
This document provides instructions on the usage of the DOCA Comm Channel Admin Tool.

**Introduction**

The Comm Channel Admin Tool is used to print a snapshot of DOCA Comch (comm channel) connections:

- On the BlueField Arm side, it includes DOCA Comch servers and their current connection information
- On the host side, it includes all active client connections and the server they are connected to
- Only client-to-server control channels are reported; fast path producer/consumer channels are not.

**Prerequisites**

The Comm Channel Admin Tool is for Linux only and requires an up-to-date BFB bundle or DOCA host packages of at least 2.7, which include in the Resource dump binary.

**Description and Execution**

The Comm Channel Admin Tool can be executed on the host or Arm CPUs. By default, the tool scans all available PCIe slots to detect supported DOCA devices and reports any Comch information available.

The tool can be run on BlueField Arm or x86 host using the following command:

```
/opt/mellanox/doca/tools/doca_comm_channel_admin
```

**Sample Output from BlueField Arm**

On the BlueField Arm side, any active DOCA Comch servers are be reported:
The following information is available:

- **Server Name** – the name assigned to the server
- **PID** – the Linux process ID of the application which created the server
- **Connections** – the number of connections active on the server out of the total allowed (e.g., 2/512 means 2 active connections of a maximum of 512)
- **PCle** – the PCIe address of the device which the server has been detected on
- **Interface Name** – the interface name associated with the PCIe address

**Note**

Connections may also be displayed on the BlueField Arm like on x86. This occurs if SF ports are detected here. The interface name associated with the PCIe address indicates the SF port.

### Sample Output from x86
The x86 host cannot run DOCA Comch servers. Therefore, individual client connections are reported:

<table>
<thead>
<tr>
<th>Server Name</th>
<th>PID</th>
<th>PCIe</th>
<th>Interface Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>comch6</td>
<td>299693</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch3</td>
<td>299688</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch2</td>
<td>299687</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch5</td>
<td>299689</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch5</td>
<td>299692</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch7</td>
<td>299696</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch5</td>
<td>299690</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch6</td>
<td>299694</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch8</td>
<td>299697</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch6</td>
<td>299695</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch8</td>
<td>299698</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch1</td>
<td>299686</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch5</td>
<td>299691</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
<tr>
<td>comch1</td>
<td>299685</td>
<td>0000:3b:00.0</td>
<td>ens1f0np0</td>
</tr>
</tbody>
</table>

The following information is available:

- **Server Name** – the name of the BlueField Arm server that a client has connected to
- **PID** – the Linux process ID of the application running a DOCA Comch client
- **PCIe** – the PCIe address of the BlueField networking platform which the destination server is running on
- **Interface Name** – the interface name associated with the PCIe address