Chelsio Communications

Accelerate

Chelsio Unified Wire for Windows

Installation and User's Guide

Chelsio Communications (Headquarters)
209 North Fair Oaks Avenue,
Sunnyvale, CA 94085
U.S.A

www.chelsio.com

Tel: 408.962.3600
Fax: 408.962.3661

Chelsio (India) Private Limited
Subramanya Arcade, Floor 3, Tower B
No. 12, Bannerghatta Road,
Bangalore-560029
Karnataka,
India

Tel: +1-91-80-4039-6800

Chelsio KK (Japan)
Yamato Building 8F,
5-27-3 Sendagaya,
Shibuya-ku,
Tokyo 151-0051,
Japan

Sales
For all sales inquiries please send email to sales@chelsio.com

Support
For all support related questions please send email to support@chelsio.com

Document History

Version	Revision Date
1.3.6	11/19/2015
1.3.7	12/11/2015
1.3.8	01/28/2016
1.3.9	03/04/2016
1.4.0	04/21/2016
1.4.1	05/13/2016
1.4.2	06/27/2016
1.4.3	07/25/2016
1.4.4	07/29/2016
1.4.5	08/16/2016
1.4.6	08/26/2016
1.4.7	10/14/2016
1.4.8	11/11/2016
1.4.9	11/22/2016
1.5.0	12/05/2016
1.5.1	12/30/2016
1.5.2	01/05/2017
1.5.3	02/07/2017
1.5.4	02/24/2017
1.5.5	03/10/2017
1.5.6	03/31/2017
1.5.7	04/11/2017
1.5.8	05/09/2017
1.5.9	05/29/2017
1.6.0	06/26/2017
1.6.1	06/30/2017
1.6.2	08/18/2017
1.6.3	08/24/2017
1.6.4	09/28/2017
1.6.5	11/27/2017
1.6.6	12/04/2017
1.6.7	12/08/2017
1.6.8	12/27/2017

TABLE OF CONTENTS

I. Chelsio Unified Wire

1. Introduction

Thank you for choosing Chelsio Unified Wire adapters. These high speed, single chip, single firmware cards provide enterprises and data centers with high performance solutions for various Network and Storage related requirements.

The **Terminator** series is Chelsio's next generation of highly integrated, hyper-virtualized 10/25/40/50/100GbE controllers. The adapters are built around a programmable protocol-processing engine, with full offload of a complete Unified Wire solution comprising NIC, TOE, iWARP RDMA, iSCSI, FCoE and NAT support. It scales to true 100Gb line rate operation from a single TCP connection to thousands of connections, and allows simultaneous low latency and high bandwidth operation thanks to multiple physical channels through the ASIC.

Ideal for all data, storage and high-performance clustering applications, the Unified wire adapters enable a unified fabric over a single wire by simultaneously running all unmodified IP sockets, Fibre Channel and InfiniBand applications over Ethernet at line rate.

Designed for deployment in virtualized data centers, cloud service installations and high-performance computing environments, Chelsio Terminator adapters bring a new level of performance metrics and functional capabilities to the computer networking industry.

1.1. Features

Chelsio Unified Wire for Windows is an easy to use utility developed to provide installation of 64-bit Windows based drivers and tools for Chelsio's Unified Wire adapters.

It consists of the following components:

- NDIS Function driver (NIC)
- NVGRE Offload
- VXLAN Task Offload
- PacketDirect (Kernel Mode and ARM Moderation)
- SMB Direct
- Ring Backbone Utility
- RDMA/NVGRE concurrent (Mode 2)
- iWARP (ND)
- NDIS SR-IOV
- Guest RDMA (Mode 3)
- VMMQ/vRSS
- iSCSI Storport Miniport
- iSER Initiator
- Data Center Bridging (DCB)
- Storage Replica (SR)

1.2. Hardware Requirements

The Chelsio Unified Wire supports all x64 architectures supporting PCIE (x4, x8, x16) slots.

- AMD CPUs, 64-bit (x86_64/amd64)
- Intel CPUs, 64-bit (x86_64)

Note *The Chelsio Unified Wire supports 3.3v PCI bus only. Running an adapter on a PCI x4 slot is not recommended as performance will be significantly reduced by the limitations of PCI.*

Note *T4 family of adapters are not supported on Windows 2016 Server & 10 AU Client versions.*

1.3. Software Requirements

The Chelsio Unified Wire software has been developed to run on Windows based platforms. To know more about the complete list of versions supported by each driver/software, please refer their respective sections.

Note *Drivers are WHQL certified on Server 2016, 10 AU Client and Server 2012 R2 versions.*

1.4. Package Contents

Unified Wire Installer

Chelsio Unified Wire comes with an interactive installer and support documentation. The documentation, consisting of README, Release Notes and User's Guide (this document), can be found in the *<system_drive>\ChelsioUwire\docs* directory after installing Chelsio Unified Wire.

Zip Package

The zip package contains driver files, firmware binaries, adapter configuration binaries and support documentation (*docs* folder).

2. Hardware Installation

i. Shutdown/power off your system.
ii. Power off all remaining peripherals attached to your system.
iii. Unpack the Chelsio adapter and place it on an anti-static surface.
iv. Remove the system case cover according to the system manufacturer's instructions.
v. Remove the PCI filler plate from the slot where you will install the Ethernet adapter.
vi. For maximum performance, it is highly recommended to install the adapter into a PCIE x8/x16 slot.

Note *All 4-ports of T6425-CR adapter will be functional only if PCIe x8 -> 2x PCIe x4 slot bifurcation is supported by the system and enabled in BIOS. Otherwise, only 2-ports will be functional.*

vii. Holding the Chelsio adapter by the edges, align the edge connector with the PCI connector on the motherboard. Apply even pressure on both edges until the card is firmly seated. It may be necessary to remove the SFP (transceiver) modules prior to inserting the adapter.
viii. Secure the Chelsio adapter with a screw, or other securing mechanism, as described by the system manufacturer's instructions. Replace the case cover.
ix. After securing the card, ensure that the card is still fully seated in the PCIE x8/x16 slot as sometimes the process of securing the card causes the card to become unseated.
x. Connect a fiber/twinax cable, multi-mode for short range (SR) optics or single-mode for long range (LR) optics, to the Ethernet adapter or regular Ethernet cable for the 1Gb Ethernet adapter.
xi. Power on your system.
xii. Verify if the adapter was installed successfully. To do so, open **Device Manager** in **Control Panel**.
xiii. Under **Other devices** section, Chelsio adapter should be listed as ***Ethernet Controller.*** If the adapter is not listed, right-click on the system name or click on the **Actions** menu and select ***Scan for hardware changes***

For Chelsio adapters, the physical functions are currently assigned as:

- Physical functions 0 - 3: for the SR-IOV functions
- Physical function 4: for all NIC functions of the card
- Physical function 5: for iSCSI
- Physical function 6: for FCoE
- Physical function 7: Currently not assigned

xiv. Once the Unified Wire package is installed, open **Device Manager** again. Expand **Network adapters** section and now Chelsio adapter should be listed.

Note *Network device names for Chelsio's physical ports are assigned using the following convention: the port farthest from the motherboard will appear as the first Ethernet interface. However, for T5 40G adapters, the association of physical Ethernet ports and their corresponding network device names is opposite. For these adapters, the port nearest to the motherboard will appear as the first network interface.*

3. Software/Driver Installation

Chelsio Unified Wire can be installed using the Installer or the zip package. Refer the relevant section below depending on the method of installation selected.

3.1. Prerequisites

Latest Microsoft Visual C++ Redistributable packages for Visual Studio (x86 and x64) are required if you are planning to use installer.

3.2. Unified Wire Installer

Chelsio Unified Wire Installer provides two methods of installation: **GUI** or **CLI** mode. GUI mode provides an interactive GUI installer with customizable options. Whereas, CLI mode enables unattended installation of Chelsio drivers and software thereby relieving the user from monitoring the installation process and providing input via dialog boxes.

3.2.1. Enabling Test Signing

In case the drivers in the package are not WHQL certified, follow the steps mentioned below to enable test signing on your system:

i. Goto Start->Run command option, enter "cmd" and press OK. This will open the command prompt utility.
ii. Run the following command:

```
C:\Users\Administrator> bcdedit /set testsigning on
```

iii. Reboot the machine for the changes to take effect.

3.2.2. GUI mode (Installer)

i. Run the **ChelsioUwire-x.x.x.xx.exe** installer application.
ii. Click **Next** for the Chelsio End User License Agreement Window.

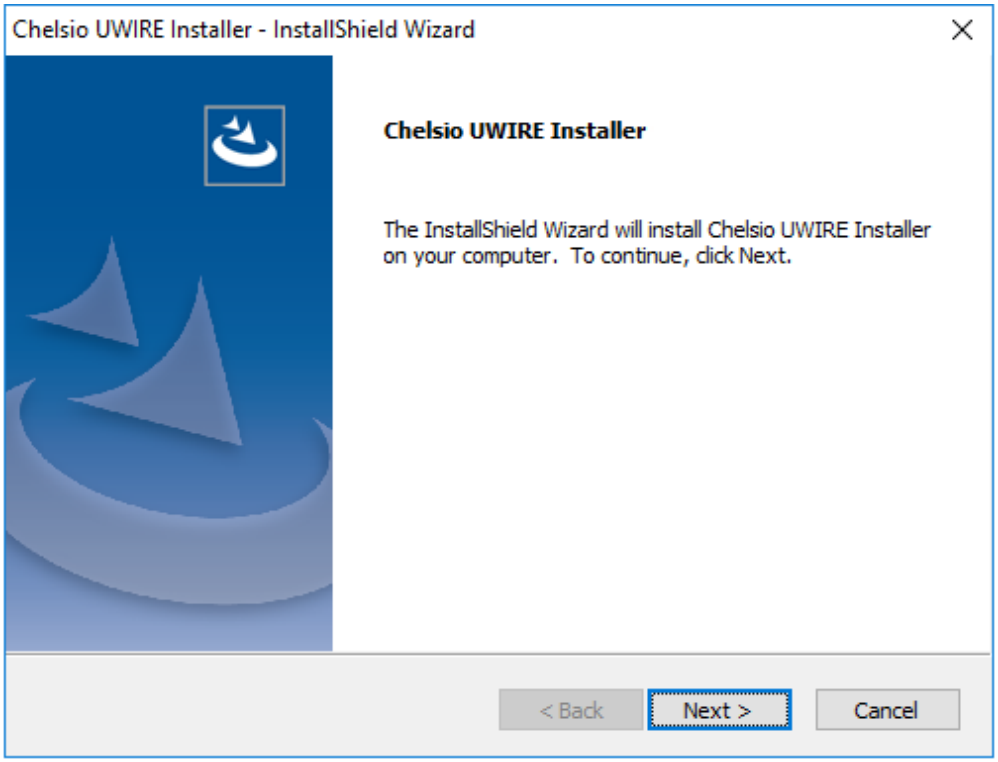


Figure 1 - Unified Wire installer welcome window

iii. Select the radio button **I accept the terms of the license agreement** and click **Next**

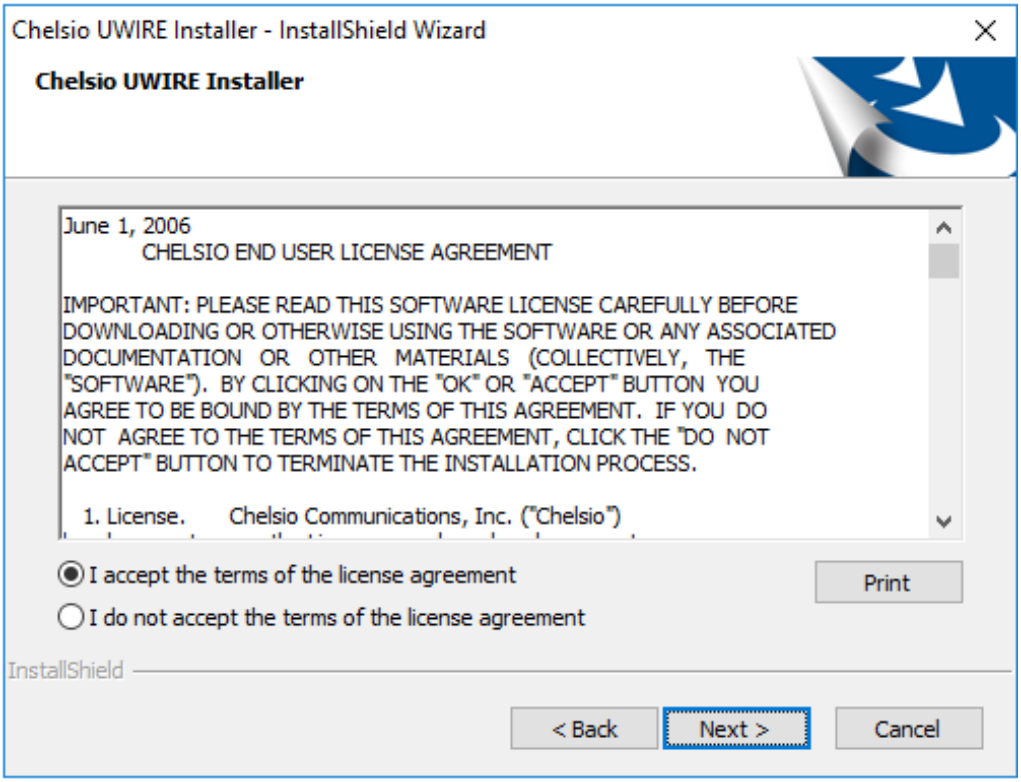


Figure 2 - Chelsio EULA window

iv. Now, either select **Complete** for complete package installation or else select **Custom** radio button to customize the installation.

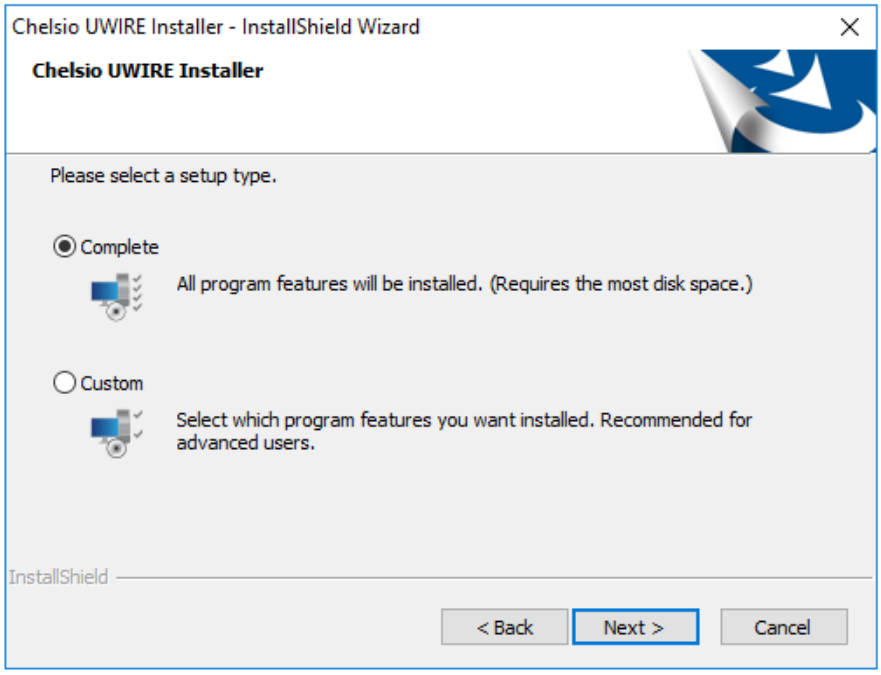


Figure 3 - Select setup (installation) type

Complete option will not install iSCSI Storport Miniport and iSER Initiator drivers. Use the **Custom** option to install them. Virtual Bus Driver (VBD) and NDIS (NIC) drivers are required for iSER Initiator to work, whereas iSCSI Storport Miniport needs VBD to function.

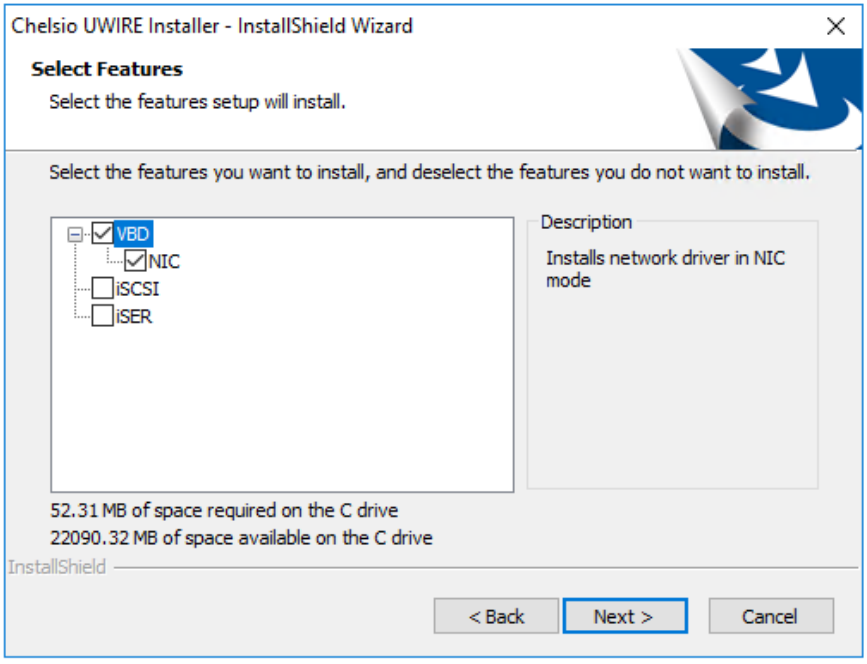


Figure 4 - Customizing the installation

v. Click **Install** to start the installation.

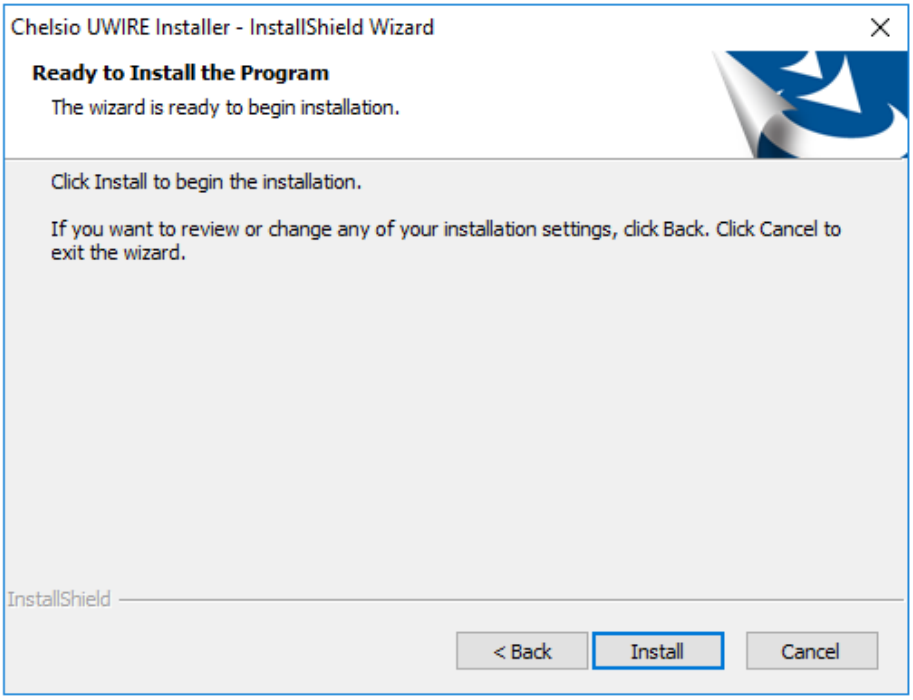


Figure 5 - Start installation

vi. Selected driver components will now be installed.

Figure 6 - Installation in progress

vii. Click **Finish** to exit from the Unified Wire Installer.

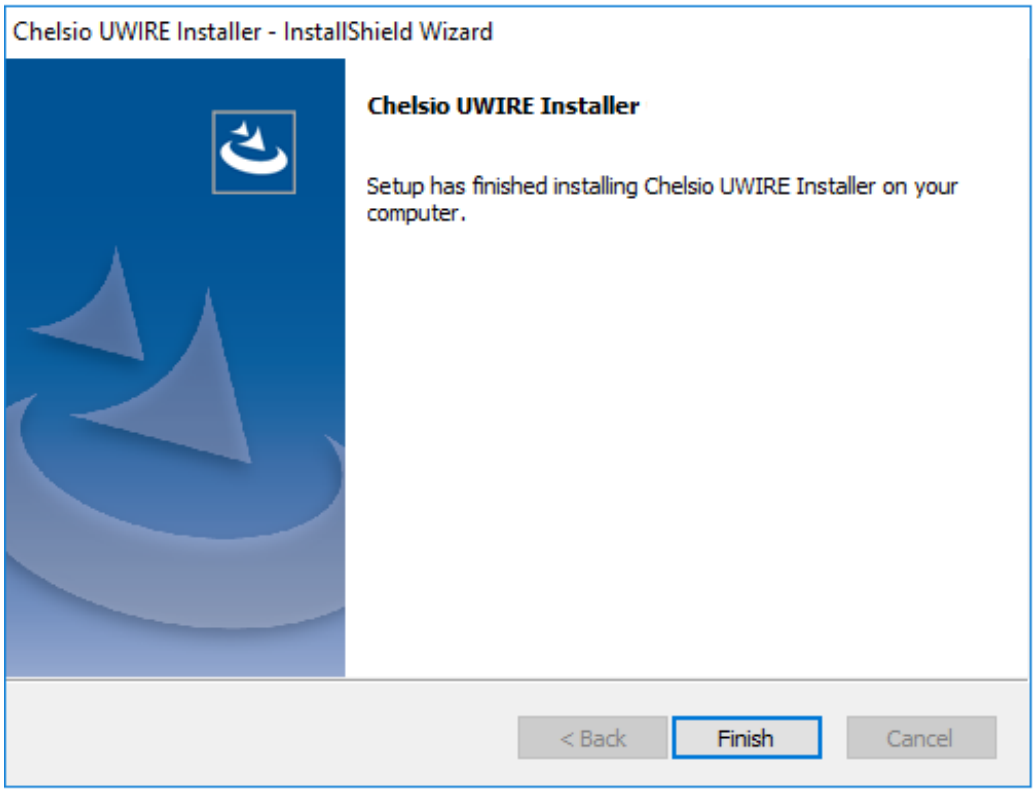


Figure 7 - Finishing Unified Wire installation

3.2.3. CLI mode (Silent Installation)

To install drivers/software using this feature, open **command prompt** and execute the following command:

```
C:\Users\Administrator>ChelsioUwire-x.x.x.xx.exe -in <driver(s)>
```

E.g.:

```
C:\Users\Administrator>ChelsioUwire-x.x.x.xx.exe -in all
```

The above command will install all drivers, except iSCSI Storport Miniport and iSER Initiator.

To add a driver to an existing list of already installed drivers, use the following command:

```
C:\Users\Administrator>ChelsioUwire-x.x.x.x.exe -add <driver(s)>
```

E.g.:

```
C:\Users\Administrator>ChelsioUwire-x.x.x.x.exe -add iSCSI
```

The above command will add iSCSI Storport Miniport driver.

To know more about other parameters and options, execute the following command:

```
C:\Users\Administrator>ChelsioUwire-x.x.x.xx.exe -help
```

The above command will create a help file, *ChelsioUwire-x.x.x.xx_help.txt,* in the present working directory which contains the complete list of command line syntax required for performing all the necessary CLI operations.

Note *A log file, ChelsioUwire-x.x.x.xx.log, is created in the same directory which keeps a record of all the commands executed and their results.*

3.3. Zip Package

Chelsio Unified Wire zip package provides two methods of installation: **Manual** and using **Offline Windows Image**. In Manual method, the driver components must be individually installed using the Device Manager. Using the second method, you can add Chelsio driver components to an offline OS image. The image can then be used to install Windows on your system.

3.3.1. Enabling Test Signing

In case the drivers in the package are not WHQL certified, please follow the steps mentioned below to enable test signing before installing Unified Wire:

i. Open command prompt and run the following command to enable test signing:

```
C:\Users\Administrator>bcdedit /set testsigning on
```

ii. Change your working directory to *ChelsioUwire-x.x.x.xx/Selfsign/* and run the following commands:

```
C:\ChelsioUwire-x.x.x.xx\Selfsign> .\certmgr.exe -add .\chelsiocert1.cer -s
-r LocalMachine root -all
C:\ChelsioUwire-x.x.x.xx\Selfsign> .\certmgr.exe -add .\chelsiocert1.cer -s
-r LocalMachine trustedpublisher -all
```

iii. Reboot the machine for the changes to take effect.

3.3.2. Manual

Important

- *The driver components need to be installed strictly in the order described here. Using any other order is not supported.*
- *Before proceeding, open Device Manager and disable NDIS Miniport driver (Network adapters->Chelsio Network Adapter) followed by Virtual Bus Driver (System devices->Chelsio Bus Enumerator).*

Virtual Bus Driver

i. Open **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on **System Devices**, right click on **Chelsio Bus Enumerator** and select **Update Driver Software.**
ii. Select **Browse my computer for driver software**.
iii. Select **Let me pick from a list of device drivers on my computer** and click **Next.**
iv. Click on **Have Disk** Button and on the next screen browse for **chvbdx64.inf** file and click **Open** and then **OK.**
v. Click **Next** and driver installation will progress. Click **Close** once the installation is complete.

NDIS Miniport driver

i. Open **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on **System Devices**, right click on the Chelsio network adapter and select **Update Driver Software**.
ii. Select **Browse my computer for driver software.**
iii. Select **Let me pick from a list of device drivers on my computer.**
iv. Click on **Have Disk** Button and on the next screen browse for **chnetx64.inf** and click **Open** and then **OK.**
v. Click **Next** and driver installation will progress. Click **Close** once the installation is complete.

Repeat the above steps for the other adapters**.**

iSCSI Storport Miniport driver

i. Open **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on **System Devices**, right click on the **Chelsio Bus Enumerator** and select **Properties.**
ii. Under **Advanced** tab, select **iSCSI Instances** and set the *Value* to **default**. Click **OK**.
iii. Under **Other Devices**, select **Chelsio iScsi Function on Port#00**, right-click and select **Update Driver Software**.
iv. Select **Browse my computer for driver software.**
v. Select **Let me pick from a list of device drivers on my computer** and click **Next.**
vi. Click on **Have Disk** Button and on the next screen, browse for **cht4iscsi.inf** file and click **Open** and then **OK.**
vii. Click **Next** and driver installation will progress. Click **Close** once the installation is complete.

Repeat the same procedure for other ports.

- **iSER Initiator**

i. Open **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on **System Devices**, right click on the **Chelsio Bus Enumerator** and select **Properties.**
ii. Under **Advanced** tab, select **iSER Instances** and set the *Value* to **default**. Click **OK**.
iii. Under **Other Devices**, select **Chelsio iSer Function on Port#00**, right-click and select **Update Driver Software**.
iv. Select **Browse my computer for driver software.**
v. Select **Let me pick from a list of device drivers on my computer** and click **Next.**
vi. Click on **Have Disk** Button and on the next screen, browse for **chiserx64.inf** file and click **Open** and then **OK.**
vii. Click **Next** and driver installation will progress. Click **Close** once the installation is complete.

Repeat the same procedure for other ports.

- **Generic Function**

i. Open **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on **System Devices**, right click on **Chelsio Generic Function** and select **Update Driver Software**.
ii. Select **Browse my computer for driver software.**
iii. Select **Let me pick from a list of device drivers on my computer.**
iv. Click on **Have Disk** Button and on the next screen browse for **chnulx64.inf** file and click **Open** and then **OK.**
v. Click **Next** and driver installation will progress. Click **Close** once the installation is complete.

3.3.3. Offline Windows Image

You can use Windows Deployment Image Servicing and Management (DISM) tool to add Chelsio driver components (*.inf* files) to an offline Windows image. For more information, visit the following links:

- http://technet.microsoft.com/en-us/library/hh825070.aspx
- http://blogs.technet.com/b/heyscriptingguy/archive/2012/09/27/use-the-powershell-dism-cmdlets-to-manage-windows-8.aspx

4. Firmware Update

The driver will auto-load the firmware if an update is required. The firmware version can be verified using *cxgbtool*:

```
C:\Users\Administrator>cxgbtool nic0 debug version
```

In case of installer, the firmware binaries will be copied to <*system_drive*>*ChelsioUwire* *Firmware*\ folder during installation.

For zip package, the binaries will be present in *ChelsioUwire-x.x.x.xx**firmware*\ folder.

5. Software/Driver Update

Chelsio Unified Wire can be updated using Windows update, installer or zip package. The update method for each is described below:

5.1. Windows Update

i. Make sure that you have an active internet connection.

ii. Open **Device Manager**, click on **System Devices**, right-click on the **Chelsio Bus Enumerator**, and then select **Update Driver Software..**

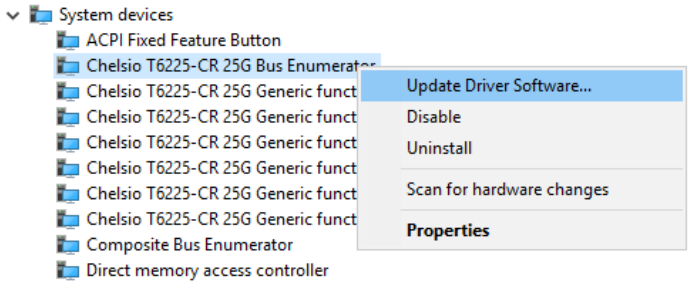


Figure 8 – Updating VBD driver

iii. Select **Search automatically for updated driver software**.

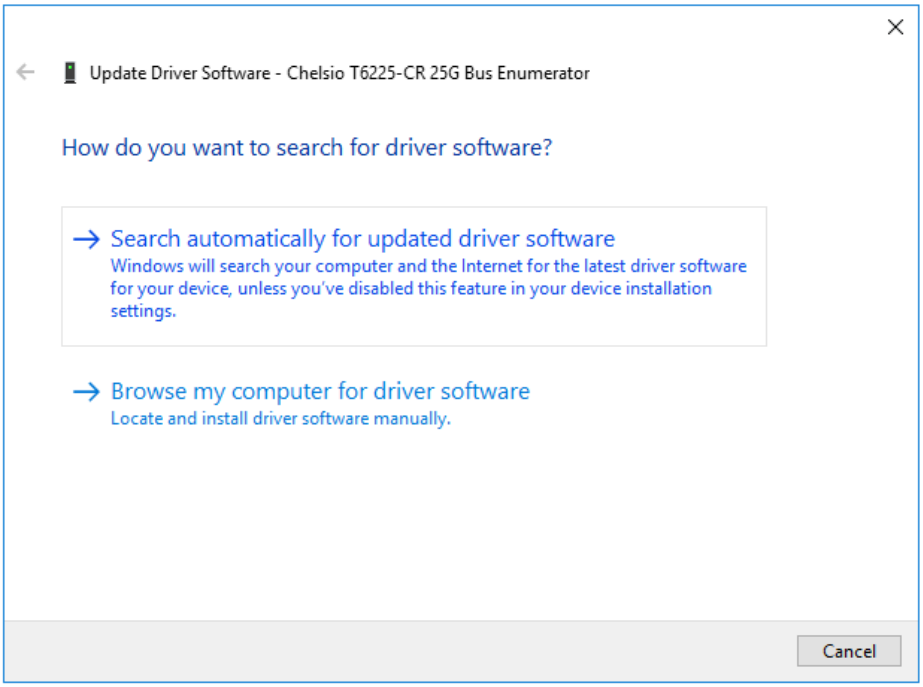


Figure 9 – Searching for driver using Windows update

iv. Windows will search and install the latest driver. Click **Close**.

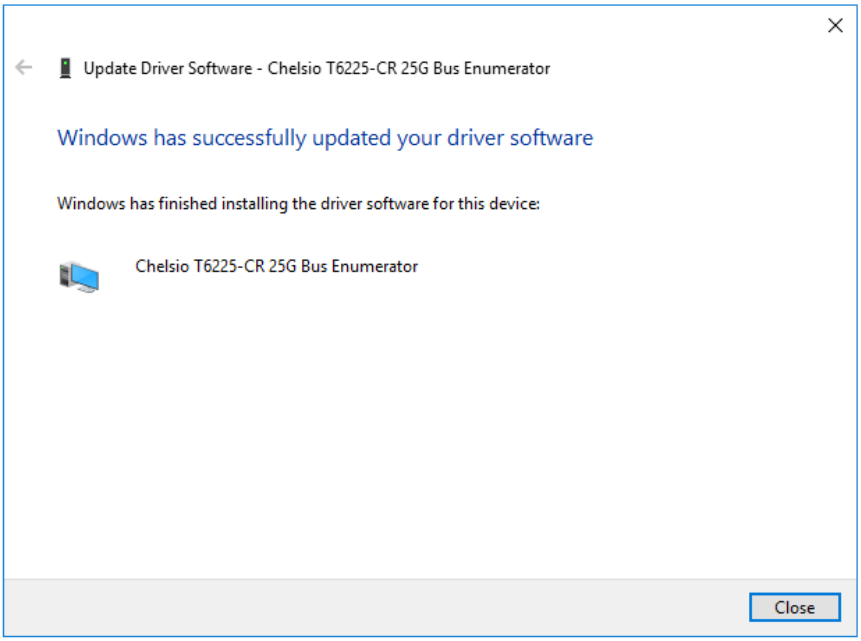


Figure 10 - Driver update successful

v. Click on **Network Adapters**, right-click on **Chelsio Network Adapter** and select **Update Driver Software..**

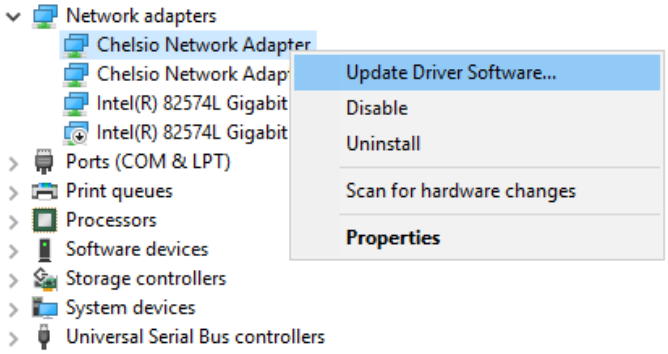


Figure 11 - Updating NDIS driver

vi. Select **Search automatically for updated driver software**

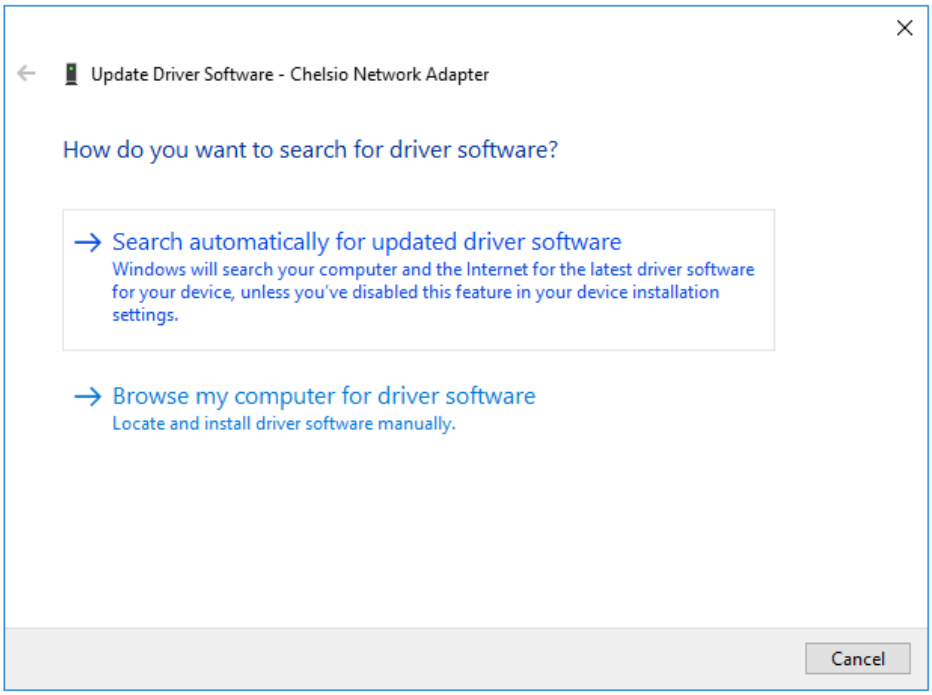


Figure 12- Searching for driver using Windows update

vii. Windows will search and install the latest driver. Click **Close**.

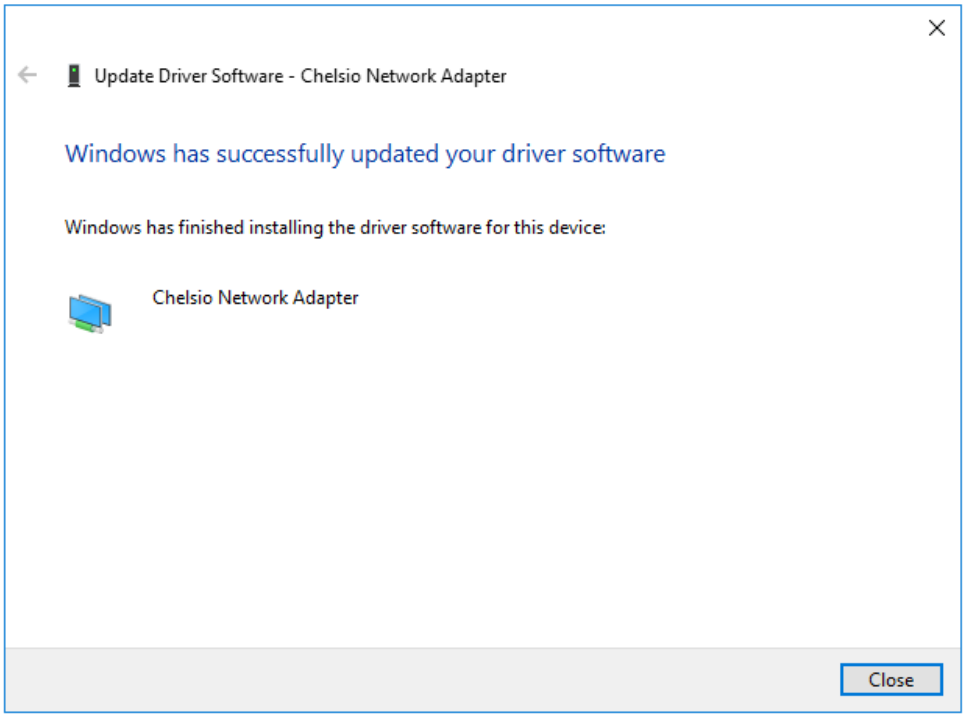


Figure 13 - Driver update successful

viii. Click on **Storage controllers**, right-click on **Chelsio T6 iSCSI Initiator** and select **Update Driver Software..**

Note *If Chelsio iSCSI interface is not listed, verify if you have specified the number of iSCSI instances in VBD parameters.*

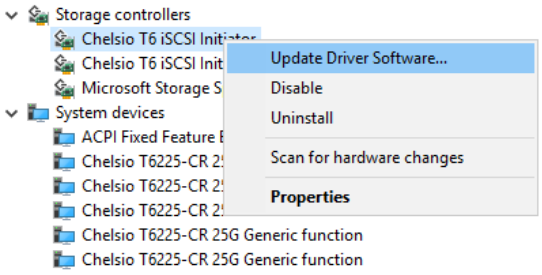


Figure 14 - Updating iSCSI Storport Miniport driver

ix. Windows will search and install the latest driver. Click **Close**.

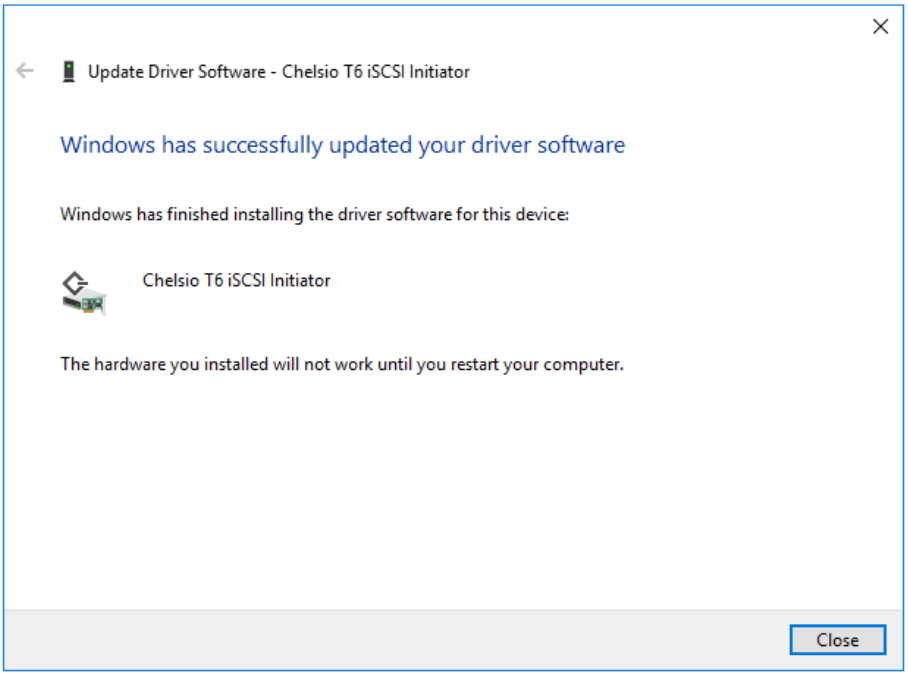


Figure 15 - Driver update successful

x. Click on **Storage controllers**, right-click on **Chelsio T6 iSER Initiator** and select **Update Driver Software..**

Note *If Chelsio iSER interface is not listed, verify if you have specified the number of iSER instances in VBD parameters.*

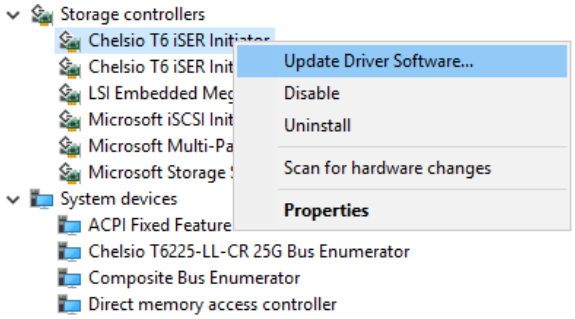


Figure 16 - Updating iSER Initiator driver

xi. Select **Search automatically for updated driver software**

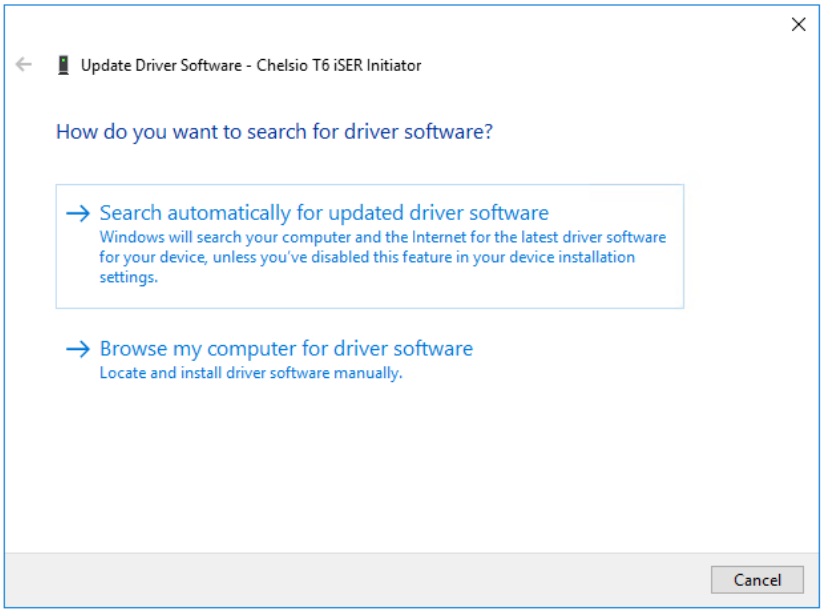


Figure 17- Searching for driver using Windows update

xii. Windows will search and install the latest driver. Click **Close**.

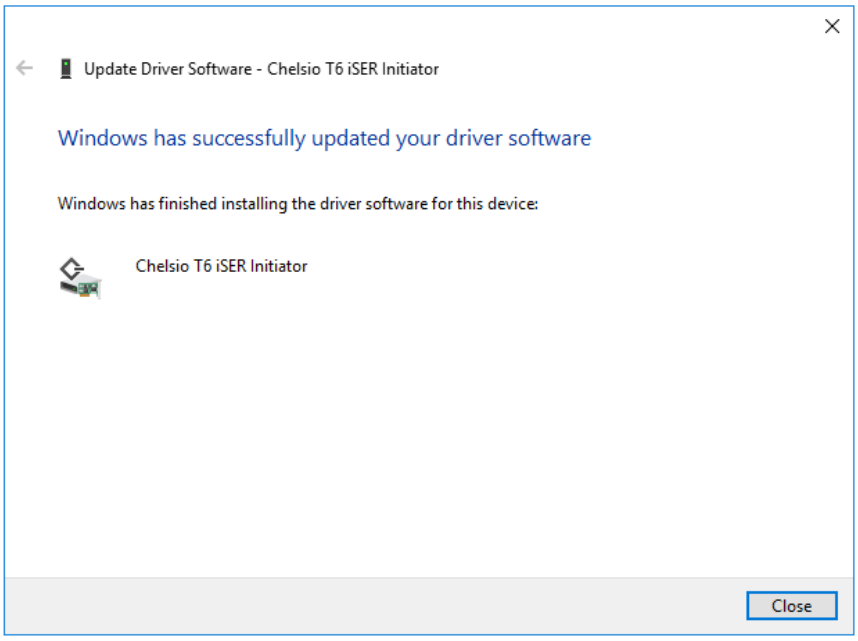


Figure 18 - Driver update successful

xiii. Open the **Device Manager,** click on **System devices**, right-click on **Chelsio Generic Function** and select **Update Driver Software..**

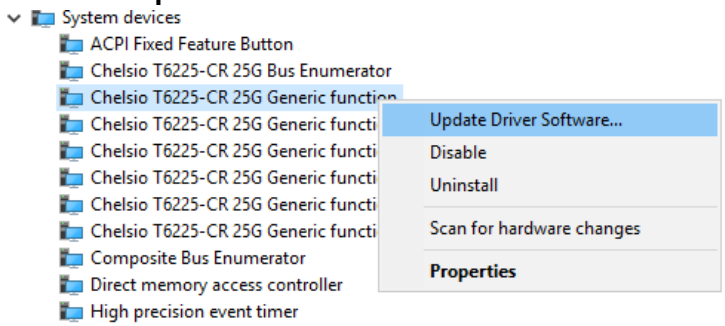


Figure 19 - Updating Generic function driver

xiv. Select **Search automatically for updated driver software**

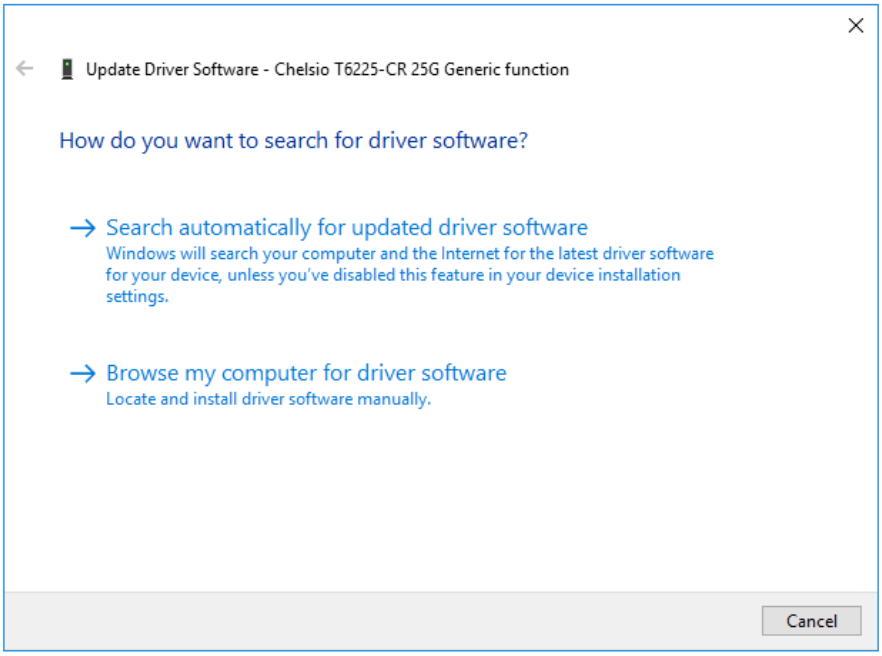


Figure 20 - Searching for driver using Windows update

xv. Windows will search and install the latest driver. Click **Close**.

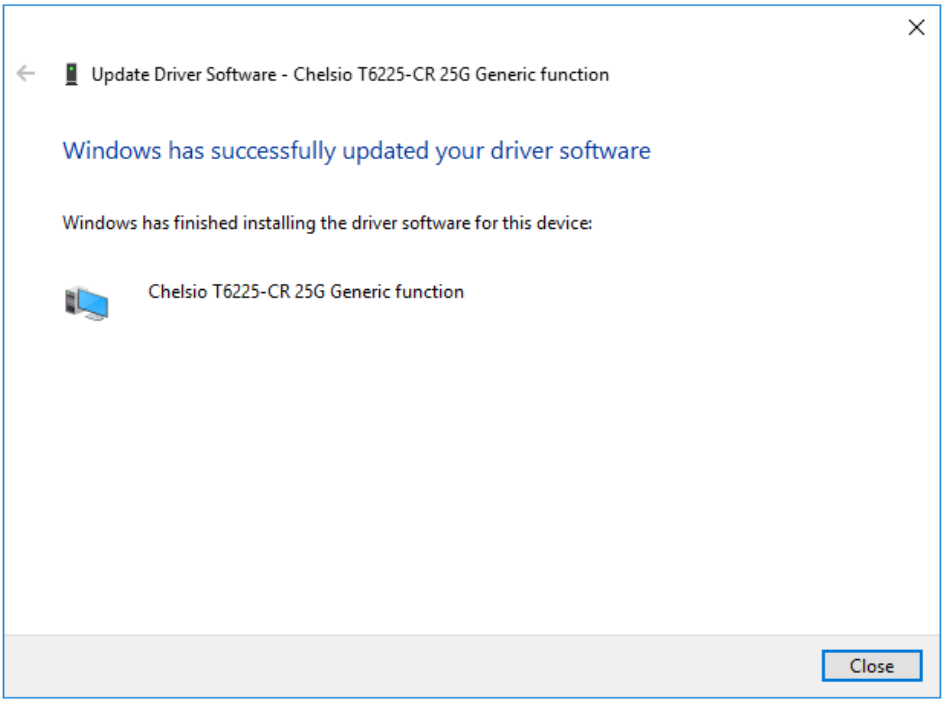


Figure 21 - Driver update successful

5.2. Unified Wire Installer

i. Run the **ChelsioUwire-x.x.x.xx.exe** installer application.
ii. Click **Next** to start the update.

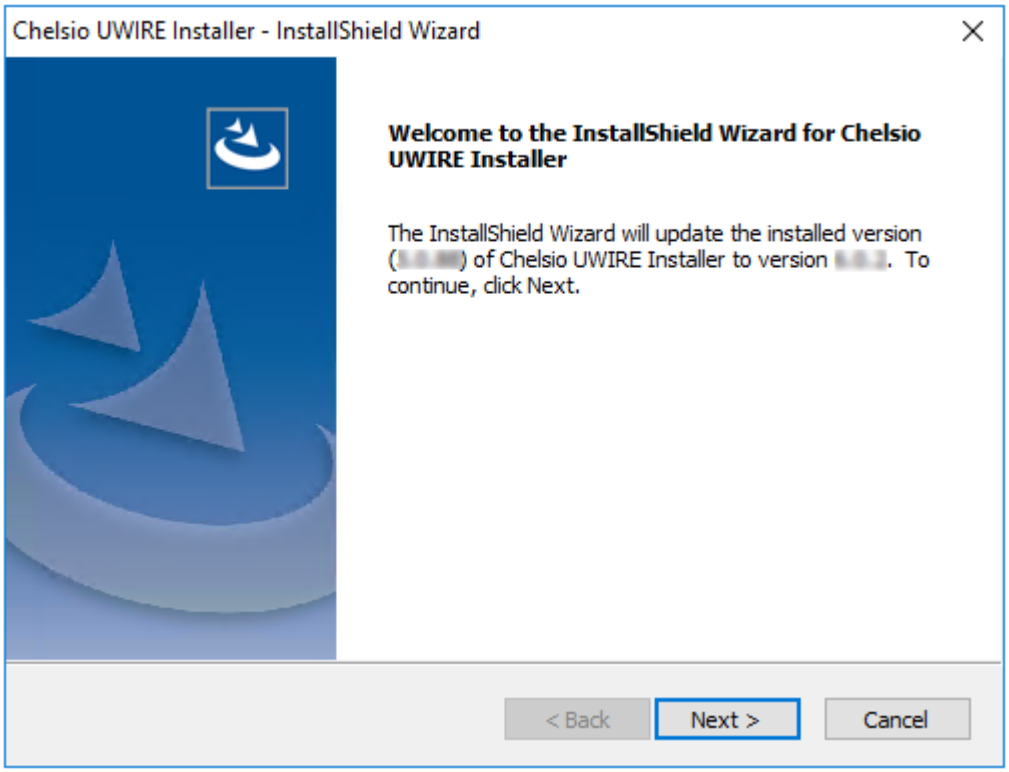


Figure 22 - Unified Wire installer welcome window

iii. Click **Finish** to exit from the Unified Wire Installer.

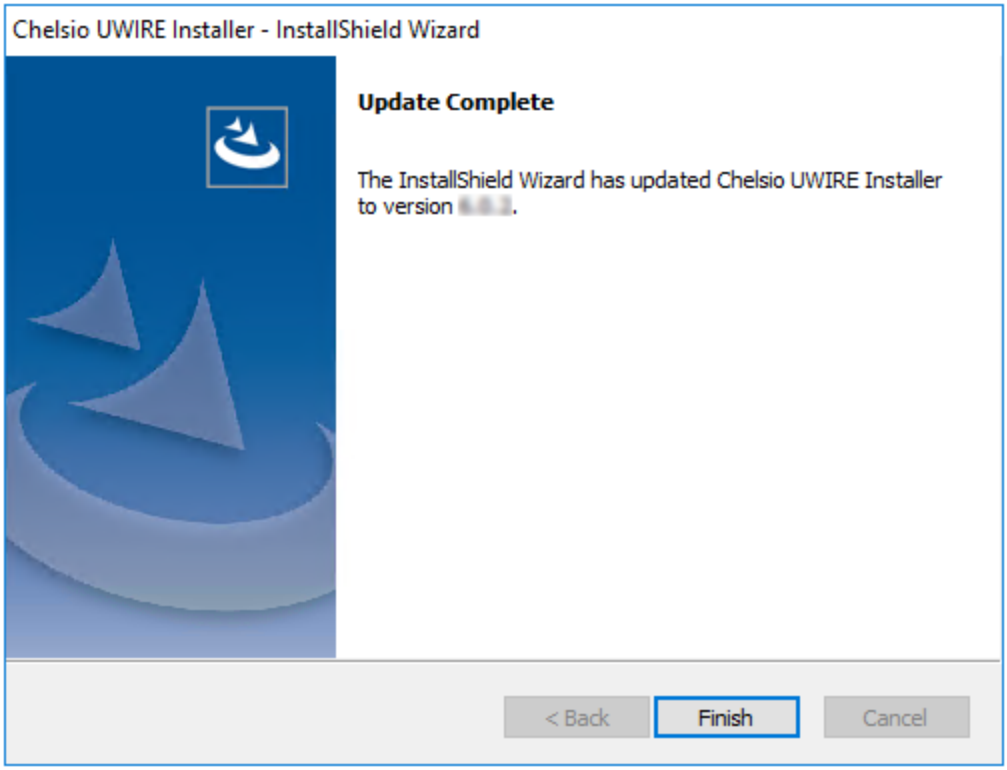


Figure 23 - Finishing Unified Wire update

5.3. Zip Package

The method to update Unified Wire using zip package is same as installation. Refer the installation section for more information.

6. Configuring Chelsio Network Interfaces

In order to test Chelsio adapters' features, it is required to use at least two machines, each with Chelsio's network adapters. These machines can be connected directly (back-to-back) or with a switch.

6.1. Configuring 40G Adapters

You can use the **chelsio_adapter_config.ps1** script to configure Chelsio 40G adapters in any of the following three operational modes:

- **DEFAULT (2X40G)**: This is the default mode of operation where each port functions as 40Gbps link. The port nearest to the motherboard will appear as the first network interface (Port 0).
- **SPIDER (4X10G)**: In this mode, port 0 functions as 4 10Gbps links and port 1 is disabled.
- **QSA (2X10G)**: This mode adds support for QSA (QSFP to SFP+) modules, enabling smooth, cost-effective, connections between 40 Gigabit Ethernet adapters and 1 or 10 Gigabit Ethernet networks using existing SFP+ based cabling. The port farthest from the motherboard will appear as the first network interface (Port 0).

6.1.1. Windows GUI Machine (Installer & Zip Package)

i. In case of Installer, the configuration script and cxgbtool.exe will be copied to *<system_drive>\Windows\System32* folder during installation.

   If you are using the zip package, copy *cxgbtool.exe* from *ChelsioUwire-x.x.x.xx* folder to *<system_drive>\Windows\System32* and change your working directory to *ChelsioUwire-x.x.x.xx\Adapter Configuration*.

ii. Open PowerShell with administrative privileges.
iii. Run the adapter configuration script and enter the index of the 40G adapter for which the configuration needs to be updated. Hit [Enter].

Installer:

```
PS C:\Users\Administrator> chelsio_adapter_config.ps1
```

Zip Package:

```
PS D:\ChelsioUwire-x.x.x.xx\Adapter Configuration>
.\chelsio_adapter_config.ps1
```

Figure 24 - Selecting adapter

iv. Select *Port settings* (option 3) as the configuration type. Hit [Enter].

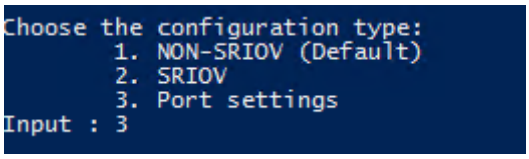


Figure 25 - Port settings

v. Select the operation mode and then enter *y* to confirm.

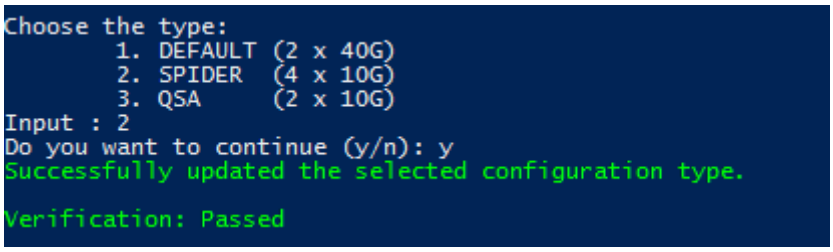


Figure 26 - Operation mode

vi. Open **Device Manager**, click on **System Devices** and right-click on the 40G Chelsio adapter selected in step (iii).
vii. Select **Disable** and then **Enable** for changes to take effect.

6.2. Assigning IP Address

If your computer is connected to a network with a DHCP server, Chelsio network interfaces will acquire DHCP IPs automatically.

To assign static IPs, follow the steps mentioned below:

i. Double click on the Network Connections icon and choose the Chelsio card entry and double click it.
ii. Click on the Properties button from the Local Area Connection X Status.
iii. Select “Internet Protocol (TCP/IP)” from the list and click on Properties button below it.
iv. From the Internet Protocol (TCP/IP) Properties window, assign an IP Address (e.g. 192.169.1.10) and subnet mask (e.g. 255.255.255.0).
v. Click on Ok and close on the other window.
vi. Check to see if you can ping to some other address on this subnet.

7. *cxgbtool* help

The *cxgbtool* command queries or sets various aspects of Chelsio network interface cards. It complements standard tools used to configure network settings and provides functionality not available through such tools.

Some of the commands provided can be used to query running statistics to aid in debugging.

Syntax

```
cxgbtool <vbdIface|nicIface|iSCSIIface|rdma> <category> <command> <options>
```

Definitions

`nicIface` is the name of the network device to work on, given in the format `nic0`, `nic1`.

`vbdIface` is the name of the Chelsio Bus Enumerator instance to work on, given in the format `vbd0`, `vbd1`

`iscsIface` is the name of the Chelsio iSCSI interface to work on, given in the format `iSCSI0`, `iSCSI1`

`rdma` is the name of the Chelsio RDMA interface to work on.

`category` is one of the cxgbtool categories, i.e., boot, cudbg, debug, firmware, hardware

`command` is the action you wish to perform on the adapter or property you wish to view/change.

`option` is the optional parameter to be provided with *command*.

- **Help**
  - Displaying cxgbtool help: `cxgbtool`
  - Displaying list of available cxgbtool commands: `cxgbtool -h all`
  - Displaying category specific help:
    `cxgbtool <vbdIface|nicIface|iSCSIIface> <category> -h`
  - Displaying list of cudbg entities: `cxgbtool <vbdIface> cudbg dbg -h`
  - Displaying RDMA help: `cxgbtool rdma -h`

boot

loadphy

Description: Load phy firmware.

Syntax: `cxgbtool <vbdIface> boot loadphy <phyFile>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 boot loadphy .\Firmware_1.25.c1.Chelsio.cld
```

loadboot

Description: Flash option ROM image.

Syntax: `cxgbtool <vbdIface> boot loadboot <bootImage> <pf|offset|clear>`

`pf`: Flash Option ROM image to the offset defined by the PFs EXPROM_OFST in the serial configuration. Valid value ranging 0-7.

`offset`: Flash Option ROM image to the offset specified by the user.

`clear`: The flash area reserved for Option ROM image will be cleared.

Examples:

- Flashing Option ROM image to the offset defined by the PFs EXPROM_OFST:

```
C:\Users\Administrator>cxgbtool vbd0 boot loadboot C:\Users\Administrator\Desktop\cubt4.bin pf 0
```

- Flashing Option ROM image to user specified offset:

```
C:\Users\Administrator>cxgbtool vbd0 boot loadboot c:\Users\Administrator\Desktop\cubt4.bin offset 0x100
```

loadboot-cfg

Description: Load boot configuration file.

Syntax: `cxgbtool <vbdIface> boot loadboot-cfg <bootConfigImg|clear>`

`clear`: The flash area reserved for boot configuration file will be cleared.

Examples:

- Loading boot configuration file:

```
C:\Users\Administrator>cxgbtool vbd0 boot loadboot-cfg Desktop\bootcfg
Cxgb_CmdLoadBootCfg: BootROM length: 1668, value2add:0
```

- Clearing the flash area:

```
C:\Users\Administrator>cxgbtool vbd0 boot loadboot-cfg clear
```

- **seeprom**

Important *Use this option with caution. Incorrect usage may render the adapter useless.*

Description: Read/Write SEEPROM (init+VPD) data.

Syntax:

- Read SEEPROM data to a bin file: `cxgbtool <vbdIface> boot seeprom read`
- Display SEEPROM data onscreen: `cxgbtool <vbdIface> boot seeprom dump`
- Write SEEPROM data: `cxgbtool <vbdIface> boot seeprom write <file>`
- Verify SEEPROM data: `cxgbtool <vbdIface> boot seeprom verify <file>`

Note *The SEEPROM file should be in binary format (.bin).*

Examples:

- Reading SEEPROM data:

```
C:\Users\Administrator>cxgbtool vbd0 boot seeprom read
Reading Seeprom data to seeprom.bin
```

- Displaying SEEPROM data onscreen

```
C:\Users\Administrator>cxgbtool vbd0 boot seeprom dump
Reading Seeprom data
Offset          Values
------          ------
0x0000:         15 07 02 00 00 00 00 00  30 00 07 22 80 00 32 04

0x0010:         31 b9 e5 68 20 70 01 10  b8 00 08 16 00 12 00 00

0x0020:         00 00 00 00 00 00 04 20  01 5c 00 00 64 de 1d 00

0x0030:         00 00 00 00 00 00 00 00  00 00 c3 18 43 00 c8 00

0x0040:         fe 01 86 36 f4 01 00 00  30 01 80 a2 05 23 00 00
```

- Writing SEEPROM data:

```
C:\Users\Administrator>cxgbtool vbd0 boot seeprom write seeprom.bin

Changing the init/vpd can cause the card to become inaccessible if the operation is interrupted
Do you want to flash your T6225-CR (SN:PT43160304,PN:11012096004,NA:0007433987F0) card? (y/n) : y
Hardware configuration changed successfully.
Please reboot for the changes to take effect
```

- Verifying SEEPROM data:

```
C:\Users\Administrator>cxgbtool vbd0 boot seeprom verify seeprom.bin
Verifying seeprom data against seeprom.bin

verification started
Verification : PASSED
```

- **serialinit**

Important *Use this option with caution. Incorrect usage may render the adapter useless.*

Description: Read/Write serialinit data.

Syntax:

- Read serialinit data to a bin file: `cxgbtool <vbdIface> boot serialinit read`
- Display serialinit data onscreen: `cxgbtool <vbdIface> boot serialinit dump`
- Write serialinit data: `cxgbtool <vbdIface> boot serialinit write <file>`
- Verify serialinit data: `cxgbtool <vbdIface> boot serialinit verify <file>`

Note *The initdata file should be in binary format (.bin).*

Examples:

- Reading serialinit data

```
C:\Users\Administrator>cxgbtool vbd0 boot serialinit read
Reading Serial Init data to seeprom_init.bin
```

- Displaying serialinit data onscreen

```
C:\Users\Administrator>cxgbtool vbd0 boot serialinit dump
Reading Serial Init data
Offset          Values
------          ------
0x0000:         15 07 02 00 00 00 00 00  30 00 07 22 80 00 32 04

0x0010:         31 b9 e5 68 20 70 01 10  b8 00 08 16 00 12 00 00

0x0020:         00 00 00 00 00 00 04 20  01 5c 00 00 64 de 1d 00

0x0030:         00 00 00 00 00 00 00 00  00 00 c3 18 43 00 c8 00

0x0040:         fe 01 86 36 f4 01 00 00  30 01 80 a2 05 23 00 00

0x0050:         00 00 f0 bc cc 08 20 27  90 98 80 91 02 c0 0b 00

0x0060:         6a 01 a4 46 12 16 c6 68  00 27 02 7d 00 14 00 40

0x0070:         99 08 4b 28 00 4a 28 f2  5f 06 42 16 e0 7d 00 00

0x0080:         00 00 00 00 00 80 08 00  04 00 08 00 04 00 53 05

0x0090:         00 00 0f c0 00 00 04 08  7d 42 9f c4 07 00 98 01

0x00a0:         00 30 10 02 02 e3 00 0c  00 40 80 d0 27 f4 49 7c

0x00b0:         00 80 29 00 00 03 21 20  30 0e c0 00 00 04 08 7d
```

- Writing serialinit data

```
C:\Users\Administrator>cxgbtool vbd0 boot serialinit write C:\Users\Administrator\Desktop\t62100_lp_cr_init_800_1050_gen3_x16_mfg.bin
Changing the init/vpd can cause the card to become inaccessible if the operation is interrupted
Do you want to flash your T62100-LP-CR (SN:RE41160048,PN:11012106002,NA:00074304B390) card? (y/n) : y
Hardware configuration changed successfully.
Please reboot for the changes to take effect
```

- Verifying serialinit data

```
C:\Users\Administrator>cxgbtool vbd0 boot serialinit verify C:\Users\Administrator\Desktop\t62100_lp_cr_init_800_1050_gen3_x16_mfg.bin
Verifying Serial Init data against C:\Users\Administrator\Desktop\t62100_lp_cr_init_800_1050_gen3_x16_mfg.bin
verification started
Verification : PASSED
```

▪ vpd

Important *Use this option with caution. Incorrect usage may render the adapter useless.*

Description: Read/write Vital Product Data (VPD).

Syntax:

- Read VPD to a bin file: `cxgbtool <vbdIface> boot vpd read`
- Display VPD onscreen: `cxgbtool <vbdIface> boot vpd dump`
- Write VPD: `cxgbtool <vbdIface> boot vpd write <file>`
- Verify VPD: `cxgbtool <vbdIface> boot vpd verify <file>`

Note *The vpd file should be in binary format (.bin).*

Examples:

- Reading VPD:

```
C:\Users\Administrator>cxgbtool vbd0 boot vpd read
Reading Vpd data to seeprom_vpd.bin
```

- Displaying VPD onscreen:

```
C:\Users\Administrator>cxgbtool vbd0 boot vpd dump
Reading Vpd data
Offset          Values
------          ------
0x0000:         82 10 00 54 36 32 31 30  30 2d 4c 50 2d 43 52 20

0x0010:         20 20 20 90 ea 00 50 4e  10 31 31 30 31 32 31 30

0x0020:         36 30 30 32 20 20 20 20  20 45 43 10 30 30 30 30

0x0030:         30 30 30 30 30 30 30 30  30 30 30 30 53 4e 18 52

0x0040:         45 34 31 31 36 30 30 34  38 20 20 20 20 20 20 20

0x0050:         20 20 20 20 20 20 20 52  56 a6 a2 00 00 00 00 00

0x0060:         00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

0x0070:         00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

0x0080:         00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

0x0090:         00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

0x00a0:         00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

0x00b0:         00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

- Writing VPD:

```
C:\Users\Administrator>cxgbtool vbd0 boot vpd write seeprom_vpd.bin

Changing the init/vpd can cause the card to become inaccessible if the operation is interrupted
Do you want to flash your T6225-CR (SN:PT43160304,PN:11012096004,NA:0007433987F0) card? (y/n) : y
Hardware configuration changed successfully.
Please reboot for the changes to take effect
```

- Verifying VPD:

```
C:\Users\Administrator>cxgbtool vbd0 boot vpd verify seeprom_vpd.bin
Verifying Vpd data against seeprom_vpd.bin

verification started
Verification : PASSED
```

- **vpdparams**

Description: Display adapter information like serial number, adapter name, EC, core clock, part Number and network address (MAC).

Syntax: `cxgbtool <vbdIface> boot vpdparams`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 boot vpdparams
Serial Number   : PT43160304
Id              : T6225-CR
Ec              : 0000000000000000
Core Clock      : 500000
Pn              : 11012096004
Network Addr    : 0007433987F0
```

cudbg

cudbg actions

- **collect**

Description: Collect Chelsio adapter debug logs to a compressed file.

Syntax: `cxgbtool <vbdIface> cudbg collect <entities> <outputFile> <options>`

Examples:

- Collecting debug log for all entities

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect all dump_file
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file dump_file... size: 12634452 bytes
cxgbtool: Done writing cudbg data to file dump_file
```

- Collecting debug log skipping specific debug entities

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect all dump_file skip edc1,edc0
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file dump_file... size: 12082984 bytes
cxgbtool: Done writing cudbg data to file dump_file
```

- Collecting debug log avoiding entities that can affect running traffic

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect all dump_file safe
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file dump_file... size: 48484 bytes
cxgbtool: Done writing cudbg data to file dump_file
```

- **view**

Description: Display debug log onscreen stored in a compressed dump file.

Syntax: `cxgbtool <vbdIface> cudbg view <entity> <inputFile> <options>`

Examples:

- Displaying debug log for a specific entity onscreen.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg view clk dump_file
cudbg_view() dbg entity : clk
Core clock period: 2 ns
TP timer tick: 32.768 us
TCP timestamp tick: 1048.576 us
DACK tick: 8.192 us
DACK timer: 488 us
Retransmit min: 960 us
Retransmit max: 9765440 us
Persist timer min: 976544 us
Persist timer max: 9765440 us
Keepalive idle timer: 7031116800 us
Keepalive interval: 73240800 us
Initial SRTT: 244128 us
FINWAIT2 timer: 9765440 us
```

- Displaying debug log on the screen skipping specific entities

```
C:\Users\Administrator>cxgbtool vbd0 cudbg view all dump_file2 skip vpddata,regdump,cimla
cudbg_view() dbg entity : devlog
      Seq#          Tstamp     Level  Facility  Message
        15          539161      INFO      CORE  configured with caps nbm|link 0x00000005 switch|nic 0x00030003 toe|rdma 0x00010003 iscsi|crypto 0x00010000 fcoe:0x0
        16          539184      INFO        HW  hw_tp_tcp_tunings: tuning for cluster environment
        17          540345      INFO        HW  MC: GPO 0x9, RLO 0x1
        18          541130      INFO        HW  MC: GPO 0x9, RLO 0x1
        19          541136      INFO        HW  DDR init complete, Beginning calibration
        20          541762      INFO        HW  MC: Initial Calibration PASSED
        21          546415      INFO       RES  le configuration: nentries 3072 clip 384 normal filter 384 hi priority filter 0 server 128 active 2560 hash 16384 nserversram 0
        22          547142      INFO       DCB  dcbx_ieee_cmdh[0] requesting DCB_IEEE_CMD
        23          548132      INFO       DCB  dcbx_ieee_cmdh[1] requesting DCB_IEEE_CMD
        24         1334252      INFO       RES  mpartition_init: moved pmrx_start from 0x00800000 to 0x006a0000 (EDRAM)
        25         1334254      INFO      CORE  flr_timer_start: flowc_id 39212 00000000A07FFD80 buf 00000000A0336180
        26         1350378      INFO      PORT  tls_key_start:0xffffffff, tls_key_size:0
        27         1351406      INFO       RES  le initialization: nentries 3072 route 0 clip 384 filter 384 server 128 active 2560 hash 16384 nserversram 0
        28         2511302      INFO      PORT  module[0]: fec ability of cable 0x800
        29         2511303      INFO      PORT  module[0]: gpio 9 vendor id 001e62, identifier 0x0d, SFP28(byte 36/192) 0x0c, SFP(byte 3/131) 0xff, 1G (byte 6) 0xff
        30         2511303      INFO      PORT  optical length(byte 15/142) 0, copper cable(byte 8/147) 0xff, length(byte 18/146) 2, module_type 0x04
        31         2511304      INFO      PORT  hw_mac_init_port[0], ptype 0x11, speed 0x20, lanes 0xf, fec 0x800
        32         2512304      INFO        TM  pktsched channel 0 sets speed (from 0) to 100000000 kbps
        33         2512305    NOTICE        TM  ch_cl_rate[0/0]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        34         2512306    NOTICE        TM  ch_cl_rate[0/1]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        35         2512307    NOTICE        TM  ch_cl_rate[0/2]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        36         2512308    NOTICE        TM  ch_cl_rate[0/3]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        37         2512308    NOTICE        TM  ch_cl_rate[0/4]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        38         2512309    NOTICE        TM  ch_cl_rate[0/5]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        39         2512310    NOTICE        TM  ch_cl_rate[0/6]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        40         2512310    NOTICE        TM  ch_cl_rate[0/7]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        41         2512311    NOTICE        TM  ch_cl_rate[0/8]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        42         2512312    NOTICE        TM  ch_cl_rate[0/9]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        43         2512312    NOTICE        TM  ch_cl_rate[0/10]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        44         2512313    NOTICE        TM  ch_cl_rate[0/11]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        45         2512314    NOTICE        TM  ch_cl_rate[0/12]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        46         2512314    NOTICE        TM  ch_cl_rate[0/13]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        47         2512315    NOTICE        TM  ch_cl_rate[0/14]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        48         2512315    NOTICE        TM  ch_cl_rate[0/15]: capped class rate from requested 100000000 to configured (effective) channel rate 99990260
        49         2518866      INFO      PORT  module[1]: fec ability of cable 0x800
        50         2518866      INFO      PORT  module[1]: gpio 14 vendor id 001e62, identifier 0x0d, SFP28(byte 36/192) 0x0c, SFP(byte 3/131) 0xff, 1G (byte 6) 0xff
        51         2518867      INFO      PORT  optical length(byte 15/142) 0, copper cable(byte 8/147) 0xff, length(byte 18/146) 2, module_type 0x04
        52         2518867      INFO      PORT  hw_mac_init_port[1], ptype 0x11, speed 0x20, lanes 0xf0, fec 0x800
        53         2519865      INFO        TM  pktsched channel 1 sets speed (from 0) to 100000000 kbps
```

- **readflash**

Description: Collect debug log from adapter flash memory to a compressed file.

Syntax: `cxgbtool <vbdIface> cudbg readflash [outputfile]`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg readflash crash_dump
cxgbtool: Writing memory block of size 626688 bytes to file crash_dump...
cxgbtool: Done writing memory block to file crash_dump
```

- **info**

Description: Display summary of debug log present in a compressed dump file onscreen.

Syntax: `cxgbtool <vbdIface> cudbg info [inputFile]`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg info dump_file
regdump              compressed size 9972
devlog               compressed size 4968
cimla                compressed size 300
cimmala              compressed size 76
cimqcfg              compressed size 300
ibqtp0               compressed size 488
ibqtp1               compressed size 1828
ibqulp               compressed size 376
ibqsge0              compressed size 436
ibqsge1              compressed size 1768
ibqncsi              compressed size 1800
obqulp0              compressed size 496
obqulp1              compressed size 456
obqulp2              compressed size 88
obqulp3              compressed size 88
obqsge               compressed size 1372
obqncsi              compressed size 88
rss                  compressed size 100
rss_pf_config        compressed size 76
rss_key              compressed size 96
rss_vf_config        compressed size 116
rss_config           compressed size 84
pathmtu              compressed size 88
swstate              compressed size 80
wtp                  compressed size 328
pmstats              compressed size 140
hwsched              compressed size 72
tcpstats             compressed size 64
tperrstats           compressed size 68
fcoestats            compressed size 64
rdmastats            compressed size 64
tpindirect           compressed size 744
sgeindirect          compressed size 200
```

- **extract**

Description: Extract the compressed debug log in a human readable format to the specified path.

Syntax: `cxgbtool <vbdIface> cudbg extract <entity> <inputFile> <dir>`

Examples:

- Extracting debug log for a specific entity to a specified path.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg extract meminfo dump_file cudbg_log
cudbg_view() dbg entity : meminfo

Debug logs extracted to cudbg_log

C:\Users\Administrator>type cudbg_log\debug_1\meminfo.txt
EDC0:           0-0x3fffff [4.00 MiB]
EDC1:           0x400000-0x7fffff [4.00 MiB]
MC:             0x800000-0x407fffff [1.00 GiB]
RQUDP region:   0xfffffff0-0xfffffffe [0 B]
IMSG contexts: 0x271880-0x32b87f [744 KiB]
ULPTX state:    0x32b880-0x33527f [38.5 KiB]
ULPRX state:    0x335280-0x339e7f [19.0 KiB]
Pstructs:       0x339e80-0x3b967f [510 KiB]
Rx FL:          0x3b9680-0x3baabf [5.06 KiB]
Tx FL:          0x3baac0-0x3beabf [16.0 KiB]
Pstruct FL:     0x3beac0-0x3c3fff [21.3 KiB]
LE hash:        0x3c4000-0x403fff [256 KiB]
TCBs:           0x404000-0x66ffff [2.42 MiB]
Rx payload:     0x670000-0x7fefff [121 MiB]
DBQ contexts:  0x8180000-0x8239fff [744 KiB]
FLM cache:      0x823a000-0x835147f [1.09 MiB]
Timers:         0x8351480-0x87fffff [4.68 MiB]
TDDP region:    0x8800000-0x8f877ff [7.52 MiB]
iSCSI region:   0x8f87800-0x93877ff [4.00 MiB]
TPT region:     0x9387800-0xb92d1ff [37.6 MiB]
STAG region:    0x9387800-0xb92d1ff [37.6 MiB]
TXPBL region:   0xb92d200-0x1b92d1ff [256 MiB]
PBL region:     0xb92d200-0x1b92d1ff [256 MiB]
RQ region:      0x1b92d200-0x2047867f [75.2 MiB]
Tx payload:     0x20800000-0x387fffff [384 MiB]
uP RAM:         0x396f4000-0x407fffff [113 MiB]
uP Extmem2:     0-0xffffffff [0 B]
```

- Extracting debug log to a specified path skipping specific entities.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg extract all dump_file cudbg_log skip cimla,mc0,mc1,swstate,maindirect,tpla,rss,letcam,dumpcontext,edc0,edc1
cudbg_view() dbg entity : regdump
cudbg_view() dbg entity : devlog
cudbg_view() dbg entity : cimmala
cudbg_view() dbg entity : cimqcfg
cudbg_view() dbg entity : ibqtp0
cudbg_view() dbg entity : ibqtp1
cudbg_view() dbg entity : ibqulp
cudbg_view() dbg entity : ibqsge0
cudbg_view() dbg entity : ibqsge1
cudbg_view() dbg entity : ibqncsi
cudbg_view() dbg entity : obqulp0
cudbg_view() dbg entity : obqulp1
cudbg_view() dbg entity : obqulp2
cudbg_view() dbg entity : obqulp3
cudbg_view() dbg entity : obqsge
cudbg_view() dbg entity : obqncsi
cudbg_view() dbg entity : rss_pf_config
cudbg_view() dbg entity : rss_key
cudbg_view() dbg entity : rss_vf_config
cudbg_view() dbg entity : rss_config
cudbg_view() dbg entity : pathmtu
cudbg_view() dbg entity : wtp
cudbg_view() dbg entity : pmstats
cudbg_view() dbg entity : hwsched
cudbg_view() dbg entity : tcpstats
cudbg_view() dbg entity : tperrstats
cudbg_view() dbg entity : fcoestats
cudbg_view() dbg entity : rdmastats
cudbg_view() dbg entity : tpindirect
cudbg_view() dbg entity : sgeindirect
cudbg_view() dbg entity : cplstats
cudbg_view() dbg entity : ddpstats
cudbg_view() dbg entity : wcstats
cudbg_view() dbg entity : ulprxla
cudbg_view() dbg entity : lbstats
cudbg_view() dbg entity : meminfo
cudbg_view() dbg entity : cimpifla
cudbg_view() dbg entity : clk
cudbg_view() dbg entity : obq_sge_rx_q0
cudbg_view() dbg entity : obq_sge_rx_q1
cudbg_view() dbg entity : macstats
cudbg_view() dbg entity : pcieindirect
cudbg_view() dbg entity : pmindirect
cudbg_view() dbg entity : full
cudbg_view() dbg entity : txrate
cudbg_view() dbg entity : tidinfo
```

- **dbg**

Description: Display debug log onscreen without storing it in any file.

Syntax: `cxgbtool <vbdIface> cudbg dbg <entity>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg full
cudbg_view() dbg entity : full

Tx0 ==0=>  T  <=0= Rx0
Tx1 ==0=>  P  <=0= Rx1

Tx0 P =0=> S ? U =>0=>  T
Tx1 C =0=> G ? T =>0=>  P
       Rd Wr
RX0 P <=0=0=0 S <=0= C <=0= T <=T <=0=  T <=0= M
RX1 C <=0=0=0 G <=0= X <=0= C <=P <=0=  E <=0= P
```

- **tcb**

Description: Read hardware TCP Control Block, which contains details regarding all offloaded connections.

Syntax: `cxgbtool <vbdIface> cudbg tcb <options>`

Examples:

- Displaying tcb information for a given tid.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg tcb -t 5050
TID Number 5050
  STATE:
    CLOSED       (0 ), IPv4, lock_tid 0, rss_fw 0
    l2t_ix 0x0, smac sel 0x0, tos 0x0
    maxseg 0, recv_scaleflag 0, recv_tstmp 0, recv_sack 0
  TIMERS:
    timer    0,  dack_timer      0
    mod_schd: tx: 0, rx: 0, reason 0x0
    max_rt   0 , rxtshift        0, keepalive   0
    timestamp_offset 0x0,  timestamp 0x0
    t_rtt_ts_recent_age 0  t_rttseq_recent 0
    t_srtt 0, t_rttvar 0
  TRANSMIT BUFFER:
    snd_una 0, snd_nxt 0, snd_max 0, tx_max 0
    core_fin 0, tx_hdr_offset 0
    rcv_adv    0     (rcv_scale 0  recv_scaleflag 0 active_open 0)
    snd_cwnd   0      snd_ssthresh 0  snd_rec 0
    cctrl: sel Reno, ecn 0, ece 0, cwr 0, rfr 0
    t_dupacks 0, dupack_count_odd 0, fast_recovery 0
    core_more    0, core_urg,       0  core_push   0,  core_flush 0
    nagle        0, ssws_disable    0, turbo       0,  tx_pdu_out 0
    tx_pace_auto 0, tx_pace_fixed   0, tx_queue    0   tx_quiesce 0
    tx_channel   0, tx_channel1     0, tx_channel0 0
    tx_hdr_ptr   0x0        tx_last_ptr 0x0       tx_compact 0
  RECEIVE BUFFER:
    last_ack_sent 0                               rx_compact 0
```

- Displaying list of used and retired tids.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg  tcb -b
TIDs:
        Currently used TIDs(in range[3072 - 19455])
        13144   13184   17968   18152

        Retired TIDs(in range [3072 - 19455]):
        6192    7552    13200   17952

STIDs:
        Currently used TIDs(in range[2560 - 2687])
        2560    2564    2568    2569
```

- Displaying tcb information of all active tids.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg  tcb
TIDs:
TID Number 13144
  STATE:
    ESTABLISHED  (4 ), IPv4, lock_tid 1, rss_fw 0
    l2t_ix 0x0, smac sel 0x41, tos 0x0
    maxseg 7, recv_scaleflag 1, recv_tstmp 0, recv_sack 0
  TIMERS:
    timer    1,  dack_timer      0
    mod_schd: tx: 0, rx: 0, reason 0x0
    max_rt   15, rxtshift        0, keepalive   1
    timestamp_offset 0x0,  timestamp 0x1b0
    t_rtt_ts_recent_age 285667744  t_rttseq_recent 518540076
    t_srtt 2, t_rttvar 3
  TRANSMIT BUFFER:
    snd_una 518540204, snd_nxt 518540204, snd_max 518540204, tx_max 518540204
    core_fin 0, tx_hdr_offset 0
    rcv_adv    32768 << 6  == 2097152 (recv_scaleflag 1 rcv_scale 6 active open 0)
    snd_cwnd   268435455  snd_ssthresh 268435455  snd_rec 251230524
    cctrl: sel Tahoe, ecn 0, ece 0, cwr 0, rfr 0
    t_dupacks 0, dupack_count_odd 0, fast_recovery 0
    core_more    0, core_urg,       0  core_push   0,  core_flush 0
    nagle        0, ssws_disable    0, turbo       0,  tx_pdu_out 0
    tx_pace_auto 1, tx_pace_fixed   0, tx_queue    0   tx_quiesce 0
    tx_channel   0, tx_channel1     0, tx_channel0 0
    tx_hdr_ptr   0x1ffff    tx_last_ptr 0x1ffff   tx_compact 1
  RECEIVE BUFFER:
    last_ack_sent 504824648                       rx_compact 1
    rcv_nxt       504824648   hdr_off 0
    frag0_idx     536898476   length  0           frag0_ptr  0x1ffff
    frag1_idx     568972304   length  0
    peer_fin 0,   rx_pdu_out 0, pdu_len 0
    rcv_wnd 2097152 >> snd_scale 6 == 32768, recv_scaleflag = 1
    dack_mss   0 dack       0,  dack_not_acked: 0
    rcv_coal   0 rcv_co_psh 0 rcv_co_last_psh  1 heart 0
    rx_channel 0 rx_quiesce 0 rx_flow_ctrl_dis 1,  rx_flow_ctrl_ddp 0
  MISCELANEOUS:
    pend_ctl: 0x0, core_bypass: 0x0,  main_slush: 0x40000000
    Migrating 0, ask_mode 1, non_offload 0, rss_info 1
    ULP: ulp_type 4 (RDMA), ulp_raw 3
    RDMA: error   0, flm_err 0
      qp_id 1030, pd_id 6, stag 0
      irs_ulp 456, iss_ulp 304
      tx_pdu_len 0
      cq_idx_sq 1031, cq_idx_rq 1030
      rq_start 1024, rq_MSN 936, rq_max_off 9, rq_write_ptr 168
      L_valid 1, rdmap opcode 3
      tx_flush: 0, tx_oos_rxmt 0, tx_oos_txmt 0
```

- Displaying tcb information for given tid from a dump file.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg  tcb -f dump_file2 -t 5050
TID Number 5050
  STATE:
    CLOSED       (0 ), IPv4, lock_tid 0, rss_fw 0
    l2t_ix 0x0, smac sel 0x0, tos 0x0
    maxseg 0, recv_scaleflag 0, recv_tstmp 0, recv_sack 0
  TIMERS:
    timer    0,  dack_timer      0
    mod_schd: tx: 0, rx: 0, reason 0x0
    max_rt   0 , rxtshift        0, keepalive   0
    timestamp_offset 0x0,  timestamp 0x0
    t_rtt_ts_recent_age 0  t_rttseq_recent 0
    t_srtt 0, t_rttvar 0
  TRANSMIT BUFFER:
    snd_una 0, snd_nxt 0, snd_max 0, tx_max 0
    core_fin 0, tx_hdr_offset 0
    rcv_adv    0     (rcv_scale 0  recv_scaleflag 0 active_open 0)
    snd_cwnd   0      snd_ssthresh 0  snd_rec 0
    cctrl: sel Reno, ecn 0, ece 0, cwr 0, rfr 0
    t_dupacks 0, dupack_count_odd 0, fast_recovery 0
    core_more    0, core_urg,       0  core_push   0,  core_flush 0
    nagle        0, ssws_disable    0, turbo       0,  tx_pdu_out 0
    tx_pace_auto 0, tx_pace_fixed   0, tx_queue    0   tx_quiesce 0
    tx_channel   0, tx_channel1     0, tx_channel0 0
    tx_hdr_ptr   0x0        tx_last_ptr 0x0       tx_compact 0
  RECEIVE BUFFER:
    last_ack_sent 0                                 rx_compact 0
    rcv_nxt       0           hdr_off 0
    frag0_idx     0           length  0           frag0_ptr  0x0
    frag1_idx     0           length  0           frag1_ptr  0x0
    frag2_idx     0           length  0           frag2_ptr  0x0
    frag3_idx     0           length  0           frag3_ptr  0x0
    peer_fin 0,   rx_pdu_out 0, pdu_len 0
    rcv_wnd 0.  (snd_scale 0, recv_scaleflag = 0)
    dack_mss   0 dack       0,  dack_not_acked: 0
    rcv_coal   0 rcv_co_psh 0 rcv_co_last_psh  0 heart 0
    rx_channel 0 rx_quiesce 0 rx_flow_ctrl_dis 0,  rx_flow_ctrl_ddp 0
  MISCELANEOUS:
    pend_ctl: 0x0, core_bypass: 0x0,  main_slush: 0x0
    Migrating 0, ask_mode 0, non_offload 0, rss_info 0
    ULP: ulp_type 0 (TOE), ulp_raw 0,  ulp_ext 0
    RDMA: error   0, flm_err 0
      aux1_slush0: 0x0 aux1_slush1 0x0
      pdu_hdr_len 0
```

- Displaying list of used and retired tids from a dump file.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg tcb -f dump_file -b
TIDs:
        Currently used TIDs(in range[3072 - 19455])
        13168   13240   17928   18112

        Retired TIDs (in range [3072 - 19455]):
        3352    3360    3400    3416    3424    3440    3456    3472    3496    3512
        3520    4864    4880    4904    4920    4928    4968    4984    5000    5024
        5040    5080    5088    5752    6176    6312    7432    7512    9176    9728
        9744    9784    9808    9832    9864    9888    9904    9928    9944    9968
        10248   10264   10272   10288   10312   10352   10384   10408   10432   10448
        10472   10488   11576   11736   11744   13080   13088   13248   13304   13832
        13848   13856   13872   13896   13912   13920   13936   13952   13968   14008
        14016   14032   14056   14336   14352   14376   14392   14400   14416   14440
        14472   14488   14496   14512   14536   14552   14560   14576   15616   15632
        15656   15672   15680   15696   15720   15736   15752   15768   15776   15792
        15816   15840   15856   17176   17184   17200   17224   17240   17248   17264
        17296   17320   17336   17344   17360   17384   17400   17992   18008   18032
        18064   18088   18512   18536

STIDs:
        Currently used TIDs(in range[2560 - 2687])
        2560    2564    2568    2569
```

- Displaying tcb information for all active tids from a dump file.

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg tcb -f dump_file
TIDs:
TID Number 13168
  STATE:
    ESTABLISHED  (4 ), IPv4, lock_tid 1, rss_fw 0
    l2t_ix 0x1, smac sel 0x41, tos 0x0
    maxseg 7, recv_scaleflag 1, recv_tstmp 0, recv_sack 0
  TIMERS:
    timer    1,  dack_timer      0
    mod_schd: tx: 0, rx: 0, reason 0x0
    max_rt   15, rxtshift        0, keepalive   1
    timestamp_offset 0x0,  timestamp 0x1b9
    t_rtt_ts_recent_age 1645396424  t_rttseq_recent 2183343868
    t_srtt 2, t_rttvar 3
  TRANSMIT BUFFER:
    snd_una 2183345212, snd_nxt 2183345212, snd_max 2183345212, tx_max 2183345212
    core_fin 0, tx_hdr_offset 0
    rcv_adv    32768 << 6  == 2097152 (recv_scaleflag 1 rcv_scale 6 active open 0)
    snd_cwnd   268435455  snd_ssthresh 268435455  snd_rec 2140287142
    cctrl: sel Tahoe, ecn 0, ece 0, cwr 0, rfr 0
    t_dupacks 0, dupack_count_odd 0, fast_recovery 0
    core_more    0, core_urg,       0  core_push   0,  core_flush 0
    nagle        0, ssws_disable    0, turbo       0,  tx_pdu_out 0
    tx_pace_auto 1, tx_pace_fixed   0, tx_queue    1   tx_quiesce 0
    tx_channel   1, tx_channel1     0, tx_channel0 1
    tx_hdr_ptr   0x1ffff    tx_last_ptr 0x1ffff   tx_compact 1
  RECEIVE BUFFER:
    last_ack_sent 2210614520                      rx_compact 1
    rcv_nxt       2210614520  hdr_off 0
    frag0_idx     2415939764  length  0           frag0_ptr  0x1ffff
    frag1_idx     2621265008  length  0
    peer_fin 0,   rx_pdu_out 0, pdu_len 0
    rcv_wnd 2097152 >> snd_scale 6 == 32768, recv_scaleflag = 1
    dack_mss   0 dack       0,  dack_not_acked: 0
    rcv_coal   0 rcv_co_psh 0 rcv_co_last_psh  1 heart 0
    rx_channel 0 rx_quiesce 0 rx_flow_ctrl_dis 1,  rx_flow_ctrl_ddp 0
  MISCELANEOUS:
    pend_ctl: 0x0, core_bypass: 0x0,  main_slush: 0x40000000
```

- **ps**

Description: Display pstruct information.

Syntax:

- Display pstruct info for given pstruct number:

```
cxgbtool <vbdIface> cudbg ps -t <psIndex>
```

- Display pstruct info of range for given index:

```
cxgbtool <vbdIface> cudbg ps -t <startIndex> <endIndex>
```

- Collect pstruct info from given dump file:

```
cxgbtool <vbdIface> cudbg ps -t <psIndex> -f <file>
```

Examples:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg ps -t 6
All region are zeros

C:\Users\Administrator>cxgbtool vbd0 cudbg ps -t 6 8
All region are zeros

C:\Users\Administrator>cxgbtool vbd0 cudbg ps -t 6 -f dump_file
All region are zeros
```

- **tddp**

Description: Displays TDDP region information.

Syntax:

- Display TDDP region info for given index:

```
cxgbtool <vbdIface> cudbg tddp -t <tddpIndex>
```

- Display TDDP region info of range for given index:

```
cxgbtool <vbdIface> cudbg tddp -t <startIndex> <endIndex>
```

- Collect tddp info from given dump file:

```
cxgbtool <vbdIface> cudbg tddp -t <tddpIndex> -f <file>
```

Examples:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg tddp -t 6
All region are zeros

C:\Users\Administrator>cxgbtool vbd0 cudbg tddp -t 6 8
All region are zeros

C:\Users\Administrator>cxgbtool vbd0 cudbg tddp -t 6 -f dump_file
All region are zeros
```

cudbg debug entities

- **all**

Description: Collect/Display debug logs for all cudbg entities

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract> all <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect all dump_file
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file dump_file... size: 12634452 bytes
cxgbtool: Done writing cudbg data to file dump_file
```

- **ibqtp0, ibqtp1**

Description: Collect/Display CIM TP inbound queue.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> ibqtp[0|1] <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg ibqtp1
cudbg_view() dbg entity : ibqtp1
000000: 00000000 585bd8c4 00000000 2635e313
0x0010: 00000000 64d43b4c 00000000 30bc7dab
0x0020: 00000000 b62c807e 00000000 1bc764f4
0x0030: 00000000 e1635d82 00000000 02e0cc05
0x0040: 00000000 0786d541 00000000 db52c45b
0x0050: 00000000 bbd2a7d8 00000000 f98d23e3
0x0060: 00000000 e5eb3829 00000000 ed1892c2
0x0070: 00000000 aa3fbcd5 00000000 7efb1c73
0x0080: 00000000 fae5a9c1 00000000 c8b3b4f8
0x0090: 00000000 f0926d2a 00000000 443cfd95
0x00a0: 00000000 fb587d8b 00000000 82517038
0x00b0: 00000000 44b0b7f6 00000000 9536ce5d
0x00c0: 00000000 602e0788 00000000 5d072471
```

- **obqulp0, obqulp1, obqulp2, obqulp3**

Description: Collect/Display ULP outbound queue.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> obqulp[0|1|2|3] <file>
<dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg extract obqulp0 obqulp0_log C:\Users\Administrator\Desktop\obqulp0_dump
cudbg_view() dbg entity : obqulp0

Debug logs extracted to C:\Users\Administrator\Desktop\obqulp0_dump

C:\Users\Administrator>type C:\Users\Administrator\Desktop\obqulp0_dump\debug_1\obqulp0.txt
000000: 0180c200 000e0007 4304b397 88cc0207
0x0010: 04000743 04b39704 07030007 4304b397
0x0020: 06020078 fe190080 c2098000 01000032
0x0030: 32000000 00000002 02020202 020202fe
0x0040: 060080c2 0b8808fe 050080c2 0c000000
0x0050: 04000000 00992a07 81000000 00000060
0x0060: 0e0007e2 00000050 c0000000 00000000
0x0070: 0180c200 000e0007 4304b397 88cc0207
0x0080: 04000743 04b39704 07030007 4304b397
0x0090: 06020078 fe190080 c2098000 01000032
0x00a0: 32000000 00000002 02020202 020202fe
0x00b0: 060080c2 0b8808fe 050080c2 0c000000
0x00c0: 04000000 00992a07 81000000 00000060
0x00d0: 0e0007e2 00000050 c0000000 00000000
0x00e0: 0180c200 000e0007 4304b397 88cc0207
0x00f0: 04000743 04b39704 07030007 4304b397
```

- **edc0, edc1**

Description: Collect/Display EDC memory details.

Syntax: `cxgbtool <vbdIface> cudbg [collect|extract] edc[0|1] <file> <dir>`

Note *It is recommended that the file be provided without any extension.*

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect edc0 edc0_log
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file edc0_log... size: 557476 bytes
cxgbtool: Done writing cudbg data to file edc0_log
```

- **rss_key**

Description: Collect/Display RSS Key.

Syntax:

`cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> rss_key <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg rss_key
cudbg_view() dbg entity : rss_key
9c28c016ae7acf821e81ce345e70ec69eb194a76eaebc7f7fdae657b006b5e731d23fc4caee6ba81
```

- **pmstats**

Description: Collect/Display Page memory statistics.

Syntax:

`cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> pmstats <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect pmstats log_pmstats
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file log_pmstats... size: 3768 bytes
cxgbtool: Done writing cudbg data to file log_pmstats

C:\Users\Administrator>cxgbtool vbd0 cudbg view pmstats log_pmstats
cudbg_view() dbg entity : pmstats
                Tx pcmds              Tx bytes
Read:                  2                     2
Write bypass:       6082               4384876
Write mem:             0                     0
Bypass + mem:     252126             336115740
                Rx pcmds              Rx bytes
Read:                  0                     0
Write bypass:     254793             335348101
Write mem:             0                     0
Flush:            491268              28987578
              Total wait       Total Occupancy
Tx FIFO wait        1820            2410484836
Rx FIFO wait        1179            2555337154
                   Reads            Total wait
Tx latency             2                   136
Rx latency             0                     0
```

- **tpindirect**

Description: Collect/Display TP indirect registers.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> tpindirect <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg tpindirect
cudbg_view() dbg entity : tpindirect


TP_PIO

[0x07e40:0x00020] TP_RX_SCHED_MAP                          0x15151515     353703189
    31:24 S_RXMAPCHANNEL3                                  0x15           21
    23:16 S_RXMAPCHANNEL2                                  0x15           21
     15:8 S_RXMAPCHANNEL1                                  0x15           21
      7:0 S_RXMAPCHANNEL0                                  0x15           21
[0x07e40:0x00021] TP_RX_SCHED_SGE                          0xf            15
    15:12 S_RXSGEMOD1                                      0              0
     11:8 S_RXSGEMOD0                                      0              0
      3:3 S_RXSGECHANNEL3                                  0x1            1
      2:2 S_RXSGECHANNEL2                                  0x1            1
      1:1 S_RXSGECHANNEL1                                  0x1            1
      0:0 S_RXSGECHANNEL0                                  0x1            1
[0x07e40:0x00022] TP_TX_SCHED_MAP                          0x7777         30583
    15:12 S_TXMAPCHANNEL3                                  0x7            7
     11:8 S_TXMAPCHANNEL2                                  0x7            7
      7:4 S_TXMAPCHANNEL1                                  0x7            7
      3:0 S_TXMAPCHANNEL0                                  0x7            7
[0x07e40:0x00023] TP_TX_SCHED_HDR                          0x21212121     555819297
    31:28 S_TXMAPHDRCHANNEL7                               0x2            2
    27:24 S_TXMAPHDRCHANNEL6                               0x1            1
    23:20 S_TXMAPHDRCHANNEL5                               0x2            2
    19:16 S_TXMAPHDRCHANNEL4                               0x1            1
    15:12 S_TXMAPHDRCHANNEL3                               0x2            2
     11:8 S_TXMAPHDRCHANNEL2                               0x1            1
      7:4 S_TXMAPHDRCHANNEL1                               0x2            2
      3:0 S_TXMAPHDRCHANNEL0                               0x1            1
```

- **lbstats**

Description: Collect/Display loopback statistics.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> lbstats <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg lbstats
cudbg_view() dbg entity : lbstats
                        Loopback 0          Loopback 1
OctetsOK:                           0                   0
FramesOK:                           0                   0
BcastFrames:                        0                   0
McastFrames:                        0                   0
UcastFrames:                        0                   0
ErrorFrames:                        0                   0
Frames64:                           0                   0
Frames65To127:                      0                   0
Frames128To255:                     0                   0
Frames256To511:                     0                   0
Frames512To1023:                    0                   0
Frames1024To1518:                   0                   0
Frames1519ToMax:                    0                   0
FramesDropped:                      0                   0
BG0FramesDropped:                   0                   0
BG1FramesDropped:                   0                   0
BG2FramesDropped:                   0                   0
BG3FramesDropped:                   0                   0
BG0FramesTrunc:                     0                   0
BG1FramesTrunc:                     0                   0
BG2FramesTrunc:                     0                   0
BG3FramesTrunc:                     0                   0
```

- **obq_sge_rx_q0, obq_sge_rx_q1**

Description: Collect/Display CIM SGE outbound queue.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> obq_sge_rx_q[0|1] <file>
<dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect obq_sge_rx_q1 log_obq1
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file log_obq1... size: 5536 bytes
cxgbtool: Done writing cudbg data to file log_obq1

C:\Users\Administrator>cxgbtool vbd0 cudbg view obq_sge_rx_q1 log_obq1
cudbg_view() dbg entity : obq_sge_rx_q1
000000: 8bd4ab13 8a9c760c 5fc4cb0c ba602958
0x0010: 5b0b3a2e 7128dbb1 59a86bbf 5fb76a36
0x0020: caa71e0d 4c529e57 ef1856bf 72c1b994
0x0030: 05e13fbb abf4790f b2ceb331 55752b17
0x0040: 06f1cca5 a4a7e487 03070ef6 593fd343
0x0050: b27508c3 3398f9fe c9fb3a3f c4764f6e
0x0060: b05d6e72 c27f2720 753d30c3 90b77fbe
0x0070: 64c942d8 5c5e12b7 95bb3f52 77df3f9f
0x0080: 2e599dfa 4a92e017 73c6ce8d e21b9cfd
0x0090: 674eeb9e c341ef7c 0a382eb3 4e88aa37
0x00a0: 5d24e302 09fb265e 7f09f610 bac48061
0x00b0: 4b9874a7 6b3ab537 0e65a4bf 6ba1999d
0x00c0: 271f051e a00473ab 289c8116 02b2140c
0x00d0: ac54bf88 8a9644fe becee46a 7046bcd4
0x00e0: 23dcf548 a31f7fa3 55c60c60 b5167208
0x00f0: cea8274b 37055cb4 e9c91678 6e6b94b9
```

- **tidinfo**

Description: Collect/Display TID information

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> tidinfo <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg tidinfo
cudbg_view() dbg entity : tidinfo


        TID INFO

TID range: 0..2559/3072..19455
STID range: 2560..2687
HW TID usage: 0 IP users, 0 IPv6 users
```

- **cctrl**

Description: Collect/Display Congestion control table.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> cctrl <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbdo cudbg dbg cctrl
cudbg_view() dbg entity : cctrl
 0:    24  108  236  268  384  492  620  724
     730  981 1004 2028 2156 4076 4480 4780
 1:     8   36   78   89  128  164  206  241
     243  327  334  676  718 1358 1493 1593
 2:     4   21   47   53   76   98  124  144
     146  196  200  405  431  815  896  956
 3:     3   15   33   38   54   70   88  103
     104  140  143  289  308  582  640  682
 4:     2   10   23   26   38   49   62   72
      73   98  100  202  215  407  448  478
 5:     2    7   16   19   27   35   44   51
      52   70   71  144  154  291  320  341
 6:     2    5   11   13   19   24   31   36
      36   49   50  101  107  203  224  239
 7:     2    3    8    9   13   17   22   25
      26   35   35   72   77  145  160  170
 8:     2    2    5    6    9   12   15   18
      18   24   25   50   53  101  112  119
```

- **mboxlog**

Description: Collect/Display firmware mailbox command/reply log information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> mboxlog <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect mboxlog log_mbox
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file log_mbox... size: 11080 bytes
cxgbtool: Done writing cudbg data to file log_mbox

C:\Users\Administrator>cxgbtool vbd0 cudbg view mboxlog log_mbox
cudbg_view() dbg entity : mboxlog
      Seq          Tstamp  Atime  Etime  Command/Reply
      3374        12225642      0      0  01c00010 00000004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3375        12225643      0      1  01000010 00560004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3376        12225643      0      0  01c00010 00000004  0000003b 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3377        12225643      0      1  01000010 00560004  0000003b 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3378        12225643      0      0  01c00010 00000004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3379        12225643      0      1  01000010 00560004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3380        12225643      0      0  01c00010 00000004  0000003b 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3381        12225643      0      1  01000010 005a0004  0000003b 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3382        12225643      0      0  01c00010 00000004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
0000000 00000000
      3383        12225643      0      1  01000010 00520004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
```

- **regdump**

Description: Collect/Display hardware module registers.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> regdump <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg regdump
cudbg_view() dbg entity : regdump
[0x1e000] SGE_PF_KDOORBELL                          0          0
    31:15 QID                                       0          0
    14:14 Sync                                      0          0
    13:13 Type                                      0          0
     12:0 PIDX                                      0          0
[0x1e004] SGE_PF_GTS                                0          0
    31:16 IngressQID                                0          0
    15:13 TimerReg                                  0          0
    12:12 SEIntArm                                  0          0
     11:0 CIDXInc                                   0          0
[0x1e008] SGE_PF_KTIMESTAMP_LO                      0x7467a7f6 1952950262
[0x1e00c] SGE_PF_KTIMESTAMP_HI                      0x3ac3     15043
[0x1e400] SGE_PF_KDOORBELL                          0          0
    31:15 QID                                       0          0
    14:14 Sync                                      0          0
    13:13 Type                                      0          0
     12:0 PIDX                                      0          0
```

- **rss_vf_config**

Description: Collect/Display RSS VF Configuration.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> rss_vf_config <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect rss_vf_config  log_rssvfconfig
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file log_rssvfconfig... size: 3696 bytes
cxgbtool: Done writing cudbg data to file log_rssvfconfig

C:\Users\Administrator>cxgbtool vbd0 cudbg view rss_vf_config  log_rssvfconfig
cudbg_view() dbg entity : rss_vf_config
     RSS                    Hash Tuple Enable
     Enable   IVF  Dis  Enb  IPv6      IPv4      UDP    Def  Secret Key
 VF  Chn Prt  Map  VLAN  uP  Four Two  Four Two  Four   Que  Idx       Hash
  0   no  no    0    no  no    no  no    no  no    no     0    0          0
  1   no  no    0    no  no    no  no    no  no    no     0    0          0
  2   no  no    0    no  no    no  no    no  no    no     0    0          0
  3   no  no    0    no  no    no  no    no  no    no     0    0          0
  4   no  no    0    no  no    no  no    no  no    no     0    0          0
  5   no  no    0    no  no    no  no    no  no    no     0    0          0
  6   no  no    0    no  no    no  no    no  no    no     0    0          0
  7   no  no    0    no  no    no  no    no  no    no     0    0          0
  8   no  no    0    no  no    no  no    no  no    no     0    0          0
  9   no  no    0    no  no    no  no    no  no    no     0    0          0
 10   no  no    0    no  no    no  no    no  no    no     0    0          0
 11   no  no    0    no  no    no  no    no  no    no     0    0          0
 12   no  no    0    no  no    no  no    no  no    no     0    0          0
```

- **hwsched**

Description: Collect/Display hardware scheduler information

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> hwsched <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg extract hwsched  log_hwsched C:\Users\Administrator\Desktop\dump_hwsched
cudbg_view() dbg entity : hwsched

Debug logs extracted to C:\Users\Administrator\Desktop\dump_hwsched

C:\Users\Administrator>type Desktop\dump_hwsched\debug_1\hwsched.txt
Scheduler  Mode   Channel  Rate (Kbps)   Class IPG (0.1 ns)   Flow IPG (us)
    0      class     0      33502731          disabled          disabled
    1      class     1      33502731          disabled          disabled
    2      class     0      33502731          disabled          disabled
    3      class     1      33502731          disabled          disabled
    4      class     0      33502731          disabled          disabled
    5      class     0      33502731          disabled          disabled
    6      class     0      33502731          disabled          disabled
    7      class     0      33502731          disabled          disabled
```

- **sgeindirect**

Description: Displays sge debug indirect registers.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> sgeindirect <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg sgeindirect
cudbg_view() dbg entity : sgeindirect

[0x10cc:0x1280]  SGE_DEBUG_DATA_HIGH_INDEX_0                0x10cc     4300
[0x10cc:0x1284]  SGE_DEBUG_DATA_HIGH_INDEX_1                0x10d0     4304
[0x10cc:0x1288]  SGE_DEBUG_DATA_HIGH_INDEX_2                0          0
[0x10cc:0x128c]  SGE_DEBUG_DATA_HIGH_INDEX_3                0x10       16
[0x10cc:0x1290]  SGE_DEBUG_DATA_HIGH_INDEX_4                0x990000bb 2566914235
[0x10cc:0x1294]  SGE_DEBUG_DATA_HIGH_INDEX_5                0x220055   2228309
[0x10cc:0x1298]  SGE_DEBUG_DATA_HIGH_INDEX_6                0          0
[0x10cc:0x129c]  SGE_DEBUG_DATA_HIGH_INDEX_7                0xf000     61440
[0x10cc:0x12a0]  SGE_DEBUG_DATA_HIGH_INDEX_8                0x2520     9504
[0x10cc:0x12a4]  SGE_DEBUG_DATA_HIGH_INDEX_9                0x8800     34816
[0x10cc:0x12a8]  SGE_DEBUG_DATA_HIGH_INDEX_10               0          0
[0x10cc:0x12ac]  SGE_DEBUG_DATA_HIGH_INDEX_11               0xffffff22 4294967074
[0x10cc:0x12b0]  SGE_DEBUG_DATA_HIGH_INDEX_12               0x200f5    131317
[0x10cc:0x12b4]  SGE_DEBUG_DATA_HIGH_INDEX_13               0x880066   8912998
[0x10cc:0x12b8]  SGE_DEBUG_DATA_HIGH_INDEX_14               0x5602c00  90188800
[0x10cc:0x12bc]  SGE_DEBUG_DATA_HIGH_INDEX_15               0          0
```

- **tpla**

Description: Collect/Display TP la information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> tpla <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg tpla
cudbg_view() dbg entity : tpla
RcfOpCodeOut: 4 State: 0 WcfState: 0 RcfOpcSrcOut: 0 CRxError: 0 ERxError: 0
        SanityFailed: 0 SpuriousMsg: 1 FlushInputMsg: 0 FlushInputCpl: 0
        RssUpBit: 0 RssFilterHit: 0 Tid: 0 InitTcb: 0 LineNumber: 0 Emsg: 0
        EdataOut: 0 Cmsg: 0 CdataOut: 0 EreadPdu: 0 CreadPdu: 0 TunnelPkt: 1
        RcfPeerFin: 0 RcfReasonOut: 0 TxCchannel: 0 RcfTxChannel: 0
        RxEchannel: 1 RcfRxChannel: 0 RcfDataOutSrdy: 0 RxDvld: 0 RxOoDvld: 0
        RxCongestion: 0 TxCongestion: 0
CplCmdIn: 0 MpsVfVld: 1 MpsPf: 4 MpsVf: 66 SynIn: 0 AckIn: 0 FinIn: 0 RstIn: 0
        DataIn: 0 DataInVld: 0 PadIn: 0 RxBufEmpty: 1 RxDdp: 0
        RxFbCongestion: 0 TxFbCongestion: 0 TxPktSumSrdy: 0 RcfUlpType: 4
        Eread: 0 Ebypass: 1 Esave: 0 Static0: 1 Cread: 0 Cbypass: 1 Csave: 0
        CPktOut: 0 RxPagePoolFull: 2 RxLpbkPkt: 0 TxLpbkPkt: 0 RxVfValid: 1
        SynLearned: 0 SetDelEntry: 0 SetInvEntry: 0 CpcmdDvld: 0 CpcmdSave: 0
        RxPstructsFull: 0 EpcmdDvld: 1 EpcmdFlush: 0 EpcmdTrimPrefix: 0
        EpcmdTrimPostfix: 0 ERssIp4Pkt: 0 ERssIp6Pkt: 0 ERssTcpUdpPkt: 0
        ERssFceFipPkt: 0
```

- **macstats**

Description: Collect/Display MAC statistics for all ports.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> macstats <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg macstats
cudbg_view() dbg entity : macstats

Mac 0 Stats:
tx_octets                                                    370538854
tx_frames                                                       497874
tx_bcast_frames                                                    130
tx_mcast_frames                                                   2912
tx_ucast_frames                                                 494832
tx_error_frames                                                      0
tx_frames_64                                                    244729
tx_frames_65_127                                                  3130
tx_frames_128_255                                                 5411
tx_frames_256_511                                                   19
tx_frames_512_1023                                                  57
tx_frames_1024_1518                                             244528
tx_frames_1519_max                                                   0
tx_drop                                                              0
tx_pause                                                             0
tx_ppp0                                                              0
tx_ppp1                                                              0
tx_ppp2                                                              0
```

- **pcieconfig**

Description: Collect/Display PCIe configuration space information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> pcieconfig <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg pcieconfig
cudbg_view() dbg entity : pcieconfig

                      PCIE CONFIG

[0x00000:0x00000] PCIE_DEVID_VENID                          0x64071425      1678185509
     15:0 VendorID                                          0x1425          5157
    31:16 DeviceID                                          0x6407          25607
[0x00000:0x00004] PCIE_STAT_CMD                             0x100506        1049862
      0:0 IOEnable                                          0               0
      1:1 MemEnable                                         0x1             1
      2:2 BusMasterEnable                                   0x1             1
      5:3 Rsvd1                                             0               0
      6:6 PERREnable                                        0               0
      7:7 Rsvd2                                             0               0
      8:8 SERREnable                                        0x1             1
      9:9 Rsvd3                                             0               0
    10:10 IntDisable                                        0x1             1
    18:11 Rsvd4                                             0               0
    19:19 IntStatus                                         0               0
    20:20 CapList                                           0x1             1
    23:21 Rsvd5                                             0               0
    24:24 MstDatParErr                                      0               0
    26:25 Rsvd6                                             0               0
    27:27 SigTgtAbort                                       0               0
    28:28 RcvTgtAbort                                       0               0
    29:29 RcvMstAbort                                       0               0
    30:30 SigSERR                                           0               0
    31:31 DetPERR                                           0               0
[0x00000:0x00008] PCIE_CCODE_REVID                          0x2000000       33554432
      7:0 RevisionID                                        0               0
     31:8 ClassCode                                         0x20000         131072
```

- **maindirect**

Description: Collect/Display MA indirect registers information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> maindirect <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg maindirect
cudbg_view() dbg entity : maindirect
[0x078f8:0x0a000] MA_SGE_THREAD_0_CLIENT_INTERFACE_EXTERNAL       0x48102242     1209016898
    31:31 CmdVld0                                                 0              0
    30:30 CmdRdy0                                                 0x1            1
    29:29 CmdType0                                                0              0
    28:21 CmdLen0                                                 0x40           64
     20:8 CmdAddr0                                                0x1022         4130
      7:7 WrDataVld0                                              0              0
      6:6 WrDataRdy0                                              0x1            1
      5:5 RdDataRdy0                                              0              0
      4:4 RdDataVld0                                              0              0
      3:0 RdData0                                                 0x2            2
[0x078f8:0x0a001] MA_SGE_THREAD_1_CLIENT_INTERFACE_EXTERNAL       0x48102242     1209016898
    31:31 CmdVld1                                                 0              0
    30:30 CmdRdy1                                                 0x1            1
    29:29 CmdType1                                                0              0
    28:21 CmdLen1                                                 0x40           64
     20:8 CmdAddr1                                                0x1022         4130
      7:7 WrDataVld1                                              0              0
      6:6 WrDataRdy1                                              0x1            1
      5:5 RdDataRdy1                                              0              0
      4:4 RdDataVld1                                              0              0
      3:0 RdData1                                                 0x2            2
[0x078f8:0x0a002] MA_ULP_TX_CLIENT_INTERFACE_EXTERNAL             0x40337f64     1077116772
    31:31 CmdVld2                                                 0              0
    30:30 CmdRdy2                                                 0x1            1
    29:29 CmdType2                                                0              0
    28:21 CmdLen2                                                 0x1            1
     20:8 CmdAddr2                                                0x137f         4991
      7:7 WrDataVld2                                              0              0
      6:6 WrDataRdy2                                              0x1            1
      5:5 RdDataRdy2                                              0x1            1
      4:4 RdDataVld2                                              0              0
      3:0 RdData2                                                 0x4            4
```

- **hmaindirect**

Description: Collect/Display HMA indirect registers information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> hmaindirect <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect hmaindirect log_hmaindirect
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file log_hmaindirect... size: 3760 bytes
cxgbtool: Done writing cudbg data to file log_hmaindirect

C:\Users\Administrator>cxgbtool vbd0 cudbg view hmaindirect log_hmaindirect
cudbg_view() dbg entity : hmaindirect
[0x51320:0x0a000] HMAT6_DEBUG_FSM_0                         0x48555       296277
    22:18 edc_fsm                                           0x1           1
    17:15 ras_fsm_slv                                       0x1           1
    14:10 fc_fsm                                            0x1           1
      9:8 cookie_arb_fsm                                    0x1           1
      7:6 pcie_chunk_fsm                                    0x1           1
      5:4 wtransfer_fsm                                     0x1           1
      3:2 wd_fsm                                            0x1           1
      1:0 rd_fsm                                            0x1           1
[0x51320:0x0a001] HMAT6_DEBUG_FSM_1                         0xa21         2593
    20:11 sync_fsm                                          0x1           1
     10:9 ochk_fsm                                          0x1           1
      8:5 tlb_fsm                                           0x1           1
      4:0 pio_fsm                                           0x1           1
[0x51320:0x0a002] HMAT6_DEBUG_PCIE_INTF                     0x4200852     69208146
    28:28 H_ReqVld                                          0             0
    27:27 H_ReqFull                                         0             0
    26:26 H_ReqSOP                                          0x1           1
    25:25 H_ReqEOP                                          0             0
    24:24 H_RspVld                                          0             0
    23:23 H_RspFull                                         0             0
    22:22 H_RspSOP                                          0             0
    21:21 H_RspEOP                                          0x1           1
    20:20 H_RspErr                                          0             0
    19:19 pcie_cmd_avail                                    0             0
    18:18 pcie_cmd_rdy                                      0             0
    17:17 pcie_wnr                                          0             0
     16:9 pcie_len                                          0x4           4
      8:8 pcie_trwdat_rdy                                   0             0
      7:7 pcie_trwdat_avail                                 0             0
```

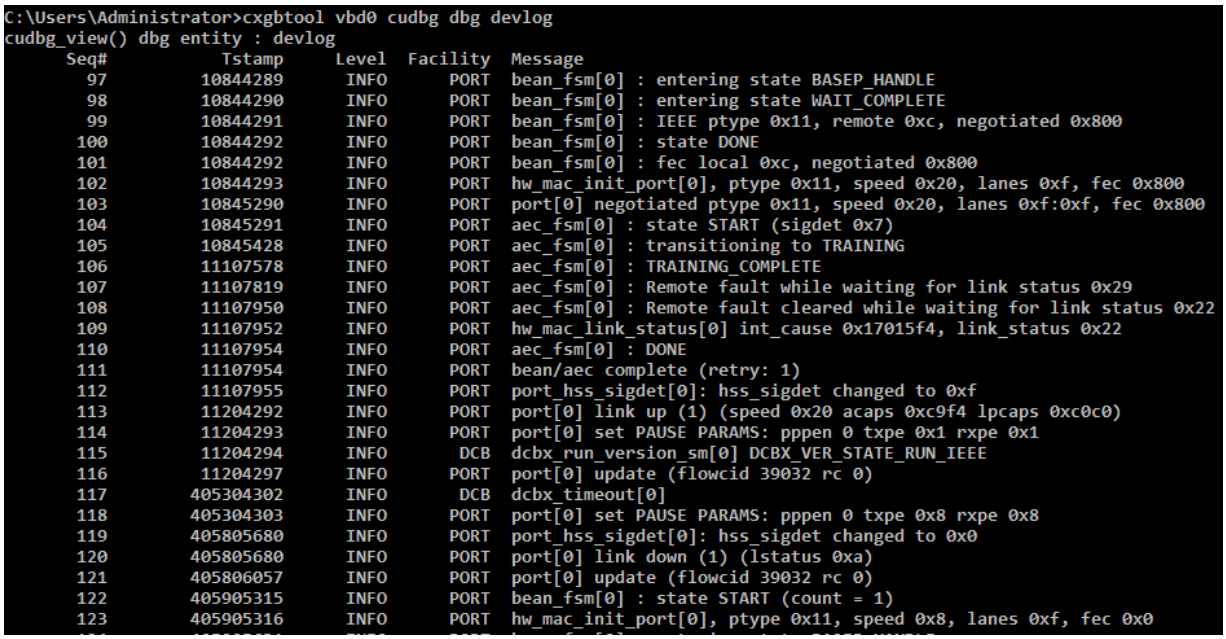
- **devlog**

Description: Collect/Display firmware device log information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> devlog <file> <dir>
```

Example:



```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg devlog
cudbg_view() dbg entity : devlog
      Seq#           Tstamp     Level  Facility  Message
        97         10844289      INFO      PORT  bean_fsm[0] : entering state BASEP_HANDLE
        98         10844290      INFO      PORT  bean_fsm[0] : entering state WAIT_COMPLETE
        99         10844291      INFO      PORT  bean_fsm[0] : IEEE ptype 0x11, remote 0xc, negotiated 0x800
       100         10844292      INFO      PORT  bean_fsm[0] : state DONE
       101         10844292      INFO      PORT  bean_fsm[0] : fec local 0xc, negotiated 0x800
       102         10844293      INFO      PORT  hw_mac_init_port[0], ptype 0x11, speed 0x20, lanes 0xf, fec 0x800
       103         10845290      INFO      PORT  port[0] negotiated ptype 0x11, speed 0x20, lanes 0xf:0xf, fec 0x800
       104         10845291      INFO      PORT  aec_fsm[0] : state START (sigdet 0x7)
       105         10845428      INFO      PORT  aec_fsm[0] : transitioning to TRAINING
       106         11107578      INFO      PORT  aec_fsm[0] : TRAINING_COMPLETE
       107         11107819      INFO      PORT  aec_fsm[0] : Remote fault while waiting for link status 0x29
       108         11107950      INFO      PORT  aec_fsm[0] : Remote fault cleared while waiting for link status 0x22
       109         11107952      INFO      PORT  hw_mac_link_status[0] int_cause 0x17015f4, link_status 0x22
       110         11107954      INFO      PORT  aec_fsm[0] : DONE
       111         11107954      INFO      PORT  bean/aec complete (retry: 1)
       112         11107955      INFO      PORT  port_hss_sigdet[0]: hss_sigdet changed to 0xf
       113         11204292      INFO      PORT  port[0] link up (1) (speed 0x20 acaps 0xc9f4 lpcaps 0xc0c0)
       114         11204293      INFO      PORT  port[0] set PAUSE PARAMS: pppen 0 txpe 0x1 rxpe 0x1
       115         11204294      INFO       DCB  dcbx_run_version_sm[0] DCBX_VER_STATE_RUN_IEEE
       116         11204297      INFO      PORT  port[0] update (flowcid 39032 rc 0)
       117        405304302      INFO       DCB  dcbx_timeout[0]
       118        405304303      INFO      PORT  port[0] set PAUSE PARAMS: pppen 0 txpe 0x8 rxpe 0x8
       119        405805680      INFO      PORT  port_hss_sigdet[0]: hss_sigdet changed to 0x0
       120        405805680      INFO      PORT  port[0] link down (1) (lstatus 0xa)
       121        405806057      INFO      PORT  port[0] update (flowcid 39032 rc 0)
       122        405905315      INFO      PORT  bean_fsm[0] : state START (count = 1)
       123        405905316      INFO      PORT  hw_mac_init_port[0], ptype 0x11, speed 0x8, lanes 0xf, fec 0x0
```

- **ibqulp**

Description: Collect/Display CIM ULP inbound queue.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> ibqulp <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg ibqulp
cudbg_view() dbg entity : ibqulp
000000: 3d000000 00980801 00001350 00000058
0x0010: 3d000000 00980801 000026e0 00000000
0x0020: 00000000 00980801 00000000 00000000
0x0030: 00000000 00980901 00000000 00000000
0x0040: 00000000 00980901 00000000 00000040
0x0050: 00000000 00980901 00000000 00000000
0x0060: 00000000 00980801 00000000 7657f010
0x0070: 00000000 00980801 00000000 01b876d0
0x0080: 00000000 00980901 00000000 00000000
0x0090: 00000000 00980901 00000000 00000000
0x00a0: 00000000 00980801 00000000 00000000
0x00b0: 00000000 00980801 00000000 00000000
0x00c0: 00000000 00980901 00000000 00000000
0x00d0: 00000000 00980901 00000000 00000018
0x00e0: 00000000 00980901 00000000 00000000
0x00f0: 00000000 00980901 00000000 008fb63c
0x0100: 00000000 00980801 00000000 00000000
0x0110: 00000000 00980801 00000000 00000018
```

- **mc0, mc1**

Description: Collect MC memory details.

Syntax: `cxgbtool <vbdIface> cudbg [collect|extract] mc[0|1] <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg extract mc0 log_mc0 C:\Users\Administrator\Desktop\dump_mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0
cudbg_view() dbg entity : mc0

Debug logs extracted to C:\Users\Administrator\Desktop\dump_mc0
```

Note *It is recommended that the dump file be provided without any extension.*

- **rss_config**

Description: Collect/Display RSS Configuration.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> rss_config <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg rss_config
cudbg_view() dbg entity : rss_config
TP_RSS_CONFIG: 0x300001c
  Tnl4TupEnIpv6:  no
  Tnl2TupEnIpv6:  no
  Tnl4TupEnIpv4:  no
  Tnl2TupEnIpv4:  no
  TnlTcpSel:      no
  TnlIp6Sel:      no
  TnlVrtSel:     yes
  TnlMapEn:      yes
  OfdHashSave:    no
  OfdVrtSel:      no
  OfdMapEn:       no
  OfdLkpEn:       no
  Syn4TupEnIpv6:  no
  Syn2TupEnIpv6:  no
  Syn4TupEnIpv4:  no
  Syn2TupEnIpv4:  no
  Syn4TupEnIpv6:  no
  SynIp6Sel:      no
  SynVrt6Sel:     no
  SynMapEn:       no
  SynLkpEn:       no
  ChnEn:          no
  PrtEn:          no
  TnlAllLkp:      no
```

- **tcpstats**

Description: Collect/Display IPv4/IPv6 TCP statistics.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> tcpstats <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg tcpstats
cudbg_view() dbg entity : tcpstats
                                IP                  IPv6
OutRsts:                         0                     0
InSegs:                     491270                     0
OutSegs:                    491262                     0
RetransSegs:                     2                     0
```

- **cplstats**

Description: Collect/Display CPL Request and Response Statistics for all channels.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> cplstats <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg view cplstats log_cplstats
cudbg_view() dbg entity : cplstats
                channel 0  channel 1
CPL requests:       246298          7
CPL responses:        1524          7
```

- **meminfo**

Description: Collect/Display memory information.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> meminfo <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg view meminfo log_meminfo
cudbg_view() dbg entity : meminfo
EDC0:           0-0x3fffff [4.00 MiB]
EDC1:           0x400000-0x7fffff [4.00 MiB]
MC:             0x800000-0x807fffff [2.00 GiB]
RQUDP region:   0xfffffff0-0xfffffffe [0 B]
IMSG contexts:  0x23a480-0x2d047f [600 KiB]
ULPTX state:    0x2d0480-0x2d9dbf [38.3 KiB]
ULPRX state:    0x2d9dc0-0x2de9bf [19.0 KiB]
Pstructs:       0x2de9c0-0x3de5bf [1023 KiB]
Rx FL:          0x3de5c0-0x3e0f7f [10.4 KiB]
Tx FL:          0x3e0f80-0x3e8f7f [32.0 KiB]
Pstruct FL:     0x3e8f80-0x3f3a7f [42.7 KiB]
LE hash:        0x3f3a80-0x433a7f [256 KiB]
TCBs:           0x433a80-0x69ffff [2.42 MiB]
Rx payload:     0x6a0000-0x1011ffff [250 MiB]
DBQ contexts:   0x10280000-0x10315fff [600 KiB]
FLM cache:      0x10316000-0x10368c3f [331 KiB]
Timers:         0x10368c40-0x107fffff [4.58 MiB]
TDDP region:    0x10800000-0x1170f07f [15.0 MiB]
iSCSI region:   0x1170f080-0x11f0f07f [8.00 MiB]
TPT region:     0x11f0f080-0x16a5a4ff [75.2 MiB]
STAG region:    0x11f0f080-0x16a5a4ff [75.2 MiB]
TXPBL region:   0x16a5a500-0x36a5a4ff [512 MiB]
PBL region:     0x16a5a500-0x36a5a4ff [512 MiB]
RQ region:      0x36a5a500-0x400f0dff [150 MiB]
Tx payload:     0x40800000-0x707fffff [768 MiB]
uP RAM:         0x79728000-0x807fffff [112 MiB]
uP Extmem2:     0-0xffffffff [0 B]
```

- **pcieindirect**

Description: Collect/Display PCIe indirect registers information.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> pcieindirect <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg pcieindirect
cudbg_view() dbg entity : pcieindirect


PCIE_PDBG

[0x05a04:0x00000] PCIE_PDEBUG_REG_0x0                              0x3504349      55591753
[0x05a04:0x00001] PCIE_PDEBUG_REG_0x1                              0xfcafbcb6     4239375542
[0x05a04:0x00002] PCIE_PDEBUG_REG_0x2                              0x1000600      16778752
    18:11 tagq_ch0_tags_used                                       0              0
    10:10 tagq_ch0_data_empty                                      0x1            1
      9:9 rdq_ch0_req_empty                                        0x1            1
      8:8 req_ctl_rd_ch0_wait_for_tagtq                            0              0
      7:7 req_ctl_rd_ch0_wait_for_cmd                              0              0
      6:6 req_ctl_rd_ch0_wait_for_data_mem                         0              0
      5:5 req_ctl_rd_ch0_wait_for_rdq                              0              0
      4:4 req_ctl_rd_ch0_wait_for_txn_disable_fifo                 0              0
      3:3 req_ctl_rd_ch0_exit_bot_vld_started                      0              0
      2:2 req_ctl_rd_ch0_exit_top_vld_started                      0              0
      1:1 req_ctl_rd_ch0_wait_for_pause                            0              0
      0:0 req_ctl_rd_ch0_wait_for_fifo_data                        0              0
[0x05a04:0x00003] PCIE_PDEBUG_REG_0x3                              0x200f840      33617984
    18:11 tagq_ch1_tags_used                                       0x1f           31
    10:10 req_ch1_data_empty                                       0              0
      9:9 rdq_ch1_req_empty                                        0              0
      8:8 req_ctl_rd_ch1_wait_for_tagtq                            0              0
      7:7 req_ctl_rd_ch1_wait_for_cmd                              0              0
      6:6 req_ctl_rd_ch1_wait_for_data_mem                         0x1            1
      5:5 req_ctl_rd_ch1_wait_for_rdq                              0              0
      4:4 req_ctl_rd_ch1_wait_for_txn_disable_fifo                 0              0
      3:3 req_ctl_rd_ch1_exit_bot_vld_started                      0              0
      2:2 req_ctl_rd_ch1_exit_top_vld_started                      0              0
      1:1 req_ctl_rd_ch1_wait_for_pause                            0              0
      0:0 req_ctl_rd_ch1_wait_for_fifo_data                        0              0
[0x05a04:0x00004] PCIE_PDEBUG_REG_0x4                              0              0
    18:11 tagq_ch2_tags_used                                       0              0
    10:10 req_ch2_data_empty                                       0              0
      9:9 rdq_ch2_req_empty                                        0              0
      8:8 req_ctl_rd_ch2_wait_for_tagtq                            0              0
      7:7 req_ctl_rd_ch2_wait_for_cmd                              0              0
      6:6 req_ctl_rd_ch2_wait_for_data_mem                         0              0
      5:5 req_ctl_rd_ch2_wait_for_rdq                              0              0
      4:4 req_ctl_rd_ch2_wait_for_txn_disable_fifo                 0              0
      3:3 req_ctl_rd_ch2_exit_bot_vld_started                      0              0
      2:2 req_ctl_rd_ch2_exit_top_vld_started                      0              0
      1:1 req_ctl_rd_ch2_wait_for_pause                            0              0
      0:0 req_ctl_rd_ch2_wait_for_fifo_data                        0              0
```

- **dumpcontext**

Description: Collect/Display SGE context.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> dumpcontext <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg view dumpcontext log_dumpcontext
cudbg_view() dbg entity : dumpcontext


Context type: egress
Queue ID: 0
DCA_ST:                                 0
StatusPgNS:                             0
StatusPgRO:                             0
FetchNS:                                0
FetchRO:                                0
Valid:                                  0x1
ReschedulePending_1:                    0
PCIeDataChannel:                        0
StatusPgTPHintEn:                       0
StatusPgTPHint:                         0
FetchTPHintEn:                          0
FetchTPHint:                            0
FCThreshOverride:                       0
WRLength:                               0
WRLengthKnown:                          0x200
ReschedulePending:                      0
TimerIx:                                0x1
```

- **ulptxla**

Description: Collect/Display ULP TX LA information.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> ulptxla <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg ulptxla
cudbg_view() dbg entity : ulptxla
==============================
DUMPING ULP_TX_LA_0
==============================
[0x8ec0] ULP_TX_LA_RDPTR_0        0
[0x8ec8] ULP_TX_LA_WRPTR_0        0
[0x8ec4] ULP_TX_LA_RDDATA_0       0
[0]   0               [0]
[0x1]   0               [0]
[0x2]   0               [0]
[0x3]   0               [0]
[0x4]   0               [0]
[0x5]   0               [0]
[0x6]   0               [0]
[0x7]   0               [0]
[0x8]   0               [0]
[0x9]   0               [0]
[0xa]   0               [0]
[0xb]   0               [0]
[0xc]   0               [0]
[0xd]   0               [0]
[0xe]   0               [0]
[0xf]   0               [0]
```

- **cimla**

Description: Collect/Display CIM LA information.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> cimla <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg cimla
cudbg_view() dbg entity : cimla
Status   Inst     Data       PC      LS0Stat  LS0Addr  LS0Data  LS1Stat  LS1Addr  LS1Data
  02   00003000 00001000 1fffcfd2 00b00020 1ffce008 00000000 00000010 a02ced98 00000000
  02   00003000 00001000 1fffcfd2 00b00020 1ffce008 00000000 00000010 a02ced98 00000000
  3c   00003003 1fffcfd2 1fffcfd2 00b00020 1ffce008 00000000 00000010 a02ced98 00000000
  02   00003000 00001000 1fffcfd5 00b00020 1ffce008 00000000 00000010 a02ced98 00000000
  3c   00003003 1fffcfd5 1fffcfd5 00b00020 1ffce008 00000000 00000010 a02ced98 00000000
  3c   00003002 1fffcfd8 1fffcfd8 00a00025 1fff1588 1fffc8f8 00000010 a02ced98 00000000
  02   00003000 00001000 1fffcfda 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  02   00003000 00001000 1fffcfda 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003003 1fffc8f8 1fffcfda 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffcfda 1fffc8f8 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffcfda 1fffc8f8 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffcfda 1fffc8f8 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  3c   00003003 1fffc8f8 1fffc8f8 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  08   00003002 1fffcfdd 1fffc8fb 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffc8fb 1fffcfdd 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffc8fb 1fffcfdd 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffc8fb 1fffcfdd 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  14   00003003 1fffcfcc 1fffcfdd 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffcfdd 1fffcfcc 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffcfdd 1fffcfcc 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  04   00003000 1fffcfdd 1fffcfcc 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  3c   00003003 1fffcfcc 1fffcfcc 00a00020 1fff1588 00000000 00000010 a02ced98 00000000
  3c   00003003 1fffcfcf 1fffcfcf 00b00025 1ffce008 00000000 00000010 a02ced98 00000000
```

- **ibqsge0, ibqsge1**

Description: Collect/Display CIM SGE inbound queue.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> ibqsge[0|1] <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg ibqsge1
cudbg_view() dbg entity : ibqsge1
000000: 00000000 969d0c76 00000000 8efa4efc
0x0010: 00000000 f8e552e4 00000000 c837c219
0x0020: 00000000 e871a8ac 00000000 f1310f54
0x0030: 00000000 fb8adbef 00000000 71f86d8f
0x0040: 00000000 519c6b7e 00000000 b8958216
0x0050: 00000000 8b7fd936 00000000 ac13bdff
0x0060: 00000000 e11694c6 00000000 7098783f
0x0070: 00000000 66eab2b5 00000000 6ba6cd67
0x0080: 00000000 00710b5f 00000000 d9a070cb
0x0090: 00000000 c883b836 00000000 2607fd88
0x00a0: 00000000 26a27306 00000000 40a82d63
0x00b0: 00000000 61cc81ad 00000000 1000a776
0x00c0: 00000000 38b5db23 00000000 be93cb01
0x00d0: 00000000 7ad76889 00000000 1275733f
0x00e0: 00000000 a2e8a4cc 00000000 fb82414c
0x00f0: 00000000 a0d2b10e 00000000 dc24aae7
0x0100: 00000000 4a0da9bc 00000000 24f5518c
0x0110: 00000000 33bb8a6e 00000000 39673cd2
0x0120: 00000000 901e7a25 00000000 18558425
0x0130: 00000000 a599f3ad 00000000 76b77ef4
0x0140: 00000000 4af71c44 00000000 65f5a5c8
```

- **pathmtu**

Description: Collect/Display hardware MTU table.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> pathmtu <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg pathmtu
cudbg_view() dbg entity : pathmtu
88 256 512 576 808 1024 1280 1488 1500 2002 2048 4096 4352 8192 9000 9600
```

- **tperrstats**

Description: Collect/Display TP error statistics for channel 0-3.

Syntax:

`cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> tperrstats <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg tperrstats
cudbg_view() dbg entity : tperrstats
                  channel 0  channel 1
macInErrs:                0          0
hdrInErrs:                0          0
tcpInErrs:                0          0
tcp6InErrs:               0          0
tnlCongDrops:             0          0
tnlTxDrops:               0          0
ofldVlanDrops:            0          0
ofldChanDrops:            0          0

ofldNoNeigh:    0
ofldCongDefer:  0
```

- **ddpstats**

Description: Collect/Display DDP (Direct Data Placement) statistics.

Syntax:

`cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> ddpstats <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg ddpstats
cudbg_view() dbg entity : ddpstats
Frames: 0
Octets: 0
Drops:  0
```

- **cimpifla**

Description: Collect/Display CIM PIF logic analyzer trace.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> cimpifla <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg cimpifla
cudbg_view() dbg entity : cimpifla
Cntl ID DataBE   Addr                 Data
 01  01  f000  e1003060 80005044000000000000000000000000
 81  00  f000  e1003060 00005044000000000000000000000000
 81  00  f000  e1003060 80004044000000000000000000000000
 01  01  0f00  e1003064 80004044000000000000000000000000
 01  01  f000  e1003060 80004044000000000000000000000000
 81  00  f000  e1003060 00004044000000000000000000000000
 81  00  f000  e1003060 80003044000000000000000000000000
 01  01  0f00  e1003064 80003044000000000000000000000000
 01  01  f000  e1003060 80003044000000000000000000000000
 81  00  f000  e1003060 00003044000000000000000000000000
 81  00  f000  e1003060 80002044000000000000000000000000
 01  01  0f00  e1003064 80002044000000000000000000000000
 01  01  f000  e1003060 80002044000000000000000000000000
 81  00  f000  e1003060 00002044000000000000000000000000
 81  00  f000  e1003060 80001044000000000000000000000000
```

- **pmindirect**

Description: Collect/Display PM indirect registers information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> pmindirect <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg pmindirect
cudbg_view() dbg entity : pmindirect


PM_RX

[0x08fd0:0x10000] PM_TX_ISPI_DBG_4B_DATA0                    0           0
     31:0 ispi_dbg_data                                      0           0
[0x08fd0:0x10001] PM_RX_ISPI_DBG_4B_DATA1                    0           0
     31:0 ispi_dbg_data                                      0           0
[0x08fd0:0x10002] PM_RX_ISPI_DBG_4B_DATA2                    0           0
     31:0 ispi_dbg_data                                      0           0
[0x08fd0:0x10003] PM_RX_ISPI_DBG_4B_DATA3                    0           0
     31:0 ispi_dbg_data                                      0           0
[0x08fd0:0x10004] PM_RX_ISPI_DBG_4B_DATA4                    0           0
     31:0 ispi_dbg_data                                      0           0
[0x08fd0:0x10005] PM_RX_ISPI_DBG_4B_DATA5                    0           0
     31:0 ispi_dbg_data                                      0           0
[0x08fd0:0x10006] PM_RX_ISPI_DBG_4B_DATA6                    0           0
     31:0 ispi_dbg_data                                      0           0
[0x08fd0:0x10007] PM_RX_ISPI_DBG_4B_DATA7                    0           0
     31:0 ispi_dbg_data                                      0           0
[0x08fd0:0x10008] PM_RX_ISPI_DBG_4B_DATA8                    0           0
     31:0 ispi_dbg_data                                      0           0
```

- **mpstcam**

Description: Collect/Display MPS TCAM configuration.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> mpstcam <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg mpstcam
cudbg_view() dbg entity : mpstcam
Idx  Ethernet address     Mask       VNI    Mask   IVLAN Vld DIP_Hit   Lookup  Port Vld Ports PF  VF                             Replication                              P0 P1 P2 P3  ML
  0 01:80:c2:00:00:0e ffffffffffff    -      -      -    N    -        O      0    Y   0x3   7  98 00000000 00000000 00000000 00000000 0000000c 00000000 00000000 00000000  0  0  0  0 0
  1 00:07:43:04:b3:90 ffffffffffff    -      -      -    N    -        O      0    Y   0x1   4  65                                                                          0  0  0  0 0
  2 01:00:5e:00:00:01 ffffffffffff    -      -      -    N    -        O      0    Y   0x3   4  65 00000000 00000000 00000000 00000000 00000000 00000006 00000000 00000000  0  0  0  0 0
  3 33:33:00:00:00:01 ffffffffffff    -      -      -    N    -        O      0    Y   0x3   4  65 00000000 00000000 00000000 00000000 00000000 00000006 00000000 00000000  0  0  0  0 0
  4 33:33:ff:f5:69:e5 ffffffffffff    -      -      -    N    -        O      0    Y   0x1   4  65                                                                          0  0  0  0 0
  5 00:07:43:04:b3:98 ffffffffffff    -      -      -    N    -        O      0    Y   0x2   4  66                                                                          0  0  0  0 0
  6 33:33:ff:fd:e6:68 ffffffffffff    -      -      -    N    -        O      0    Y   0x2   4  66                                                                          0  0  0  0 0
  7 33:33:00:00:00:fb ffffffffffff    -      -      -    N    -        O      0    Y   0x3   4  66 00000000 00000000 00000000 00000000 00000000 00000006 00000000 00000000  0  0  0  0 0
  8 33:33:00:01:00:03 ffffffffffff    -      -      -    N    -        O      0    Y   0x3   4  66 00000000 00000000 00000000 00000000 00000000 00000006 00000000 00000000  0  0  0  0 0
  9 01:00:5e:00:00:fb ffffffffffff    -      -      -    N    -        O      0    Y   0x3   4  66 00000000 00000000 00000000 00000000 00000000 00000006 00000000 00000000  0  0  0  0 0
 10 01:00:5e:00:00:fc ffffffffffff    -      -      -    N    -        O      0    Y   0x3   4  66 00000000 00000000 00000000 00000000 00000000 00000006 00000000 00000000  0  0  0  0 0
```

- **upcimindirect**

Description: Collect/Display UP CIM Indirect registers information.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> upcimindirect <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg upcimindirect
cudbg_view() dbg entity : upcimindirect
[0x07b50:0x02000] CIM_CTL_CONFIG                                  0xc060         49248
    21:17 AutoPrefLoc                                             0              0
    16:16 AutoPrefEn                                              0              0
    15:15 DisMATimeOut                                            0x1            1
    14:14 DisSlowTimeOut                                          0x1            1
      9:9 IntlRspEn                                               0              0
      8:8 PIFMultiCmd                                             0              0
      7:7 UPSelfResetTOut                                         0              0
      6:6 PLSwapDisWr                                             0x1            1
      5:5 PLSwapDisRd                                             0x1            1
      4:4 Timer1En                                                0              0
      3:3 Timer0En                                                0              0
      1:1 TimerEn                                                 0              0
      0:0 PrefEn                                                  0              0
[0x07b50:0x02004] CIM_CTL_PREFADDR                                0              0
     31:0 PrefAddr                                                0              0
[0x07b50:0x02008] CIM_CTL_ALLOCADDR                               0              0
     31:0 IbqGen0                                                 0              0
[0x07b50:0x0200c] CIM_CTL_INVLDTADDR                              0              0
     31:0 InvldtAddr                                              0              0
[0x07b50:0x02010] CIM_CTL_STATIC_PREFADDR0                        0              0
     31:0 StaticPrefAddr                                          0              0
[0x07b50:0x02014] CIM_CTL_STATIC_PREFADDR1                        0              0
     31:0 StaticPrefAddr                                          0              0
[0x07b50:0x02018] CIM_CTL_STATIC_PREFADDR2                        0              0
     31:0 StaticPrefAddr                                          0              0
[0x07b50:0x0201c] CIM_CTL_STATIC_PREFADDR3                        0              0
     31:0 StaticPrefAddr                                          0              0
```

- **cimmala**

Description: Collect/Display results of CIM MA logic analyzer trace.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> cimmala <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect cimmala log_cimmala
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file log_cimmala... size: 3668 bytes
cxgbtool: Done writing cudbg data to file log_cimmala

C:\Users\Administrator>cxgbtool vbd0 cudbg view cimmala log_cimmala
cudbg_view() dbg entity : cimmala
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

- **obqsge**

Description: Collect/Display SGE outbound queue.

Syntax:

`cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> obqsge <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg obqsge
cudbg_view() dbg entity : obqsge
000000: 01000022 0063db01 01000449 00987803
0x0010: 01000320 0063dc01 0100ca4c 00987803
0x0020: 01286a20 0063dd01 0100b8eb 00987803
0x0030: 01000023 0063de01 01007eec 00987803
0x0040: 01000023 0063df01 01006f5a 00987803
0x0050: 01000320 0063e001 010087d0 00987803
0x0060: 0128ea20 0063e101 0100738d 00987803
0x0070: 01000023 0063e201 0100398c 00987803
0x0080: 01000023 0063e301 0100143a 00987803
0x0090: 01000320 0063e401 0100da26 00987803
0x00a0: 01296a20 0063e501 0100d77e 00987803
0x00b0: 01000022 0063e601 01009df3 00987803
0x00c0: 01000022 0063e701 0100a58b 00987803
0x00d0: 01000322 0063e801 01006b75 00987803
```

- **rss**

Description: Collect/Display RSS information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> rss <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg rss
cudbg_view() dbg entity : rss
   0:     0     0     0     0     0     0     0     0
   8:     0     0     0     0     0     0     0     0
  16:     0     0     0     0     0     0     0     0
  24:     0     0     0     0     0     0     0     0
  32:     0     0     0     0     0     0     0     0
  40:     0     0     0     0     0     0     0     0
  48:     0     0     0     0     0     0     0     0
  56:     0     0     0     0     0     0     0     0
  64:     0     0     0     0     0     0     0     0
  72:     0     0     0     0     0     0     0     0
  80:     0     0     0     0     0     0     0     0
  88:     0     0     0     0     0     0     0     0
  96:     0     0     0     0     0     0     0     0
 104:     0     0     0     0     0     0     0     0
 112:     0     0     0     0     0     0     0     0
 120:     0     0     0     0     0     0     0     0
 128:    37    41    36    38    39    40    42    43
 136:    37    41    36    38    39    40    42    43
 144:    37    41    36    38    39    40    42    43
 152:    37    41    36    38    39    40    42    43
 160:    37    41    36    38    39    40    42    43
 168:    37    41    36    38    39    40    42    43
```

- **swstate**

Description: Collect/Display software state information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> swstate <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg swstate
cudbg_view() dbg entity : swstate

        FW STATE   : Alive
        OS         : Unknown
        CALLER     : Unknown
```

- **fcoestats**

Description: Collect/Display FCoE statistics.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> fcoestats <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg fcoestats
cudbg_view() dbg entity : fcoestats
                  channel 0        channel 1
octetsDDP:                0                0
framesDDP:                0                0
framesDrop:               0                0
```

- **wcstats**

Description: Collect/Display write coalescing statistics.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> wcstats <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg wcstats
cudbg_view() dbg entity : wcstats
WriteCoalSuccess: 0
WriteCoalFail:    0
```

- **clk**

Description: Collect/Display core clock information.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> clk <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg clk
cudbg_view() dbg entity : clk
Core clock period: 1.252 ns
TP timer tick: 41.25536 us
TCP timestamp tick: 1312.817152 us
DACK tick: 5.128192 us
DACK timer: 485 us
Retransmit min: 984 us
Retransmit max: 9984730 us
Persist timer min: 998473 us
Persist timer max: 9984730 us
```

- **full**

Description: Verify if egress and ingress buffers are becoming full.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> full <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg full
cudbg_view() dbg entity : full

Tx0 ==0=>  T  <=0= Rx0
Tx1 ==0=>  P  <=0= Rx1

Tx0 P =0=> S ? U =>0=>  T
Tx1 C =0=> G ? T =>0=>  P
       Rd Wr
RX0 P <=0=0=0 S <=0= C <=0= T <=T <=0=  T <=0= M
RX1 C <=0=0=0 G <=0= X <=0= C <=P <=0=  E <=0= P
```

- **pbttables**

Description: Collect/Display PBT Table information.

Syntax:

`cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> pbttables <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg pbttables
cudbg_view() dbg entity : pbttables
Dynamic Addr Table [0x000]: 0x002d880c
    [28:28]  vld         0
    [27:27]  alloc       0
    [26:26]  pending     0
    [25: 0]  address<<6  0x0b620300
Dynamic Addr Table [0x004]: 0x02a2298d
    [28:28]  vld         0
    [27:27]  alloc       0
    [26:26]  pending     0
    [25: 0]  address<<6  0x288a6340
Dynamic Addr Table [0x008]: 0x012bb725
    [28:28]  vld         0
    [27:27]  alloc       0
    [26:26]  pending     0
    [25: 0]  address<<6  0x4aedc940
Dynamic Addr Table [0x00c]: 0x008032ae
    [28:28]  vld         0
```

- **vpddata**

Description: Collect/Display vpd information like serial number, vpd version, firmware version, etc.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> vpddata <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect vpddata log_vpddata
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file log_vpddata... size: 3720 bytes
cxgbtool: Done writing cudbg data to file log_vpddata

C:\Users\Administrator>cxgbtool vbd0 cudbg view vpddata log_vpddata
cudbg_view() dbg entity : vpddata
MN T520-SO
SN PT26130519
BN 11011695002
NA 000743288600
SCFG Version 0x1005000
VPD Version  0x1
Firmware Version: 1.16.38.0
```

- **cimqcfg**

Description: Collect/Display CIM queue configuration details.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> cimqcfg <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect cimqcfg  log_cimqcfg
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file log_cimqcfg... size: 3896 bytes
cxgbtool: Done writing cudbg data to file log_cimqcfg

C:\Users\Administrator>cxgbtool vbd0 cudbg view cimqcfg  log_cimqcfg
cudbg_view() dbg entity : cimqcfg
  Queue Base Size Thres  RdPtr WrPtr  SOP  EOP Avail
  TP0     0  2048    0    570   570 2412 2412  2048
  TP1   800  2048    0      0     0    0    0  2048
  ULP  1000  2048    0     20    20    2    2  2048
 SGE0  1800  2048    0    480   480  762  762  2048
 SGE1  2000  2048    0      0     0    0    0  2048
NC-SI  2800  2048    0      0     0    0    0  2048
 ULP0     0  2048         7f0   7f0 3895 3895  2048
 ULP1   800  2048         1d0   1d0   53   53  2048
 ULP2  1000  2048           0     0    0    0  2048
 ULP3  1800  2048           0     0    0    0  2048
  SGE  2000  2048         440   440  840  840  2048
NC-SI  2800  2048           0     0    0    0  2048
```

- **ibqncsi**

Description: Collect/Display CIM NCSI inbound queue.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> ibqncsi <file> <dir>
```

Example:

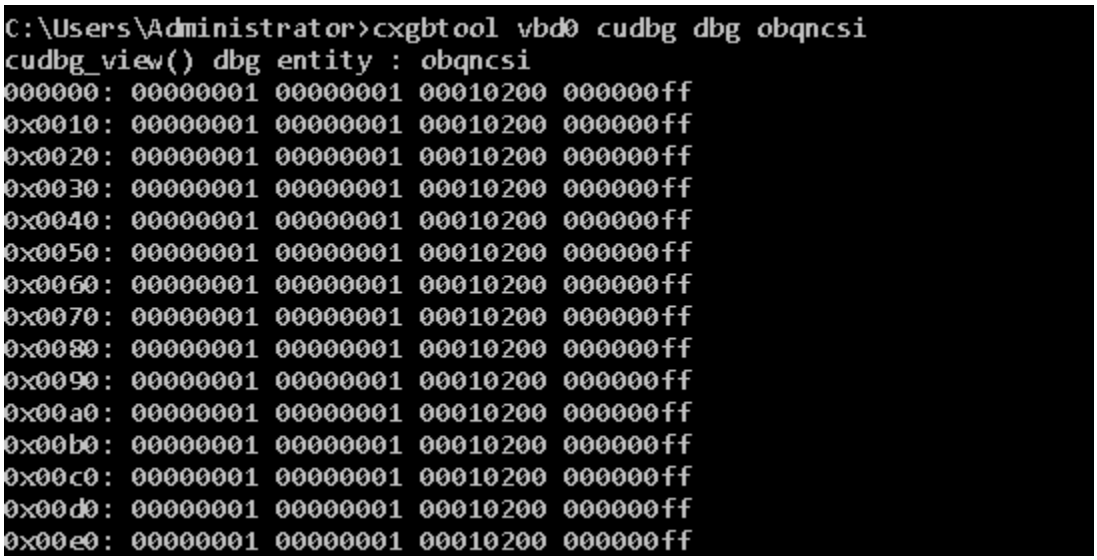
- **obqncsi**

Description: Collect/Display NCSI outbound queue.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> obqncsi <file> <dir>
```

Example:



- **rss_pf_config**

Description: Collect/Display RSS PF Configuration

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> rss_pf_config <file>
<dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg rss_pf_config
cudbg_view() dbg entity : rss_pf_config
PF Map Index Size = 0

     RSS                PF   VF    Hash Tuple Enable          Default
     Enable         IPF Mask Mask  IPv6       IPv4       UDP   Queue
 PF  Map Chn Prt  Map Size Size  Four Two  Four Two  Four  Ch1   Ch0
  0  yes  no  no    0    7    5    no  no    no  no    no    0     0
  1  yes  no  no    1    7    5    no  no    no  no    no    0     0
  2  yes  no  no    2    7    5    no  no    no  no    no    0     0
  3  yes  no  no    3    7    5    no  no    no  no    no    0     0
  4  yes  no  no    4    7    5    no  no    no  no    no    0     0
  5  yes  no  no    5    7    5    no  no    no  no    no    0     0
  6  yes  no  no    6    7    5    no  no    no  no    no    0     0
  7  yes yes  no    7    7    7   yes yes   yes yes    no    0     0
```

- **wtp**

Description: Trace the Ingress and Egress path of a packet through the Chelsio adapter.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> wtp <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg wtp
cudbg_view() dbg entity : wtp
ifaces = nic0 nic1
*************************EGGRESS (TX) PATH *********************************
MOD :  core---->PCIE---->SGE<-|    #Ring Doorbell
SOP        ?      ???         |
EOP        ?      ???         |
MOD |<-core<----PCIE<----SGE<-|    #Request Work Request
SOP_CH0  AB       0b
SOP |    AB       16
EOP |    AB       16
MOD |->core---->PCIE---->SGE------>CIM/uP->| uP<-CIM<-CSW #->Work req. <-Pkts
SOP_CH0  AB       00      08               |       1
SOP_CH1                   00               |
SOP      AB       00       8               |       1
EOP      AB        0       8               |
MOD |<-core<----PCIE<----SGE<------UTX<--------|#data dma requests
SOP_CH0  0F
SOP_CH1  00
SOP |     F
EOP |     F
MOD |->core-->PCIE-->SGE-->UTX---->TPC------->TPE---->MPS--->MAC--->MACOK->wire
SOP_CH0        BF          8       8        8     1   E1  E1  E1      FF    FF
EOP_CH0        03          8       8        8     1   E1  E1  E1      FF    FF
SOP_CH1        50          7       7        7     0   40  40  40      FF  FF
EOP_CH1        03          7       7        7     0   40  40  40      FF    FF
SOP_CH2        00          0       0        0     0   00  00
EOP_CH2        00          0       0        0     0   00  00
SOP_CH3        00          0       0        0     0   00  00
EOP_CH3        00          0       0        0     0   00  00
SOP            10F          F       0        F     1   121  121  121      1FE    1FE
EOP             6          0       0        F     1   121  121  121      1FE    1FE
*************************INGRESS (RX) PATH *********************************
MOD   core<-PCIE<---SGE<--CSW<-----TPC<-URX<-LE-TPE<-----MPS<--MAC<-MACOK<--wire
SOP_CH0       9   2     A     B    B   D   D  8   F     0   5F  B7      FF    FF
EOP_CH0       9   2     A     B    B   D   D  8   F     0   5F  B7      FF    FF
SOP_CH1       C   9     0     0    0   0   0      7     0   00  B6      FF    FF
EOP_CH1       C   9     0     0    0   0   0      7     0   00  B6      FF    FF
SOP_CH2                                          0         37
EOP_CH2                                          0         37
SOP_CH3                                          0         00
EOP_CH3                                          0         00
SOP_CH4                                                    00
EOP_CH4                                                    00
SOP_CH5                                                    00
EOP_CH5                                                    00
SOP_CH6
EOP_CH6
SOP_CH7
EOP_CH7
```

- **rdmastats**

Description: Collect/Display RDMA Statistics.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> rdmastats <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg rdmastats
cudbg_view() dbg entity : rdmastats
NoRQEModDefferals: 0
NoRQEPktDefferals: 0
```

- **ulprxla**

Description: Collect/Display ULP RX LA.

Syntax:

```
cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> ulprxla <file> <dir>
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg ulprxla
cudbg_view() dbg entity : ulprxla
      Pcmd            Type   Message                 Data
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
0000000000000000       0   00000000   00000000000000000000000000000000
```

- **txrate**

Description: Collect/Display Tx rate for NIC and offload traffic.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> txrate <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg txrate
cudbg_view() dbg entity : txrate


                TX_RATE

              channel 0   channel 1
NIC B/s:              0           0
Offload B/s:          0           0
```

- **letcam**

Description: Collect/Display LE TCAM information.

Syntax: `cxgbtool <vbdIface> cudbg <collect|view|extract|dbg> letcam <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg letcam
cudbg_view() dbg entity : letcam


Routing table index: 0x0
Lip comp table index: 0x0
Filter table index: 0xA80
Server index: 0xA00

tid start: 0

tid end: 19456

========================================================================================
This is a LE_DB_DATA_READ command: on TID 0 at index 0
Region: active

READ:
DBGICMDMODE: LE
READING TID: 0x0
Write: LE_DB_DBGI_REQ_TCAM_CMD: 0x400000
Write: LE_DB_DBGI_CONFIG 0x5
Polling: LE_DB_DBGI_CONFIG: busy bit
Read: LE_DB_DBGI_RSP_STATUS: 0x1 [1]
Read: LE_DB_DBGI_RSP_DATA:
Response data for TID 0x0:
        0x19DA0: 0x00000000
        0x19DA4: 0x00000000
        0x19DA8: 0x00000000
        0x19DAC: 0x00000000
        0x19DB0: 0x00000000
        0x19DB4: 0x00000000
        0x19DB8: 0x00000000
        0x19DBC: 0x00000000
        0x19DC0: 0x00000000
        0x19DC4: 0x00000000
        0x19DC8: 0x00000000
        0x19DCC: 0x00000000
        0x19DD0: 0x00000000
        0x19DD4: 0x00000000
        0x19DD8: 0x00000000
        0x19DDC: 0x00000000
        0x19DE0: 0x00000000
DATA READ: 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

- **hma**

Description: Collect HMA region.

Syntax: `cxgbtool <vbdIface> cudbg <collect|extract> hma <file> <dir>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg collect hma hma_dbg
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file hma_dbg... size: 437484 bytes
cxgbtool: Done writing cudbg data to file hma_dbg
```

debug

- **filter**

Description: Display list of configured hardware filters

Syntax: `cxgbtool <nicIface> debug filter`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug filter

Ftid       Prot       FPORT      LPORT      Type       Que_Id     DMAC_Idx   Locked     Hits

0          0          0          0          IPv4       0          0          1          23
1          0          0          0          IPv4       0          0          1          0
```

- **qsets**

Description: Read number of qsets

Syntax: `cxgbtool <nicIface> debug qsets`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug qsets

QueType    AbsId      RelId      Fl0Id      IngId      Msix       QDepth

TxEth      5          5          n/a        5          n/a        1024
TxCtrl     6          6          n/a        6          n/a        1024
TxRdma     7          7          n/a        6          n/a        512
RxIng      6          6          0          n/a        2          1023
RxEth      5          5          4          n/a        35         2047
RxRdma     1          1          0          n/a        3          511
RxRdma     2          2          1          n/a        15         511
RxRdma     3          3          2          n/a        19         511
RxRdma     4          4          3          n/a        31         511
TxVPort    13         13         n/a        11         n/a        1024
RxVPort    11         11         12         n/a        51         2047
TxVPort    19         19         n/a        14         n/a        1024
RxVPort    14         14         18         n/a        7          2047
```

- **qstats**

Description: Display statistics for each Tx & Rx queue.

Syntax: `cxgbtool <nicIface|iscsiIface> debug qstats <queueType|clr>`

`clr`: Clear queue statistics.

Queue Type	Description
txeth	Tx tunnel queue statistics.
rxeth	Rx tunnel queue statistics.
txvirt	Tx VM queue statistics.
rxvirt	Rx VM queue statistics.
txtoe	Tx Chimney queue statistics.
rxtoe	Rx Chimney queue statistics.
txrdma	Tx RDMA queue statistics.
rxrdma	Rx RDMA queue statistics.
txctrl	Chimney control queue statistics.
txfwd	Chimney forwarding queue statistics.
txnvgre *	Tx NVGRE statistics.
rxnvgre *	Rx NVGRE statistics.
txiscsi	Tx iSCSI queue statistics.
rxiscsi	Rx iSCSI queue statistics.
txpd	Tx PacketDirect queue statistics.
rxpd	Rx PacketDirect queue statistics.

** not supported on T6 adapters*

Qstats Parameter	Description
BcBytes	Broadcast packet size in bytes.
BcPkts	Number of broadcast packets.
CoalBytes	Coalesced packet size in bytes.
CoalPkts	Number of coalesced packets.
DropPkts	Number of dropped packets.
EncapBytes #	NVGRE/VXLAN packet size in bytes.
EncapPkts #	Number of NVGRE/VXLAN packets.
VlanMismatch	Number of packets with VLAN mismatch.
Fw4Ack	Number of SGE update requests.
LargeRxDropPkts	Dropped packets due to large size counter.
LSO	Number of large send offloaded packets.
McBytes	Multicast packet size in bytes.
McPkts	Number of multicast packets.
NblsRecv	Total Nbls received.
NblsSent	Total Nbls sent.
NoRxBufs	Packets couldn't process because of lack of memory.
NvPkts	Send NVGRE packets.
PeerAbort	Peer abort request.
CqComp	Number of CQ completions.

PktsDefrag	Number of defragged packets.
PktsDrop	Total packets dropped.
PktsQued	Total packets queued.
PktsSent	Total packets sent.
Quefull	Number of time queue full happened.
RecvNbls	Total Nbls received.
RssGroup	RSS group number allocated to queue.
VPortId	Vport ID of queue.
RssProc	RSS processor number allocated to queue.
RxCsumErr	Number of received packets with checksum error.
RxCsumErrIp	Number of received IP packets with checksum error.
RxCsumErrTcp	Number of received TCP packets with checksum error.
SysThreadEvts	Number of system thread events.
TxCsumOfld	Number of checksum offloaded Tx packets.
UcBytes	Unicast packets size in bytes.
UcPkts	Number of unicast packets.
VlanEx	Number of VLAN extracted packets.
VLANin	Number of VLAN inserted packets.

[#] *supported only on T6 adapters*

Example of rxeth qstats

```
C:\Users\Administrator>cxgbtool nic0 debug qstats rxeth

StatsType           RxEth#36            RxEth#37            RxEth#38            RxEth#39

UcPkts              21                  3                   3                   1
UcBytes             1324                260                 239                 93
McPkts              0                   2                   3                   0
McBytes             0                   180                 258                 0
BcPkts              6                   0                   45                  0
BcBytes             360                 0                   4572                0
CoalPkts            0                   0                   0                   0
CoalBytes           0                   0                   0                   0
DropPkts            0                   0                   0                   0
RecvNbls            27                  5                   51                  1
RxCsumErr           0                   0                   0                   0
RxCsumErrTcp        0                   0                   0                   0
RxCsumErrIp         0                   0                   0                   0
VLANex              0                   0                   0                   0
SysThreadEvts       0                   0                   0                   0
LargeRxDropPkts     0                   0                   0                   0
RssProc             0                   16                  24                  32
RssGroup            0                   0                   0                   0
```

- **dumpctx**

Description: Display adapter context.

Syntax: `cxgbtool <nicIface> debug dumpctx`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug dumpctx

Type            Value

Link State      Connected
Link Speed      10 Gbps
Hw Addr         00:07:43:39:87:f0
Cur Addr        00:07:43:39:87:f0
Port No         0
NetIfIdx        4
Mtu             1500
Pkt Filter      0xf
CurOfldCaps     0x17fe00
FwdTx Pend      0
CtrlTx RefCount 1
ToeTx Pend      0
RdmaTx RefCount 1
ToeRx Pend      0
ToeRxData Pend  0
Ndk Connect     0
Ndk Accept      0
Ndk ConnectFail 0
Ndk ConnectErr  0
```

- **version**

Description: Display adapter information like part number, serial number, device ID, firmware Version, TP, etc.

Syntax: `cxgbtool <vbdIface|nicIface|iscsiIface> debug version`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug version
P/N:            11012096004
S/N:            PT43160304
DeviceId:       6401 (T6225-CR)
FW Ver:         1.16.38.0
TP Ver:         0.1.23.2
Driver Ver:     6.5.4.10
Nic File Ver:   6.5.4.10
VBD File Ver:   6.5.4.10
Cudbg lib Ver:  1.14.0
SCFG Ver:       0x1402000
```

- **coalesce**

Description: Change the coalescing settings for tunnel Rx queues of the specified Ethernet device.

Syntax:

```
cxgbtool <nicIface> debug coalesce <options|persistent>
```

Options

`rx-usecs-irq`: Rx Coalescing Timer. Number of microseconds after which interrupt will be sent.
`rx-frames-irq`: Rx Coalescing Threshold Packets. Number of packets after which interrupt will be sent.

Examples:

- Setting Rx Coalescing Timer:

```
C:\Users\Administrator>cxgbtool nic0 debug coalesce rx-usecs-irq 5
```

- Setting Rx Coalescing Threshold Packets

```
C:\Users\Administrator>cxgbtool nic0 debug coalesce rx-frames-irq 20
```

- **eps**

Description: Print endpoints (eps) if NDK/ND is enabled.

Syntax: `cxgbtool <nicIface> debug eps`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug eps
ep FFFF888EDC304810 cm_id FFFF8E88E308EB30 state 1 flags 0x80 stid 2564 backlog 1024 fe80:0000:0000:0000:15b4:5e09:90fd:e668:5445
ep FFFF888ED7C33AE0 cm_id FFFF888ED65F84F0 state 1 flags 0x80 stid 2569 backlog 1024 169.254.230.104:5445
```

- **qps**

Description: Print queue pairs (qps) if NDK/ND is enabled.

Syntax: `cxgbtool <nicIface> debug qps`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug qps
rc qp sq id 1024 in_use 0 rq id 1025 in_use 256 state 1 onchip 0 ep tid 14576 state 7 102.1.8.2:5445 -> 102.1.8.3:1030
rc qp sq id 1026 in_use 0 rq id 1027 in_use 256 state 1 onchip 0 ep tid 15680 state 7 102.1.8.2:5445 -> 102.1.8.3:1031
rc qp sq id 1028 in_use 0 rq id 1029 in_use 224 state 1 onchip 0 ep tid 17296 state 7 102.1.8.2:5445 -> 102.1.8.3:1034
rc qp sq id 1030 in_use 0 rq id 1031 in_use 256 state 1 onchip 0 ep tid 13856 state 7 102.1.8.2:5445 -> 102.1.8.3:1035
rc qp sq id 1048 in_use 0 rq id 1049 in_use 256 state 1 onchip 0 ep tid 3400 state 7 102.1.8.2:5445 -> 102.1.8.3:1038
rc qp sq id 1050 in_use 0 rq id 1051 in_use 256 state 1 onchip 0 ep tid 10488 state 7 102.1.8.2:5445 -> 102.1.8.3:1039
rc qp sq id 1052 in_use 0 rq id 1053 in_use 256 state 1 onchip 0 ep tid 3416 state 7 102.1.8.2:5445 -> 102.1.8.3:1042
rc qp sq id 1054 in_use 0 rq id 1055 in_use 256 state 1 onchip 0 ep tid 10472 state 7 102.1.8.2:5445 -> 102.1.8.3:1043
rc qp sq id 1064 in_use 0 rq id 1065 in_use 256 state 1 onchip 0 ep tid 13872 state 7 102.1.8.2:5445 -> 102.1.8.3:1047
rc qp sq id 1066 in_use 0 rq id 1067 in_use 256 state 1 onchip 0 ep tid 7512 state 7 102.1.8.2:5445 -> 102.1.8.3:1045
rc qp sq id 1068 in_use 0 rq id 1069 in_use 256 state 1 onchip 0 ep tid 14560 state 7 102.1.8.2:5445 -> 102.1.8.3:1050
rc qp sq id 1070 in_use 0 rq id 1071 in_use 256 state 1 onchip 0 ep tid 15696 state 7 102.1.8.2:5445 -> 102.1.8.3:1051
rc qp sq id 1080 in_use 0 rq id 1081 in_use 256 state 1 onchip 0 ep tid 5000 state 7 102.1.8.2:5445 -> 102.1.8.3:1055
rc qp sq id 1082 in_use 0 rq id 1083 in_use 256 state 1 onchip 0 ep tid 9784 state 7 102.1.8.2:5445 -> 102.1.8.3:1054
```

- **rdma_stats**

Description: Print RDMA statistics if NDK/ND is enabled

Syntax: `cxgbtool <nicIface> debug rdma_stats`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug rdma_stats
   Object:      Total    Current        Max       Fail
     PDID:      65536        108        116          0
      QID:      24576        240        240          0
   TPTMEM:   78951552     114048     122496          0
   PBLMEM:  536870912     884736     950272          0
   RQTMEM:  157903104    7077888    7602176          0
  OCQPMEM:          0          0          0          0
  DB FULL:          0
 DB EMPTY:          0
  DB DROP:          0
 DB State: NORMAL Transitions 0
TCAM_FULL:          0
ACT_OFLD_CONN_FAILS:          0
PAS_OFLD_CONN_FAILS:          0
AVAILABLE IRD:     621728
```

- **stags**

Description: Print STAG contents if NDK/ND is enabled.

Syntax: `cxgbtool <nicIface> debug stags`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug stags

Stag(0):0x700
0x1150f160 0000000000000000 0000000000000000 0000000000000000 0000000000000000
0x1150f160: (inactive): stag idx 0x0700 key 0x00 state INV type NSMR pdid 0x0
                    perms none rem_inv_dis 0 addr_type ZBTO
                    bind_enable 0 pg_size 4096 qpid 0x0 pbl_off 0x0000
                    pbl_addr 0x16a5a500 len 0 va 0000000000000000 bind_cnt 0

Stag(1):0xa00
0x1150f1c0 0000000000000000 0000000000000000 0000000000000000 0000000000000000
0x1150f1c0: (inactive): stag idx 0x0a00 key 0x00 state INV type NSMR pdid 0x0
                    perms none rem_inv_dis 0 addr_type ZBTO
                    bind_enable 0 pg_size 4096 qpid 0x0 pbl_off 0x0000
                    pbl_addr 0x16a5a500 len 0 va 0000000000000000 bind_cnt 0

Stag(2):0xe00
0x1150f240 0000000000000000 0000000000000000 0000000000000000 0000000000000000
0x1150f240: (inactive): stag idx 0x0e00 key 0x00 state INV type NSMR pdid 0x0
                    perms none rem_inv_dis 0 addr_type ZBTO
                    bind_enable 0 pg_size 4096 qpid 0x0 pbl_off 0x0000
                    pbl_addr 0x16a5a500 len 0 va 0000000000000000 bind_cnt 0

Stag(3):0x1c00
0x1150f400 0000000000000000 0000000000000000 0000000000000000 0000000000000000
0x1150f400: (inactive): stag idx 0x1c00 key 0x00 state INV type NSMR pdid 0x0
                    perms none rem_inv_dis 0 addr_type ZBTO
                    bind_enable 0 pg_size 4096 qpid 0x0 pbl_off 0x0000
                    pbl_addr 0x16a5a500 len 0 va 0000000000000000 bind_cnt 0
```

- **l2t**

Description: Display l2t table contents.

Syntax: `cxgbtool <nicIface> debug l2t`

Example:

```
C:\Users\Administrator>cxgbtool nic0 debug l2t
IDX     REF_COUNT       MTU     DEST_ADDR       DEST_MAC        PORT
0       114             1500    102.1.7.3       00:07:43:04:B3:70       0
1       114             1500    102.1.8.3       00:07:43:04:B3:78       1
```

- **iscsi_log**

Description: Dump debug prints to system event log.

Syntax: `cxgbtool <iscsIface> debug iscsi_log`

Example:

```
C:\Users\Administrator>cxgbtool iscsi0 debug iscsi_log
```

- **fwtoc**

Description: Convert firmware binary file to source file header.

Syntax: `cxgbtool debug fwtoc [firmwareFile] filename= [CFile].bin`

Example:

```
C:\Users\Administrator>cxgbtool debug fwtoc Desktop\t6fw-1.16.33.0.bin filename=t6fw-1.16.33.0.bin filetype=bin
filename=t6fw-1.16.33.0.bin filetype=bin
```

- **inst**

Description: Install driver package specified in the setup information file (.inf).

Syntax: `cxgbtool debug inst <file>`

Example:

```
C:\Users\Administrator>cxgbtool debug inst Desktop\chvbdx64.inf
ENTER:  DriverPackageInstallA
ENTER:  DriverPackageInstallW
Installing INF file 'C:\Users\Administrator\Desktop\chvbdx64.inf' (Plug and Play).
Looking for Model Section [Chelsio.NTamd64]...
Installing devices with Id "PCI\VEN_1425&DEV_5410&SUBSYS_00001425&REV_00" using INF "C:\Windows\System32\DriverStore\FileRepository\chvbdx64.inf_amd64_db5d99a4eb080b3f\chvbdx64.
inf".
Installation did not occur because the hardware isn't currently present.
Installing devices with Id "PCI\VEN_1425&DEV_6401&SUBSYS_00001425&REV_00" using INF "C:\Windows\System32\DriverStore\FileRepository\chvbdx64.inf_amd64_db5d99a4eb080b3f\chvbdx64.
inf".
ENTER UpdateDriverForPlugAndPlayDevices...
RETURN UpdateDriverForPlugAndPlayDevices.
Installation was successful.
Marked Phantom Device with Hardware/Compatible Id 'PCI\VEN_1425&DEV_5410&SUBSYS_00001425&REV_00' for reinstall on next plug-in.
Install completed
RETURN: DriverPackageInstallW  (0x0)
RETURN: DriverPackageInstallA  (0x0)
```

- **uninst**

Description: Uninstall driver package specified in the setup information file (.inf).

Syntax: `cxgbtool debug uninst <file>`

Example:

```
C:\Users\Administrator>cxgbtool debug uninst Desktop\chvbdx64.inf
Removing device instance PCI\VEN_1425&DEV_6401&SUBSYS_00001425&REV_00\4&2A43D483&0&0410
Removing device instance PCI\VEN_1425&DEV_5410&SUBSYS_00001425&REV_00\4&31024D2F&0&0418
ENTER:  DriverPackageGetPathA
RETURN: DriverPackageGetPathA  (0x7A)
ENTER:  DriverPackageGetPathA
RETURN: DriverPackageGetPathA  (0x0)
ENTER:  DriverPackageUninstallA
ENTER:  DriverPackageUninstallW
Uninstalling driver package C:\Windows\System32\DriverStore\FileRepository\chvbdx64.inf_amd64_db5d99a4eb080b3f\chvbdx64.inf...
Successfully uninstalled 'C:\Windows\INF\oem6.inf'.
No devices found for C:\Windows\System32\DriverStore\FileRepository\chvbdx64.inf_amd64_db5d99a4eb080b3f\chvbdx64.inf uninstall.
Successfully deleted properties for driver store entry 'C:\Windows\System32\DriverStore\FileRepository\chvbdx64.inf_amd64_db5d99a4eb080b3f\chvbdx64.inf'.
Uninstall completed.
RETURN: DriverPackageUninstallW  (0x0)
RETURN: DriverPackageUninstallA  (0x0)
```

- **update**

Description: Update driver package.

Syntax: `cxgbtool debug update <file>`

Example:

```
C:\Users\Administrator>cxgbtool debug update C:\Users\Administrator\Desktop\v6.5.8.0\bin\fre\x64\chvbdx64.inf
Driver for HwId:PCI\VEN_1425&DEV_6407&SUBSYS_00001425&REV_00 updated successfully!!!
```

- **rescan**

Description: Scan for hardware changes in the device manager.

Syntax: `cxgbtool debug rescan`

Example:

```
C:\Users\Administrator>cxgbtool debug rescan
```

- **trace**

Description: Enable driver debug prints.

Syntax: `cxgbtool debug trace [vbd <flags> <level>]|[nic <flags> <level>]`

Example:

```
C:\Users\Administrator>cxgbtool debug trace nic 8 7
Tracing nic flags=0x8 level=7
Press 'q' to quit tracing:
```

firmware

mbox

Description: Display the last command in each mailbox.

Syntax: `cxgbtool <vbdIface|iscsiIface> firmware mbox [0|1|2|…|7]`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 firmware mbox 0
mailbox[0] owned by driver

0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
```

hardware

- **mdio**

Description: Read/Write MDIO register.

Syntax:

```
cxgbtool <vbdIface|nicIface|iscsiIface> hardware mdio <physAddr> <mngDevAddr>
<regAddr> [<value>]
```

Example

Read MDIO register

```
C:\Users\Administrator>cxgbtool vbd0 hardware mdio 1 1 1
0xffff [65535]
```

Note *Write MDIO register currently not supported.*

- **reg**

Description: Read/Write register.

Syntax:

- Register Read:

```
cxgbtool <vbdIface|nicIface|iscsiIface> hardware reg <readAddrs>[=<val>]
```

- Register Write:

```
cxgbtool <vbdIface|nicIface|iscsiIface> hardware reg <writeAddr>[=<val>]
```

Example:

- Register read:

```
C:\Users\Administrator>cxgbtool vbd0 hardware reg 0x19428
00000008 [8]
```

- Register write:

```
C:\Users\Administrator>cxgbtool vbd0 hardware reg 0x19428 = 0x3
```

- **sgedbg**

Description: Display sge debug indirect registers.

Syntax: `cxgbtool <vbdIface> hardware sgedbg`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 hardware sgedbg

SGE_DEBUG_DATA_HIGH00                   0x332200aa 857866410
SGE_DEBUG_DATA_HIGH01                   0x990055   10027093
SGE_DEBUG_DATA_HIGH02                   0          0
SGE_DEBUG_DATA_HIGH03                   0x4620     17952
SGE_DEBUG_DATA_HIGH04                   0xd020     53280
SGE_DEBUG_DATA_HIGH05                   0x8800     34816
SGE_DEBUG_DATA_HIGH06                   0          0
SGE_DEBUG_DATA_HIGH07                   0x111177aa 286357418
SGE_DEBUG_DATA_HIGH08                   0x200bb    131259
SGE_DEBUG_DATA_HIGH09                   0xbb0022   12255266
SGE_DEBUG_DATA_HIGH10                   0x5602c00  90188800
SGE_DEBUG_DATA_HIGH11                   0          0
SGE_DEBUG_DATA_HIGH12                   0          0
SGE_DEBUG_DATA_HIGH13                   0          0
SGE_DEBUG_DATA_HIGH14                   0x45475301 1162302209
SGE_DEBUG_DATA_HIGH15                   0xbab8acfe 3132665086

SGE_DEBUG_DATA_LOW00                    0          0
SGE_DEBUG_DATA_LOW01                    0x3f000    258048
SGE_DEBUG_DATA_LOW02                    0          0
SGE_DEBUG_DATA_LOW03                    0x60253001 1613049857
SGE_DEBUG_DATA_LOW04                    0x700026   7340070
SGE_DEBUG_DATA_LOW05                    0x6002c    393260
SGE_DEBUG_DATA_LOW06                    0x60066    393318
SGE_DEBUG_DATA_LOW07                    0x60401    394241
SGE_DEBUG_DATA_LOW08                    0x446      1094
SGE_DEBUG_DATA_LOW09                    0x604e1    394465
SGE_DEBUG_DATA_LOW10                    0xa50025   10813477
SGE_DEBUG_DATA_LOW11                    0x25       37
SGE_DEBUG_DATA_LOW12                    0x25       37
```

- **sensor**

Description: Display sensor data.

Syntax: `cxgbtool <nicIface> hardware sensor`

Example:

```
C:\Users\Administrator>cxgbtool nic0 hardware sensor
Temperature: 80C
Core VDD: 855mV
```

- **flash**

Description: Redirect Flash memory details to a file.

Syntax: `cxgbtool <vbdIface> hardware flash <file>`

Note *It is recommended that the file should be provided without any extension.*

Example:

```
C:\Users\Administrator>cxgbtool vbd0 hardware flash dump_flash
cxgbtool: This may take a while. Please be patient
cxgbtool: Reading Flash memory of size 4194304 bytes
cxgbtool: Writing memory block of size 4194304 bytes to file dump_flash...
cxgbtool: Done writing memory block to file dump_flash
```

- **mps_trc_rd trace**

Description: Read MPS trace filter.

Syntax: `cxgbtool <vbdIface> hardware mps_trc_rd trace<0|1|2|3>`

Example:

```
C:\Users\Administrator>cxgbtool vbd0 hardware mps_trc_rd trace0 tx0
tracer is capturing nic0 Tx, snap length: 0, min length: 0
packets captured match filter
filter pattern: 00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
```

- **mps_trc_wr trace**

Description: Set MPS trace filter.

Syntax:

`cxgbtool <vbdIface> hardware mps_trc_wr trace[0|1|2|3]`
`<tx{`*`portInstance`*`}|rx{`*`portInstance`*`}>`

Example:

Setting up trace0 with tx0 (tx0 is port0 Tx path)

```
C:\Users\Administrator>cxgbtool vbd0 hardware mps_trc_wr trace0 tx0
```

- **tid_info**

Description: Display TID information

Syntax:

```
cxgbtool <nicIface> hardware tid_info
```

Example:

```
C:\Users\Administrator>cxgbtool nic0 hardware tid_info
TID range: 0..2559/3072..19455, in use: 0
STID range: 2560..2687, in use: 5
ATID range: 0..4095, in use: 0
FTID range: 2688..3055
HW TID usage: 0 IP users, 0 IPv6 users
```

- **fec**

Description: Set/Display Forward Error Correction (FEC) parameters.

Syntax:

```
cxgbtool <nicIface> hardware fec [<config params>]
```

Example:

- Displaying currently set FEC mode.

```
C:\Users\Administrator>cxgbtool nic0 hardware fec
supported:   : Base-R/Reed-Solomon Reed-Solomon
advertising:: no FEC
```

- Setting FEC mode:

```
C:\Users\Administrator>cxgbtool nic0 hardware fec rs

C:\Users\Administrator>cxgbtool nic0 hardware fec
supported:   : Base-R/Reed-Solomon Reed-Solomon
advertising:: Reed-Solomon
```

- **link_cfg**

Description: Set/Display link configuration parameters.

Syntax:

```
cxgbtool <nicIface> hardware link_cfg <options>
```

Example:

- Displaying link configuration information:

```
C:\Users\Administrator>cxgbtool nic0 hardware link_cfg
Link Configuration
 Supported ports                : [ FIBRE ]
 Supported link modes           : 1000baseT/Full
                                  10000baseT/Full
                                  25000baseCR/Full
 Supported auto-negotiation     : Yes

 Advertised link modes          : 1000baseT/Full
                                  10000baseT/Full
                                  25000baseCR/Full
 Advertised auto-negotiation    : Yes

 Speed                          : 25Gbps
 port                           : Other
 Auto-negotiation               : On
```

- Enabling/Disabling auto-negotiation:

Auto-negotiation is enabled by default. To disable it, use the *autoneg off* parameter on all connected machines of the link.

```
C:\Users\Administrator>cxgbtool nic0 hardware link_cfg autoneg off

C:\Users\Administrator>cxgbtool nic0 hardware link_cfg
Link Configuration
        Supported auto-negotiation     : Yes
        Advertised auto-negotiation    : No
        Speed                          : 25Gbps
        Auto-negotiation               : Off
        Link State                     : Disconnected
```

- Changing link speed:

25G port can be set to 10G and 100G port can be set to 25G/40G/50G. Auto-negotiation should be disabled on all connected machines while changing link speed.

```
C:\Users\Administrator>cxgbtool nic0 hardware link_cfg autoneg off speed 10000

C:\Users\Administrator>cxgbtool nic0 hardware link_cfg
Link Configuration
        Supported auto-negotiation     : Yes
        Advertised auto-negotiation    : No
        Speed                          : 10Gbps
        Auto-negotiation               : Off
        Link State                     : connected
```

- **ipfilter**

Description: Set/Display IP filter rules.

Syntax:

```
cxgbtool <nicIface> hardware ipfilter <options> [<actions> <params>]
```

Examples:

- Setting filter rules:

```
C:\Users\Administrator>cxgbtool nic0 hardware ipfilter set -action drop -fip 102.1.1.2
```

- Listing configured filter rules:

```
C:\Users\Administrator>cxgbtool nic0 hardware ipfilter show

Index      Type       ACTION     IPORT      EPORT      FIP             FIP-MASK        LIP             LIP-MASK        Locked     Hits

0          IPv4       DROP       0          0          102.001.001.002  255.255.255.255 000.000.000.000  000.000.000.000  1          0
1          IPv4       DROP       0          0          102.002.002.002  255.255.255.255 000.000.000.000  000.000.000.000  1          0
```

- Removing filter rules:

```
C:\Users\Administrator>cxgbtool nic0 hardware ipfilter remove 0
```

RDMA

Description: Display RDMA information.

Syntax: `cxgbtool rdma -d [iWARPAdapterIndex] <options> | -l`

Examples:

- Displaying RDMA statistics

```
C:\Users\Administrator>cxgbtool rdma -s
   Object:      Total    Current        Max       Fail
     PDID:      65536          0          2          0
      QID:      24576          8          8          0
   TPTMEM:   78951552          0       2112          0
   PBLMEM:  536870912          0      16384          0
   RQTMEM:  157903104          0     131072          0
  OCQPMEM:          0          0          0          0
  DB FULL:          0
 DB EMPTY:          0
  DB DROP:          0
 DB State: NORMAL Transitions 0
TCAM_FULL:          0
ACT_OFLD_CONN_FAILS:          0
PAS_OFLD_CONN_FAILS:          0
AVAILABLE IRD:     622592
```

- Displaying RDMA stags

```
C:\Users\Administrator>cxgbtool rdma -t
0x1e000
0x1ed00
0x1fb00
0x1fc00
0x20100
0x20e00
0x21c00
0x21d00
0x22300
0x22400
0x22900
0x23000
0x23900
0x23a00
0x23f00
0x24400
0x24500
0x24d00
0x24e00
0x25700
0x26600
0x26800
0x27000
0x27600
0x27e00
0x28c00
0x29400
```

- Displaying RDMA queue pairs (qps):

```
C:\Users\Administrator>cxgbtool rdma -d 0 -q
rc qp sq id 1028 in_use 0 rq id 1029 in_use 256 state 1 onchip 0 ep tid 9952 state 7 102.1.7.2:5445 -> 102.1.7.3:1024
rc qp sq id 1030 in_use 0 rq id 1031 in_use 256 state 1 onchip 0 ep tid 4944 state 7 102.1.7.2:5445 -> 102.1.7.3:1025

C:\Users\Administrator>cxgbtool rdma -d 1 -q
rc qp sq id 1032 in_use 0 rq id 1033 in_use 256 state 1 onchip 0 ep tid 9768 state 7 102.1.8.2:5445 -> 102.1.8.3:1026
rc qp sq id 1034 in_use 0 rq id 1035 in_use 256 state 1 onchip 0 ep tid 5016 state 7 102.1.8.2:5445 -> 102.1.8.3:1027
```

- Displaying RDMA endpoints:

```
C:\Users\Administrator>cxgbtool rdma -d 0 -e
ep FFFF8C808ABD9A90 cm_id FFFF888EDBFB5A10 state 1 flags 0x80 stid 2560 backlog 1024 102.1.7.2:5445
ep FFFF8A8896F0C5E0 cm_id FFFF8A8896B63C60 state 1 flags 0x80 stid 2568 backlog 1024 fe80:0000:0000:0000:7093:1096:46f5:69e5:5445
ep FFFF8489CE53E010 cm_id FFFF8489CE562AD0 state 7 flags 0x70 history 0x42b0 hwtid 9952 atid 0 102.1.7.2:5445 <-> 102.1.7.3:1024
ep FFFF8489CE322400 cm_id FFFF8C808B08F010 state 7 flags 0x70 history 0x42b0 hwtid 4944 atid 0 102.1.7.2:5445 <-> 102.1.7.3:1025
```

- Displaying list of iWARP adapters:

```
C:\Users\Administrator>cxgbtool rdma -l

        T6/T5 iwarp adapter 0
        T6/T5 iwarp adapter 1
```

- Starting krping server/client:

```
C:\Users\Administrator>cxgbtool rdma -d 0 -k "-s:102.1.7.2"
Starting krping in server mode:-
        ip: 102.1.7.2
        port: 9999
        max_conns: 1
        bufsz: 1048576
        mem_type: 1
        write to file: 0
cr:1    cs:1    dis:1   ce:0    sb:40   rb:40   rr:65536        rw:0
```

```
C:\Users\Administrator>cxgbtool rdma -d 0 -k "-c:102.1.7.2 -rr"
Starting krping in client mode:-
        ip:
        port: 9999
        conns: 1
        iosz: 65536
        mem_type: 1
        op:1
        server invalidate: 0
        write to file: 0

Krping stats:-
         connect requests: 1
         connects: 1
         disconnects: 1
         connect errors: 0
         send bytes: 40 (1)
         recv bytes: 40 (1)
         read bytes: 0 (0)
         write bytes: 0 (0)
```

```
C:\Users\Administrator>cxgbtool rdma -d 0 -k "-c:102.1.5.2 -rw"
Starting krping in client mode:-
        ip:
        port: 9999
        conns: 1
        iosz: 65536
        mem_type: 1
        op:2
        server invalidate: 0
        write to file: 0

Krping stats:-
         connect requests: 1
         connects: 1
         disconnects: 1
         connect errors: 0
         send bytes: 40 (1)
         recv bytes: 40 (1)
         read bytes: 0 (0)
         write bytes: 0 (0)
```

```
C:\Users\Administrator>cxgbtool rdma -d 0 -k "-s:102.1.5.2"
Starting krping in server mode:-
        ip: 102.1.5.2
        port: 9999
        max_conns: 1
        bufsz: 1048576
        mem_type: 1
        write to file: 0
cr:1    cs:1    dis:1   ce:0    sb:40   rb:40   rr:0    rw:65536
```

```
C:\Users\Administrator>cxgbtool rdma -d 0 -k "-s:102.1.7.2"
Starting krping in server mode:-
        ip: 102.1.7.2
        port: 9999
        max_conns: 1
        bufsz: 1048576
        mem_type: 1
        write to file: 0

Krping stats:-
         connect requests: 1
         connects: 1
         disconnects: 1
         connect errors: 0
         send bytes: 40 (1)
         recv bytes: 40 (1)
         read bytes: 65536 (1)
         write bytes: 0 (0)
```

```
C:\Users\Administrator>cxgbtool rdma -d 0 -k "-s:102.1.5.2"
Starting krping in server mode:-
        ip: 102.1.5.2
        port: 9999
        max_conns: 1
        bufsz: 1048576
        mem_type: 1
        write to file: 0

Krping stats:-
         connect requests: 1
         connects: 1
         disconnects: 1
         connect errors: 0
         send bytes: 40 (1)
         recv bytes: 40 (1)
         read bytes: 0 (0)
         write bytes: 65536 (1)
```

8. Software/Driver Uninstallation

Similar to installation, Chelsio Unified Wire can be uninstalled using the Installer or zip package. Refer the relevant section depending on the method of installation used to install drivers.

8.1. Installer

Chelsio Unified Wire Installer can be uninstalled using two methods: **GUI** or **CLI** mode. GUI mode requires user interaction and uninstallation occurs with options specified by the user. Whereas, CLI mode does not require any user input.

8.1.1. GUI mode (Installer)

i. Run the **ChelsioUwire-x.x.x.xx.exe** application.
ii. Select **Modify** to add or remove features. Select **Repair** to repair the previous installation. Select **Remove** to uninstall the application. After you have selected the appropriate option, click **Next**.

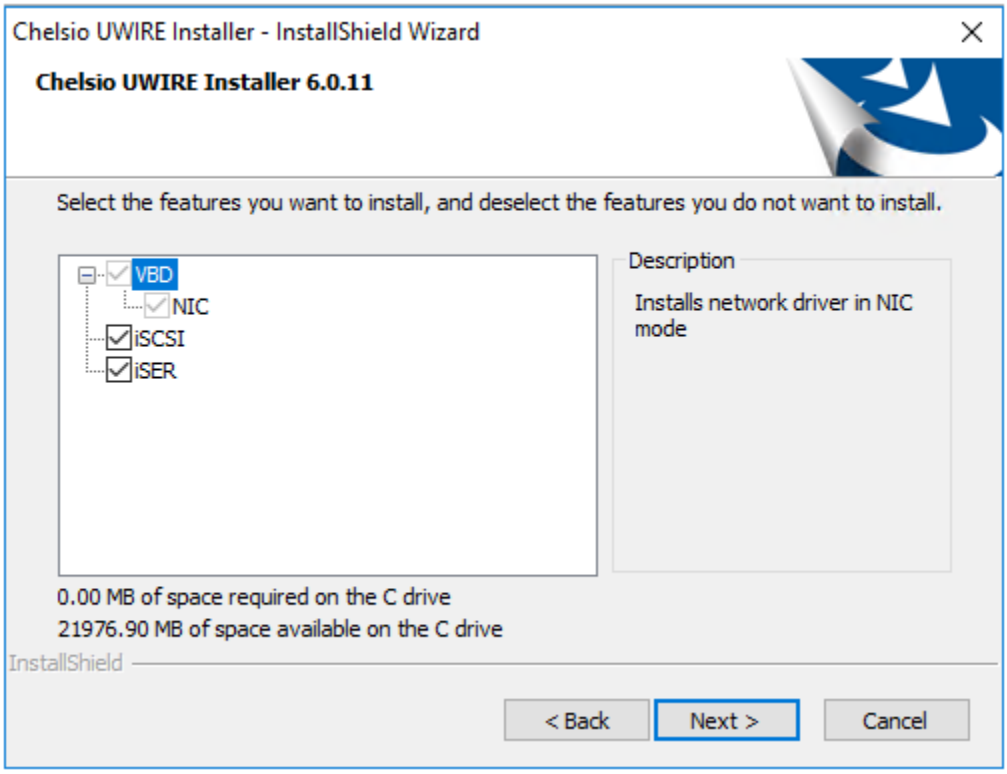


Figure 27 - Adding or removing features using 'Modify' option

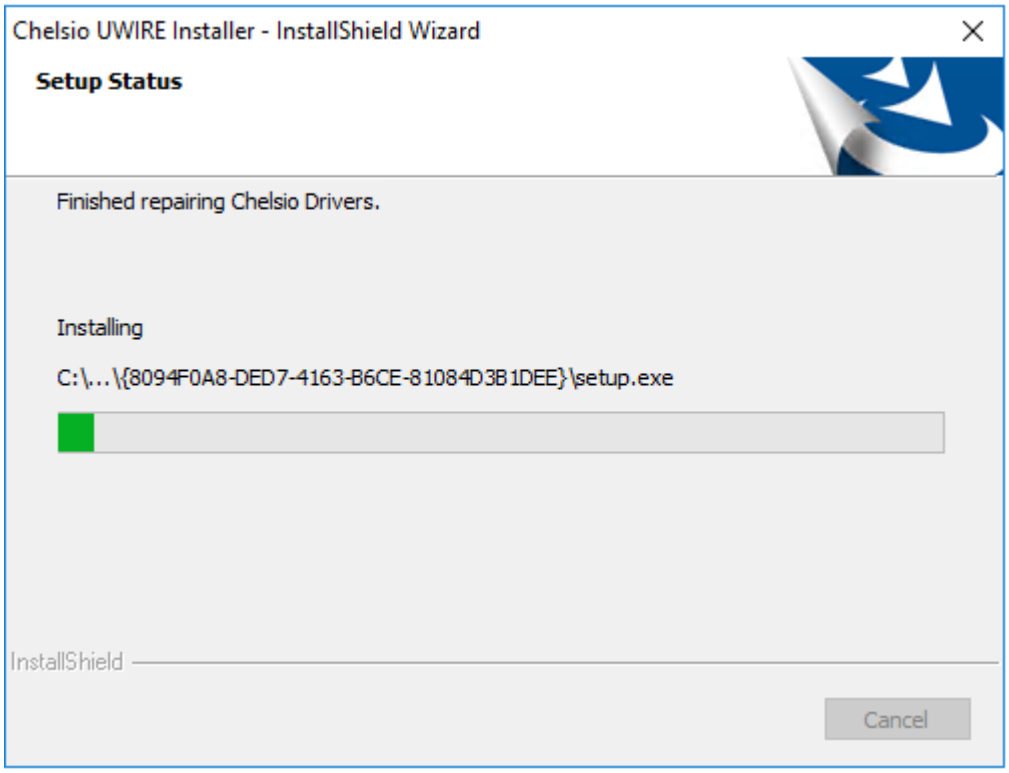


Figure 28 - Repairing previous installation using 'Repair' option

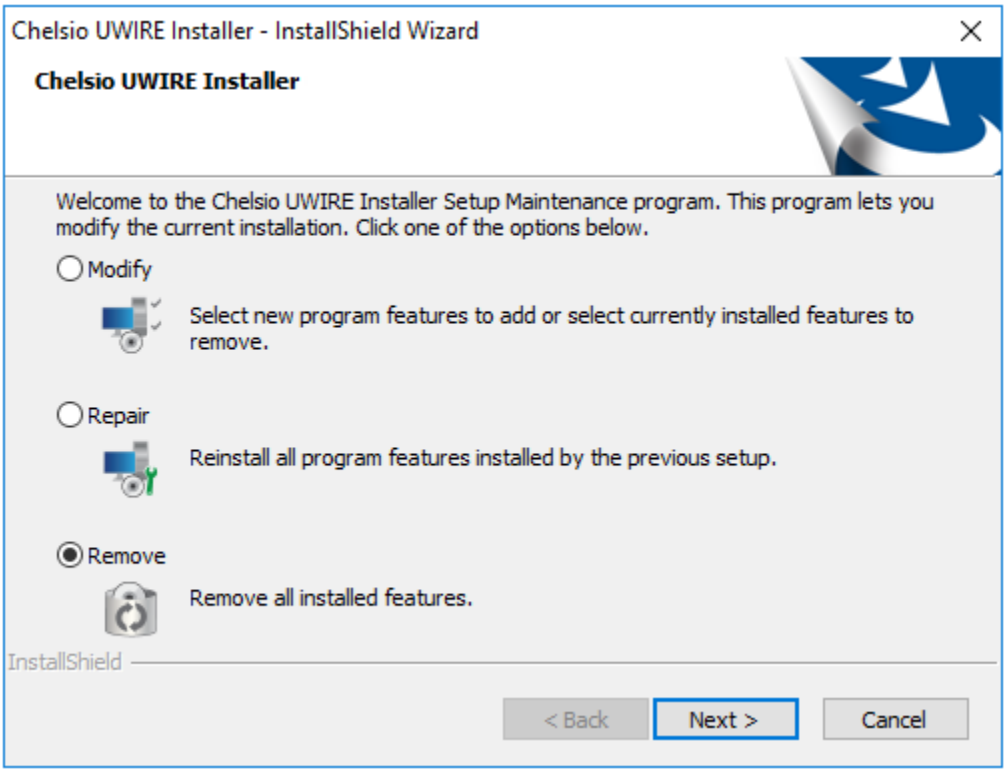


Figure 29 – Uninstalling Unified Wire using 'Remove' option

iii. Click on the **Finish** button to exit from the installer.

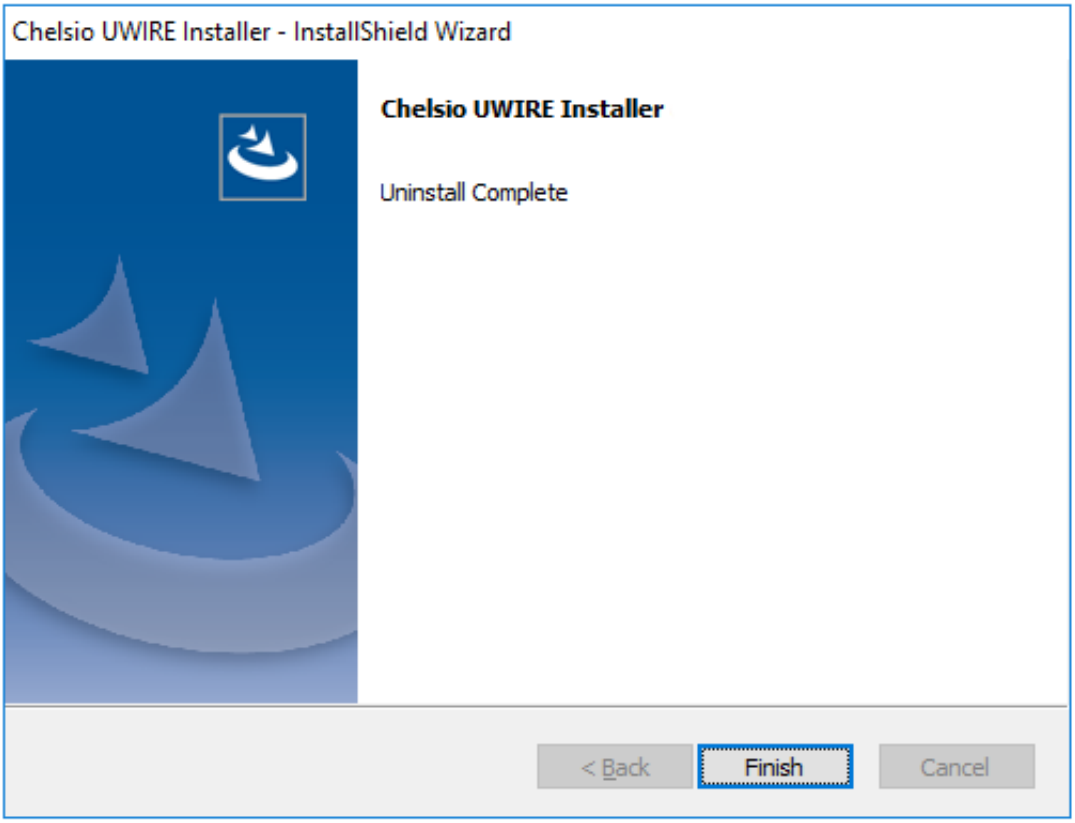


Figure 30 - Finishing uninstallation

Note *This method of uninstallation is possible only if the drivers were installed using Unified Wire Installer.*

8.1.2. CLI Mode

To uninstall all the drivers, execute the following command:

```
C:\Users\Administrator>ChelsioUwire-x.x.x.xx.exe -un all
```

To uninstall a particular driver, execute the following command:

```
C:\Users\Administrator>ChelsioUwire-x.x.x.x.exe -rm <driver(s)>
```

Example:

```
C:\Users\Administrator>ChelsioUwire-x.x.x.x.exe -rm iSCSI
```

The above command will uninstall iSCSI Storport Miniport driver.

8.2. Zip Package

If the driver components were installed using the zip package, they will have to be uninstalled manually. The following section lists the various components and their respective methods of uninstallation.

Important *The driver components need to be uninstalled strictly in the order described here. Using any other order is not supported.*

- ### NDIS Miniport driver

i. Open the **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on the **Network Adapters**, right click on the **Chelsio Network Adapter** and select **Uninstall.**

Note *In case of Virtual Functions in VM, use Chelsio VF Network Adapter.*

ii. Select the **Delete the driver software for this device** checkbox when **Confirm Device Uninstall** is prompted and click **OK.**

Repeat the same procedure for the other **Chelsio Network Adapters.**

- ### iSCSI Storport Miniport

i. Open the **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on **Storage controllers**, right-click on **Chelsio iSCSI Initiator** and select **Uninstall.**
ii. Select **Delete the driver software for this device** checkbox when **Confirm Device Uninstall** is prompted and click **OK.**

Repeat the same procedure for other ports**.**

- ### iSER Initiator

iii. Open the **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on **Storage controllers**, right-click on **Chelsio iSER Initiator** and select **Uninstall.**
iv. Select **Delete the driver software for this device** checkbox when **Confirm Device Uninstall** is prompted and click **OK.**

Repeat the same procedure for other ports.

- ## Virtual Bus Driver

i. Open the **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on the **System Devices**, right click on the **Chelsio Bus Enumerator** and select **Uninstall.**

> Note *In case of Virtual Functions in VM, use **Chelsio Bus Enumerator [Virtual Function].***

ii. Select the **Delete the driver software for this device** checkbox when **Confirm Device Uninstall** is prompted and click **OK.**

- ## Generic Function

i. Open the **Device Manager** (Control Panel -> System & Security-> System -> Device Manager), click on the **Network Adapters**, right click on the **Chelsio Generic Function** and select **Uninstall.**

ii. Select the **Delete the driver software for this device** checkbox when **Confirm Device Uninstall** is prompted and click **OK.**

Repeat the same procedure for the other **Generic Functions.**

II. NDIS Function

1. Introduction

Chelsio's Terminator series of Unified Wire adapters provide extensive support for NIC operation, including all stateless offload mechanisms for both IPv4 and IPv6 (IP, TCP and UDP checksum offload, LSO - Large Send Offload aka TSO - TCP Segmentation Offload, Network Direct and assist mechanisms for accelerating RSC - Receive Segment Coalescing).

1.1. Hardware Requirements

1.1.1. Supported Adapters

Following is the list of NDIS components and supported Chelsio adapters:

Table 1 - Chelsio adapters and supported NDIS components

Chelsio Adapter	NDIS Components
T62100-SO-CR	NIC
T6225-SO-CR	
T6425-CR	NIC, NVGRE, VXLAN, VMMQ/vRSS
T6225-LL-CR	
T62100-CR	NIC, NVGRE, VXLAN, PacketDirect, VMMQ/vRSS
T62100-LP-CR	
T6225-CR	
T580-CR	
T520-CR	
T580-LP-CR	
T580-OCP-SO	NIC, NVGRE, VXLAN
T520-OCP-SO	
T580-SO-CR	NIC, NVGRE, VXLAN, VMMQ/vRSS
T520-SO-CR	
T520-LL-CR	
T540-CR	
T520-BT	

1.2. Software Requirements

1.2.1. Windows Requirement

Following is the list of NDIS components and supported Windows versions:

Table 2 – Windows version and supported NDIS components

Windows Version	NDIS Components
Server 2016	NIC, NVGRE, VXLAN, PacketDirect, VMMQ/vRSS
Server 2012 R2	NIC, NVGRE
Server 2012	NIC
10 AU Client	

2. Software/Driver Configuration and Fine-tuning

2.1. Advanced Configuration

The Chelsio network driver provides advanced configuration options under the **Device Properties**. In the **Network Connections** window, select the Chelsio adapter's **Local Area Connection** interface (right-click -> properties). Click the **Configure** button.

2.1.1. VBD Driver Parameters

To see all tunable VBD driver parameters, open the **Device Manager**, click on the **System Devices**, double click on the **Chelsio Bus Enumerator**, and then click the **Advanced** tab.

- **iSCSI Instances**
  - *Description*: Specify the number of iSCSI instances.
  - *Value*: 0-2 | default
  - *Default*: 0
- **iSER Instances**
  - *Description*: Specify the number of iSER instances.
  - *Value*: 0-2 | default
  - *Default*: 0
- **NIC Instances:**
  - *Description*: Specify the number of NIC instances.
  - *Value*: 0-8 | default
  - *Default*: default

2.1.2. NDIS Miniport Driver Parameters

To see all tunable NDIS Miniport Driver parameters, open the **Device Manager**, click on the **Network adapters**, double click on any **Chelsio Network Adapter**, and then click the **Advanced** tab.

- **Encapsulated Task Offload**
  - *Description*: Enable or disable stateless offloads (checksums, LSO, VMQ and filtering) of encapsulated traffic.
  - *Value*: Disabled | Enabled
  - *Default*: Enabled
- **Encapsulated Overhead**
  - *Description*: Specify the amount of overhead required in Ethernet frames due to virtual network overlay encapsulation such as VXLAN and NVGRE.

- *Value*:
  - *32*
  - 64
  - 96
  - 128
  - 160
  - 192
  - 224
  - 256
- *Default*: 0

Flow Control

- *Description*: Enable or disable the receipt or transmission of PAUSE frames. If Rx is enabled, all incoming PAUSE frames will be honored. If Tx is enabled, the HW will send PAUSE frames when its MAC Rx FIFO usage is beyond a high-watermark. If Disabled option is selected, no PAUSE will be sent. When Rx MAC FIFO gets full, further ingress packets will be dropped. If Rx & Tx Enabled is selected, both Rx and Tx options will be enabled.
- *Value*:
  - Auto Negotiation
  - Disabled
  - Rx & Tx Enabled
  - Rx Enabled
  - Tx Enabled
- *Default*: Auto Negotiation

Interrupt Moderation

- *Description*: Control the interrupts generated by NIC hardware.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Interrupt Moderation Rate

- *Description*: Set the interrupt rate.
- *Value*:
  - Extreme
  - High
  - Low
  - Minimal
  - Moderate
- *Default*: Low

IPv4 Checksum Offload

- *Description*: Allow the checksum to be computed by the adapter for IPV4 packets.
- *Value*:

    - Disabled
    - Rx & Tx Enabled
    - Rx Enabled
    - Tx Enabled
- *Default*: Rx & Tx Enabled

Jumbo Packet (Maximum Transmission Unit)

- *Description*: Specify the Maximum Transmission Unit (MTU) value.
- *Value*:
    - 4088 Bytes
    - 9014 Bytes
    - Disabled
- *Default*: Disabled

Large Send Offload V2 (IPv4)

- *Description*: Allow for configuring Large Send Offload (LSO) using version 2 for the IPv4 traffic.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Large Send Offload V2 (IPv6)

- *Description*: Allow for configuring Large Send Offload (LSO) using version 2 for the IPv6 traffic.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Locally Administered Address

- *Description*: Specify a new MAC address for the port. If Not Present, the default MAC from EEPROM is used. If specified, the new MAC overwrites the default MAC. This specified MAC is persistent across machine reboot.
- *Value*: Not Present | Any legal 6-byte MAC address
- *Default*: Not Present

Maximum Number of RSS Processors

- *Description*: Change the number of RSS Processors.
- *Value*: 1-64
- *Default*: 8

Maximum Number of RSS Queues

- *Description*: Change the number of RSS Queues.
- *Value*: 1-16
- *Default*: 8

- ### Maximum RSS Processor Number
  - *Description*: Set the number of RSS processors to help the overall performance of the computer.
  - *Value*: 0-63
  - *Default*: 63
- ### NDIS QoS
  - *Description*: Enable or disable NDIS Quality of Service for DCB interface.
  - *Value*: Disabled | Enabled
  - *Default*: Disabled
- ### NetworkDirect Functionality
  - *Description*: Enable or disable Network Direct Functionality.
  - *Value*: Disabled | Enabled
  - *Default*: Enabled
- ### NetworkDirect Interrupt Moderation
  - *Description*: Control the interrupts generated during NDK traffic.
  - *Value*: Disabled | Enabled
  - *Default*: Enabled

  For more information, refer http://msdn.microsoft.com/en-us/library/windows/hardware/ff556017%28v=vs.85%29.aspx
- ### Nvgre Task Offload
  - *Description*: Enable or disable offloading processing of NVGRE encapsulated frames.
  - *Value*: Disabled | Enabled
  - *Default*: Enabled
- ### Packet Direct
  - *Description*: Enable or disable PacketDirect feature.
  - *Value*: Disabled | Enabled
  - *Default*: Disabled
- ### Preferred NUMA node
  - *Description*: The NUMA node that the adapter can allocate memory from.
  - *Value*:
    - Default Settings
    - Node 0
    - Node 1
    - Node 2
    - Node 3
    - Node 4
    - Node 5

    - Node 6
    - Node 7
- *Default*: Default Settings

Receive Side Scaling (RSS)

- *Description*: Control the RSS functions. If on, the Microsoft RSS function is enabled. If off, the Chelsio RSS function is enabled.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Recv Segment Coalescing (IPv4)

- *Description*: Enable or disable the Receive Segment Coalescing feature for IPv4 traffic. This is a receiver packets aggregating feature which helps reduce the receive host CPU load and improve throughput in a network environment where CPU can be the bottleneck.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Recv Segment Coalescing (IPv6)

- *Description*: Enable or disable the Receive Segment Coalescing feature for IPv6 traffic. This is a receiver packets aggregating feature which helps reduce the receive host CPU load and improve throughput in a network environment where CPU can be the bottleneck.
- *Value*: Disabled | Enabled
- *Default*: Enabled

RSS Base processor

- *Description*: Specify the number of the base RSS processors.
- *Value*: 0-63
- *Default*: 0

RSS load balancing profile

- *Description*: Control the RSS load balancing profile.
- *Value*:
    - Closest Processor
    - Closest Processor Static
    - Conservative Scaling
    - NUMA Scaling
    - NUMA Scaling Static
- *Default*: NUMA Scaling Static

For more information, refer http://technet.microsoft.com/en-us/library/hh997036.aspx

Rx Ethernet Queue Size

- *Description*: Set different Rx Queue sizes. Driver ships with reasonable defaults.
- *Value*: 512-4096
- *Default*: 2048

Rx Offload Queue Size

- *Description*: Set different Offload Rx Queue sizes. Driver ships with reasonable defaults.
- *Value*: 256-2048
- *Default*: 512

Note *This feature is not fully tested and available for experimental usage only.*

SR-IOV

- *Description*: Enable or disable SR-IOV feature.
- *Value*: Disabled | Enabled
- *Default*: Enabled

TCP Checksum Offload (IPv4):

- *Description*: Enable or disable computation of TCP checksum by the adapter for IPv4 packets.
- *Value*:
  - Disabled
  - Rx & Tx Enabled
  - Rx Enabled
  - Tx Enabled
- *Default*: Rx & Tx Enabled

TCP Checksum Offload (IPv6):

- *Description*: Enable or disable computation of TCP checksum by the adapter for IPv6 packets.
- *Value*:
  - Disabled
  - Rx & Tx Enabled
  - Rx Enabled
  - Tx Enabled
- *Default*: Rx & Tx Enabled

Tx Ethernet Queue Size

- *Description*: Set different Tx Queue sizes. Driver ships with reasonable defaults.
- *Value*: 512-4096
- *Default*: 1024

Tx Offload Queue Size

- *Description*: Set different Offload Tx Queue sizes. Driver ships with reasonable defaults.
- *Value*: 256-4096
- *Default*: 512

Note *This feature is not fully tested and available for experimental usage only.*

UDP Checksum Offload (IPv4)

- *Description*: Enable or disable computation of UDP checksum by the adapter for IPv4 packets.
- *Value*:
  - Disabled
  - Rx & Tx Enabled
  - Rx Enabled
  - Tx Enabled
- *Default*: Rx & Tx Enabled

UDP Checksum Offload (IPv6)

- *Description*: Enable or disable computation of UDP checksum by the adapter for IPv6 packets.
- *Value*:
  - Disabled
  - Rx & Tx Enabled
  - Rx Enabled
  - Tx Enabled
- *Default*: Rx & Tx Enabled

User Mode NetworkDirect

- *Description*: Enable or disable Microsoft's Remote Direct memory Access (RDMA) interface for high speed, low latency networks.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Virtual Machine Queues

- *Description*: Enable or disable the virtual machine queue feature on the interface.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Virtual Switch RSS

- *Description*: Enable or disable Virtual Switch Receive Side Scaling.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Note *If NDIS function driver is installed using Unified Wire installer, the vRSS feature will be enabled by default. However, if the driver is installed manually (using zip package), the feature will be disabled.*

- **VLAN Identifier**
  - *Description*: Specify the VLAN ID to be inserted.
  - *Value*: 0-4095
  - *Default*: 0
- **VMQ LookAhead Split**
  - *Description*: Enable or disable the ability to split receive buffers into lookahead and post-lookahead buffers.
  - *Value*: Disabled | Enabled
  - *Default*: Enabled
- **VMQ VLAN ID Filtering**
  - *Description*: Enable or disable the ability to filter network packets by using the VLAN identifier in the media access control (MAC) header.
  - *Value*: Disabled | Enabled
  - *Default:* Enabled
- **Vxlan Task Offload**
  - *Description*: Enable or disable offloading processing of VXLAN encapsulated frames.
  - *Value*: Disabled | Enabled
  - *Default*: Enabled
- **Vxlan UDP Port Number**
  - *Description*: Specify the current VXLAN UDP destination port number that is currently operational in the NIC.
  - *Value*: 1-65535
    Default: 4789

2.2. NVGRE Offload

Chelsio's Terminator based adapters are uniquely capable of offloading the processing of NVGRE encapsulated frames such that all stateless offloads (checksums, LSO, VMQ, RSS and filtering) are preserved, resulting in significant performance benefits. This feature is enabled by default. Configure the relevant customer and provider network settings on the host.

2.3. VXLAN Task Offload

Virtual Extensible LAN (VXLAN), an extension of the existing VLAN protocol, is an encapsulation protocol which creates an overlay network on the existing Layer 3 infrastructure. VXLAN not only extends VLAN's Ethernet Layer 2 network services but also provides more extensibility, flexibility and scalability for large scale cloud deployments than VLAN.

Chelsio's Terminator based adapters are capable of offloading the processing of VXLAN encapsulated frames such that all stateless offloads (checksums, LSO, VMQ, RSS and filtering) are preserved, resulting in significant performance benefits. This feature is enabled by default.

2.4. PacketDirect

PacketDirect Provider Interface (PDPI), an extension to the existing NDIS miniport driver model, gives applications complete control over buffers, poll processors, and sending and receiving packets over a miniport adapter.

Chelsio's Unified Wire adapters offer extensive support for this feature leading to a higher throughput performance. This feature is disabled by default.

Follow the steps mentioned below to enable and configure PacketDirect:

i. Open **Device Manager**, click on **Network adapters**, double click on **Chelsio Network Adapter**, and then click the **Advanced** tab.
ii. Select the *PackDirect* property from the list and enable it. Click OK

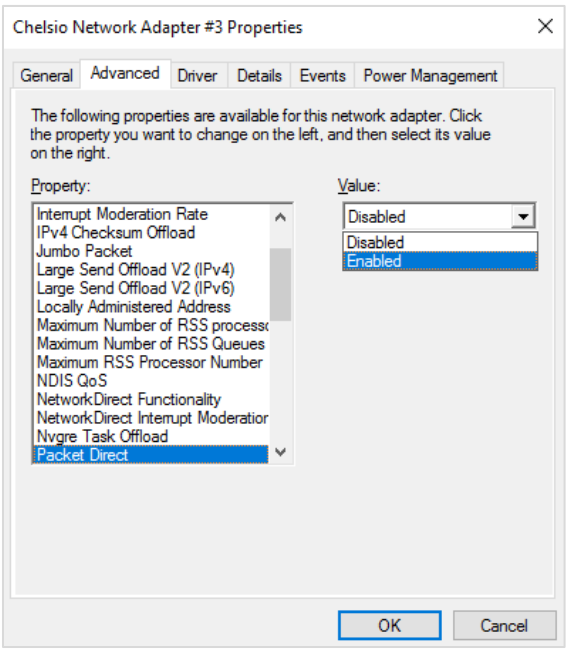


Figure 31 - Enabling Packet Direct

iii. Create a new virtual switch and enable PacketDirect on the host:

```
C:\Users\Administrator> New-VMSwitch -SwitchName <virtual_switch>
-NetAdapterName <chelsio_interface> -EnablePacketDirect $true
```

```
PS C:\Users\Administrator> New-VMSwitch -SwitchName sw0 -NetAdapterName "Ethernet 3" -EnablePacketDirect $true

Name SwitchType NetAdapterInterfaceDescription
---- ---------- ------------------------------
sw0  External   Chelsio Network Adapter
```

iv. Enable *Azure Virtual Filtering Platform (VFP) Switch Extension* on the switch.

```
C:\Users\Administrator> Enable-VMSwitchExtension -VMSwitchName
<virtual_switch> -Name "Microsoft Azure VFP Switch Extension"
```

```
PS C:\Users\Administrator> Enable-VMSwitchExtension -VMSwitchName sw0 -Name "Microsoft Azure VFP Switch Extension"


Id                : E9B59CFA-2BE1-4B21-828F-B6FBDBDDC017
Name              : Microsoft Azure VFP Switch Extension
Vendor            :
Version           :
ExtensionType     : Forwarding
ParentExtensionId :
ParentExtensionName :
SwitchId          : 48f433e2-10ad-447a-8566-bb4864dbcb12
SwitchName        : sw0
Enabled           : True
Running           : True
CimSession        : CimSession: .
ComputerName      : DUKE2
IsDeleted         : False
```

v. Add a virtual network adapter to the virtual machine and connect it to the virtual switch:

```
C:\Users\Administrator> Add-VMNetworkAdapter -VMName <virtual_machine>
-SwitchName <virtual_switch> -Name <virtual_adapter>
```

```
C:\Users\Administrator> Add-VMNetworkAdapter -VMName VM1 -SwitchName sw0 -Name pd_nw1
C:\Users\Administrator>
```

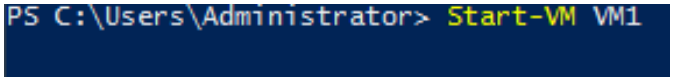
vi. Set the number of PacketDirect processors:

```
C:\Users\Administrator> Set-VMNetworkAdapter -VMName <virtual_machine>
-PacketDirectNumProcs <1-8>
```

```
PS C:\Users\Administrator> Set-VMNetworkAdapter -VMName VM1 -PacketDirectNumProcs 4
PS C:\Users\Administrator>
```

vii. Start the VM:

```
C:\Users\Administrator> Start-VM <virtual_machine>
```



viii. On the host, create a PowerShell script (with *.ps1* extension) with the following code:

```
param(
    [string]$switchName = $(throw "please specify a switch name")
    )

$switches = Get-WmiObject -Namespace root\virtualization\v2 -Class
Msvm_VirtualEthernetSwitch
foreach ($switch in $switches) {
               if ( $switch.ElementName -eq $switchName) {
                              $ExternalSwitch = $switch
                              break
               }
}

$vfpCtrlExe = "vfpctrl.exe"
$ports = $ExternalSwitch.GetRelated("Msvm_EthernetSwitchPort",
"Msvm_SystemDevice", $null, $null, $null, $null, $false, $null)
foreach ($port in $ports) {
#if ($port.ElementName -eq "Dynamic Ethernet Switch Port")
#{
    $portGuid = $port.Name
    echo "Disabling VFP on port: " $portGuid
    & $vfpCtrlExe  /disable-port /port $portGuid
#}

}
```

ix. Run the script:

```
PS C:\Users\Administrator> <script>.ps1 <virtual_switch>
```

```
PS C:\Users\Administrator> .\disableVFP.ps1 sw0
Disabling VFP on port:
0D46B798-B405-40FE-86BC-6C67A870F368
ERROR: failed to execute disable-port
Error (1): Incorrect function.
Disabling VFP on port:
670970D3-23AF-45B0-8741-422C79F2B3DC
ERROR: failed to execute disable-port
Error (1): Incorrect function.
Disabling VFP on port:
C0C0B3ED-F097-4815-AD12-66B0492D360D
Command disable-port succeeded!
Disabling VFP on port:
D9049B8A-D9AA-4D08-B746-D0960EC25848
Command disable-port succeeded!
PS C:\Users\Administrator>
```

Note *The errors seen in the output are expected.*

x. Use system performance monitoring utilities, such as *PerfMon*, to add PacketDirect counters and verify if they are updated.

PacketDirect Transmit Counters	EC 1300: IfIndex 6: VPort 1: TX 3:	EC 1300: IfIndex 6: VPort 2: TX 8:	EC 2604: IfIndex 6: VPort 2: TX 6:	EC 2848: IfIndex 6: VPort 2: TX 5:	EC 4312: IfIndex 6: VPort 0: TX 1:	EC 4312: IfIndex 6: VPort 2: TX 7:
Bytes Transmitted	33,969.000	132.000	1,160.000	30,439.000	0.000	344.000
Bytes Transmitted/sec	0.000	0.000	0.000	0.000	0.000	0.000
Packets Transmitted	135.000	2.000	14.000	96.000	0.000	4.000
Packets Transmitted/sec	0.000	0.000	0.000	0.000	0.000	0.000

Processor Information	_Total
% Processor Time	0.473

2.5. VMMQ/vRSS

Network throughput of a multi-core virtual machine is limited by the processing power of a single virtual processor, since only one processor handles all the interrupts from a virtual network adapter. Virtual Machine Multi-Queue (VMMQ) or Virtual Switch Receive-side Scaling (vRSS) solves this bottleneck by distributing the network traffic across multiple virtual processors.

If NDIS function driver is installed using Unified Wire installer, VMMQ/vRSS will be enabled by default. However, if the driver is installed manually (using zip package), the feature will be disabled. Use **NDIS Miniport Driver Parameters** (Device manager → Chelsio Network Adapter Properties → Advanced Tab → Virtual Switch RSS) to enable it. Additionally, VMMQ/vRSS must be enabled for each VM on the host using:

```
PS C:\Users\Administrator> Set-VMNetworkAdapter -VMName <VM Name>
-VrssEnabled:$true -VmmqEnabled:$true -VmmqQueuePairs <No. of Queues>
```

Example:

```
PS C:\Users\Administrator> Set-VMNetworkAdapter -VMName vm1 -VrssEnabled 1 -VmmqEnabled 1 -VmmqQueuePairs 8
PS C:\Users\Administrator> Get-VMNetworkAdapter -VMName vm1 | fl | findstr "Enabled"
DynamicMacAddressEnabled               : True
VrssEnabledRequested                   : True
VrssEnabled                            : True
VmmqEnabledRequested                   : True
VmmqEnabled                            : True
```

- ## Dynamic VMMQ/RSSv2

Note *Available on Windows Server 2016 version 1709 and later.*

The latest update, Dynamic VMMQ/RSSv2, makes RSS algorithm dynamic by taking CPU measurement and making balancing decision at DISPATCH level. The feature also avoids taking global locks by changing only the subset of indirection table which affects current CPUs only. The feature will be enabled/disabled depending on how NDIS function driver was installed, as described previously. Additionally, the feature must be enabled for each VM on the host using:

```
PS C:\Users\Administrator> Set-VMNetworkAdapter -VMName <VM Name>
-VrssEnabled 1 -VmmqEnabled 1 -VmmqQueuePairs <No. of Queues>
-VrssQueueSchedulingMode Dynamic
```

Example:

```
PS C:\Users\Administrator> Set-VMNetworkAdapter -VMName vm1 -VrssEnabled 1 -VmmqEnabled 1 -VmmqQueuePairs 8
-VrssQueueSchedulingMode Dynamic
PS C:\Users\Administrator> Get-VMNetworkAdapter -VMName vm1 | fl | findstr "VrssQueueSchedulingMode"
VrssQueueSchedulingModeRequested       : Dynamic
VrssQueueSchedulingMode                : Dynamic
```

III. iWARP (ND)

1. Introduction

ND or Network Direct is Microsoft's new Remote Direct memory Access (RDMA) interface for high speed, low-latency networks such as those running on 10/25/40/50/100 Gigabit Ethernet or InfiniBand, with an architecture that directly bypasses OS and TCP/IP overhead. ND achieves better performance for massively parallel programs that can utilize very low-latency, high-bandwidth, and enables efficient CPU utilization.

1.1. Hardware Requirements

1.1.1. Supported Adapters

The following are the currently shipping Chelsio adapters that are compatible with Chelsio iWARP (ND) driver:

- T62100-LP-CR
- T6225-CR
- T580-CR
- T580-LP-CR
- T540-CR
- T520-CR
- T520-LL-CR
- T520-BT

1.2. Software Requirements

1.2.1. Windows Requirement

Currently Chelsio iWARP (ND) driver is available for the following Windows version(s):

- Server 2016
- Server 2012 R2

Other versions have not been tested and are not guaranteed to work.

2. Software/Driver Configuration and Fine-tuning

2.1. Registering iWARP(ND) driver

Chelsio Unified Wire Installer registers Chelsio iWARP provider automatically during installation.

In case of zip package, run the following command to register:

```
C:\Users\Administrator>chinstallsp.exe -i
```

The iWARP provider should show up using:

```
C:\Users\Administrator>chinstallsp.exe -l
```

You should see a similar output:

```
0000001019 - Chelsio Network Direct provider
```

IV. SMB Direct

1. Introduction

SMB Direct is an extension of the Server Message Block (SMB) technology by Microsoft used for file operations. The *Direct* part implies the use of various high speed Remote Data Memory Access (RDMA) methods to transfer large amounts of data with little CPU intervention. By using RDMA as a transport medium for SMB, unprecedented levels of performance and efficiency can be achieved. With fully offloaded RDMA support, Chelsio Unified Wire adapters deliver large performance and efficiency gains to Windows users in a seamless, plug and play fashion.

1.1. Hardware Requirements

1.1.1. Supported Adapters

The following are the currently shipping Chelsio adapters that are compatible with Chelsio SMB Direct and RDMA/NVGRE concurrent (Mode 2) driver:

- T62100-CR
- T62100-LP-CR
- T6425-CR
- T6225-CR
- T6225-LL-CR
- T6225-SO-CR (*Memory-free; limited number of offload connections supported*)
- T580-CR
- T580-LP-CR
- T540-CR
- T520-CR
- T520-LL-CR
- T520-BT

1.2. Software Requirements

1.2.1. Windows Requirement

Following is the list of SMB Direct components and supported Windows versions:

Windows Version	SMB Direct Components
Server 2016	SMB Direct, Mode 2, SR
10 AU Client	SMB Direct
Server 2012 R2	

2. Software/Driver Configuration and Fine-tuning

2.1. Enabling SMB Direct

- ### Windows Server 2016/Server 2012 R2

SMB Direct is installed and enabled by default on Windows Server 2016/Server 2012 R2. RDMA functionality on Chelsio adapters will be enabled after installing the driver from the package.

- ### Windows 10 AU Client

On Windows 10 AU Client systems, SMB Direct feature is installed but disabled by default. To enable it, follow the steps mentioned below:

i. Open **Control Panel** and click **Programs**.
ii. Under **Programs and Features**, click **Turn Windows features on or off**.
iii. In the **Windows Features** window, scroll down and select the *SMB Direct* checkbox.
iv. Click **OK**.

RDMA functionality on Chelsio adapters will be enabled after installing the driver from the package.

- ### Configuration

You can use the PowerShell command `Get-NetAdapterRdma` to get a list of RDMA capable interfaces in your system. At least one adapter should indicate “RDMA Capable” true to take advantage of SMB RDMA capability.

The next step is to configure your network so that SMB server and client machines can communicate with each other. Please configure the Microsoft firewall to allow traffic between the SMB clients and server.

2.2. Verifying RDMA

There are number of ways to verify if RDMA is working:

2.2.1. Verify active connections using PowerShell

Open PowerShell window and execute the command `Get-SmbConnection`

2.2.2. Use Event Viewer

You can use Event Viewer to view SMB connection events. To do this:

i. Open Event Viewer
ii. In the console tree, expand **Event Viewer.**
iii. Navigate to **Application and Service Logs**-> **Microsoft**->**Windows**->**SMBClient** ->**Operational**

2.3. RDMA/NVGRE concurrent (Mode 2)

In this mode, you can run RDMA traffic using a virtual switch. This feature is enabled by default in the driver, but must be enabled on the Hyper-V Virtual Ethernet adapter. Follow the steps mentioned below to enable the feature:

i. If you haven't done already, run the Chelsio Unified Wire Installer which will install NDIS function driver.
ii. Verify if the driver is installed and loaded using the *Get-NetAdapter* command. The *InterfaceDescription* field should list Chelsio adapter as shown in the image below:

```
PS C:\Users\Administrator> Get-NetAdapter

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
vEthernet (p0)            Hyper-V Virtual Ethernet Adapter             18 Up           00-07-43-04-B4-20       25 Gbps
Ethernet 4                Chelsio Network Adapter #2                    9 Up           00-07-43-04-B4-28       25 Gbps
Ethernet 3                Chelsio Network Adapter                       2 Up           00-07-43-04-B4-20       25 Gbps
Ethernet 2                Intel(R) 82574L Gigabit Network Co...#2       8 Disconnected 00-25-90-35-95-FB        0 bps
Ethernet                  Intel(R) 82574L Gigabit Network Conn...      11 Not Present  00-25-90-35-95-FA        0 bps
Local Area Connection* 1  Microsoft Kernel Debug Network Adapter      10 Up           00-25-90-35-95-FA        1 Gbps
```

Figure 32 - Verifying Chelsio driver

iii. Next, create a virtual switch using the following syntax:

```
PS C:\Users\Administrator> New-VMSwitch –Name <virtual_switch>
-NetAdapterName <chelsio_interface>
```

```
PS C:\Users\Administrator> New-VMSwitch -Name switch0 -NetAdapterName "Ethernet 4"

Name     SwitchType NetAdapterInterfaceDescription
----     ---------- ------------------------------
switch0 External   Chelsio Network Adapter #2
```

Figure 33 - Creating virtual switch

iv. Using *Get-NetAdapter* command, verify if the virtual switch is created successfully. The Mac address of the switch and the port on which the switch was created, should be same.

```
PS C:\Users\Administrator> Get-NetAdapter

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
vEthernet (switch0)       Hyper-V Virtual Ethernet Adapter #4           3 Up           00-07-43-04-B4-28       25 Gbps
vEthernet (p0)            Hyper-V Virtual Ethernet Adapter             18 Up           00-07-43-04-B4-20       25 Gbps
Ethernet 4                Chelsio Network Adapter #2                    9 Up           00-07-43-04-B4-28       25 Gbps
Ethernet 3                Chelsio Network Adapter                       2 Up           00-07-43-04-B4-20       25 Gbps
Ethernet 2                Intel(R) 82574L Gigabit Network Co...#2       8 Disconnected 00-25-90-35-95-FB        0 bps
Ethernet                  Intel(R) 82574L Gigabit Network Conn...      11 Not Present  00-25-90-35-95-FA        0 bps
Local Area Connection* 1  Microsoft Kernel Debug Network Adapter      10 Up           00-25-90-35-95-FA        1 Gbps
```

Figure 34 - Verifying virtual switch

v. Enable RDMA on the virtual switch using *Enable-NetAdapterRdma* command.

```
PS C:\Users\Administrator> Enable-NetAdapterRdma
PS C:\Users\Administrator> _
```

Figure 35 - Enabling RDMA on virtual switch

vi. Run *Get-NetAdapterRdma* command to verify if RDMA is enabled on the switch.

```
PS C:\Users\Administrator> Get-NetAdapterRdma

Name                      InterfaceDescription                    Enabled
----                      --------------------                    -------
vEthernet (switch0)       Hyper-V Virtual Ethernet Adapter #4     True
vEthernet (p0)            Hyper-V Virtual Ethernet Adapter        True
Ethernet 4                Chelsio Network Adapter #2              True
Ethernet 3                Chelsio Network Adapter                 True
```

Figure 36 - Verifying RDMA on virtual switch

2.4. Storage Replica

Storage Replica (SR) is a Windows Server 2016 feature which enables block-level replication between clusters or individual servers for disaster recovery, and stretching of failover clusters to metropolitan (MAN) and wide area (WAN, US coast-to-coast) distances for high availability. SR provides two modes of operation: *synchronous* and *asynchronous* replication. Synchronous replication enables mirroring of data with zero data loss at the volume level, whereas asynchronous replication trades off full data replication guarantees for reduced latency by locally completing I/O operations.

Currently, Unified Wire supports Server to Server replication mode. In this mode, data is synced between two servers and each server keeps an identical copy of the same volume. For more information on how to setup and configure in this mode, please refer to Microsoft's official documentation.

2.5. HMA

The Host Memory Access(HMA) implements a bridge between the Memory Arbiter (MA) and the PCIE so that the Host Memory is available to all clients. The translation from the MA memory map to the Host Memory map is performed based on the client number and the address of the transaction, both of which are used as index to look up a 64-bit offset within the Host Memory.

Currently 256 IPv4/128 IPv6 offload connections are supported on T6225-SO-CR adapter. You can view the number of connections offloaded by using

```
C:\Users\Administrator> cxgbtool <nicIface> hardware tid_info
```

```
PS C:\> cxgbtool nic0 hardware tid_info
TID range: 0..255, in use: 128
STID range: 256..319, in use: 10
ATID range: 0..127, in use: 0
FTID range: 320..687
HW TID usage: 0 IP users, 128 IPv6 users
```

2.6. Troubleshooting

If RDMA is not working in your system, please check for the following:

- Operating system version should be Windows Server 2016, Server 2012 R2 or 10 AU Client.
- Verify network interface configurations on both sides.
- Please ensure that the interfaces are reachable using *ping* command.
- In the Event Viewer (on the client side), please ensure that the traffic running is not over TCP.
- Verify that SMB driver is loaded and working. Open PowerShell window and execute the following command:

```
C:\Users\Administrator>driverquery.exe | findstr \I smb
```

- Verify that Chelsio driver is loaded and working. Open PowerShell window and execute the following command:

```
C:\Users\Administrator>driverquery.exe | findstr \I chel
```

- Confirm that traffic is running over Chelsio adapter.
- Look for potential errors in Event Log.

V. Ring Backbone

1. Introduction

Chelsio Ring Backbone Utility is an easy to use tool developed to configure machines (with Chelsio adapters) in a ring backbone. This replaces the need for a ToR switch to connect a set of servers.

1.1. Hardware Requirements

1.1.1. Supported Adapters

The following are the currently shipping Chelsio adapters that are compatible with the Chelsio Ring Backbone utility:

- T62100-CR
- T62100-LP-CR
- T62100-SO-CR*
- T6225-CR
- T6225-LL-CR
- T6225-SO-CR (*Memory-free; limited number of offload connections supported*)
- T580-CR
- T580-LP-CR
- T580-SO-CR*
- T520-CR
- T520-LL-CR
- T520-SO-CR*
- T520-BT

** NIC only supported*

1.2. Software Requirements

Currently the Ring Backbone utility is available for the following Windows version(s):

- Server 2016

Other versions have not been tested and are not guaranteed to work.

1.3. Prerequisites

Please ensure that the following requirements are met before proceeding with installation:

- There must be an active management network (apart from Chelsio test network) which enables the machines to be accessed to carry out the required configuration.

- PowerShell remoting should be enabled on each machine in the topology. Use the following command:

```
PS C:Users\Administrator> Enable-PSRemoting -Force
```

- Windows Remote Management (WinRM) should be enabled on each machine in the topology. Use the following command:

```
PS C:Users\Administrator> Set-item wsman:localhost\client\trustedhosts -
value *
```

- Firewall should be disabled from Control Panel.

2. Software/Driver Installation

i. Ensure that Unified Wire is installed on all the machines in the ring backbone. Refer to Chelsio Unified Wire chapter.

ii. In case of Installer, the *backbone_cfg.ps1* script and *cxgbtool.exe* will be copied to *<system_drive>\Windows\System32* folder during installation.
If you are using the zip package, copy *backbone_cfg.ps1* script from *<zip_package>\RingBackbone* and *cxgbtool.exe* from *<zip_package>\drivers* to *<system_drive>\Windows\System32* folder.

iii. All the nodes should have the same user credentials.

3. Software/Driver Configuration and Fine-tuning

Important

- *Port 1 should be not configured for any other purpose in the operating system.*
- *Only Single port (port 0) is available to the user after configuring ring backbone.*

i. Connect the machines in ring backbone. Connect port 1 of one machine to the port 0 of next machine such that a closed ring is formed.
ii. Open PowerShell and run the *backbone_cfg.ps1* script on any one of the machines to configure ring backbone.

```
PS C:\Users\Administrator> backbone_cfg.ps1 -Servers <host1,host2,host3> -
Username <username> -Password <password> BaseIpAddress <test_ip>
```

Example:

3 servers M1, M2 and M3 are connected in a ring backbone and configure with network address 102.1.1.0 with netmask 24 (selected by default).

```
PS C:\Users\Administrator> backbone_cfg.ps1 -Servers M1,M2,M3 -Username
administrator -Password windows@123 -BaseIpAddress 102.1.1.0
```

Therefore, M1 will be assigned with IP 102.1.1.1, M2 with 102.1.1.2 and M3 with 102.1.1.3 respectively.

iii. You should be able to run traffic between the machines/nodes now.

Reconfiguring Ring Backbone

Ring Backbone must be reconfigured if

- any of the driver parameters is changed (driver will be reloaded).
- vSwitch is created.
- a new VM is added.

To reconfigure, run the following command on any node:

```
PS C:\Users\Administrator> backbone_cfg.ps1 -Servers <host1,host2> -Username
<username> -Password <password> -Reconfigure
```

host1, *host2*: Only affected hosts should be listed here.

Clear Configuration

To clear previously set configuration:

```
PS C:\Users\Administrator> backbone_cfg.ps1 -Servers <host1,host2,...,hostN>
-Username <username> -Password <password> -Clear
```

VI. NDIS SR-IOV

1. Introduction

The ever-increasing network infrastructure of IT enterprises has led to a phenomenal increase in maintenance and operational costs. IT managers are forced to acquire more physical servers and other data center resources to satisfy storage and network demands. To solve the Network and I/O overhead, users are opting for server virtualization which consolidates I/O workloads onto lesser physical servers thus resulting in efficient, dynamic and economic data center environments. Other benefits of Virtualization include improved disaster recovery, server portability, cloud computing, Virtual Desktop Infrastructure (VDI), etc. Chelsio's Unified Wire family of adapters deliver increased bandwidth, lower latency and lower power with virtualization features to maximize cloud scaling and utilization. The adapters also provide full support for PCI-SIG SR-IOV to improve I/O performance on a virtualized system.

1.1. Hardware Requirements

1.1.1. Supported Adapters

The following are the currently shipping Chelsio adapters that are compatible with the Chelsio NDIS SR-IOV components:

Chelsio Adapter	NDIS SR-IOV Components
T62100-CR	NDIS SR-IOV
T62100-SO-CR	
T6425-CR	
T6225-LL-CR	
T6225-SO-CR	
T580-SO-CR	
T520-LL-CR	
T520-SO-CR	
T520-BT	
T62100-LP-CR	NDIS SR-IOV, Guest RDMA
T6225-CR	
T580-CR	
T580-LP-CR	
T540-CR	
T520-CR	

1.2. Software Requirements

Following is the list of NDIS SR-IOV components and supported Windows versions:

Windows Version	NDIS Components
Server 2016	NDIS SR-IOV, Guest RDMA
Server 2012 R2	NDIS SR-IOV

Other versions have not been tested and are not guaranteed to work.

1.3. Prerequisites

SR-IOV should be enabled in the machine.

2. Software/Driver Configuration and Fine-tuning

Please ensure that Unified Wire is installed on the host before proceeding. Refer Software/Driver Installation section of the **Chelsio Unified Wire** chapter for step-by-step instructions.

2.1. Enabling SR-IOV

Important

*The adapter's configuration should be updated to make use of the SR-IOV feature. You can do this by running the **chelsio_adapter_config.ps1** configuration script. To use the adapter in non-SRIOV scenarios in Windows or with other operating systems, the configuration type must be changed to **NON-SRIOV (Default)**.*

2.1.1. Windows GUI Machine

In case of Installer, the script will be copied to *<system_drive>\Windows\System32* during installation.

If you are using the zip package, copy *cxgbtool.exe* from *ChelsioUwire-x.x.x.xx* folder to *<system_drive>\Windows\System32.* Also, change your working directory to *ChelsioUwire-x.x.x.xx\Adapter Configuration*.

i. Open PowerShell with administrative privileges.
ii. Check if the machine is SR-IOV capable:

```
PS C:\Users\Administrator> Get-VMHost | fl *iov*
```

The *IovSupport* field should display “True” as shown in the image below:

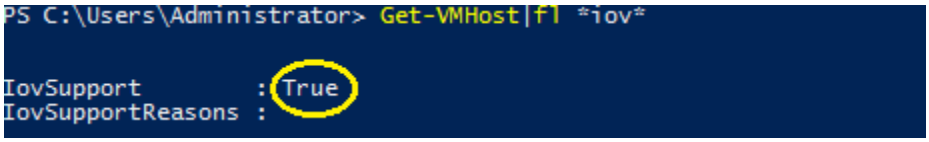


Figure 37 - Checking SR-IOV capability

iii. Run the adapter configuration script and enter the index of the adapter for which the configuration needs to be updated. Hit [Enter].

Installer:

```
PS C:\Users\Administrator> chelsio_adapter_config.ps1
```

Zip Package:

```
PS D:\ChelsioUwire-x.x.x.xx\Adapter Configuration>
.\chelsio_adapter_config.ps1
```

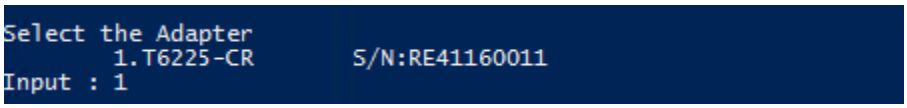


Figure 38 - Selecting adapter

iv. Select *SRIOV* (option 2) as the configuration type and enter *y* to confirm. Hit [Enter].

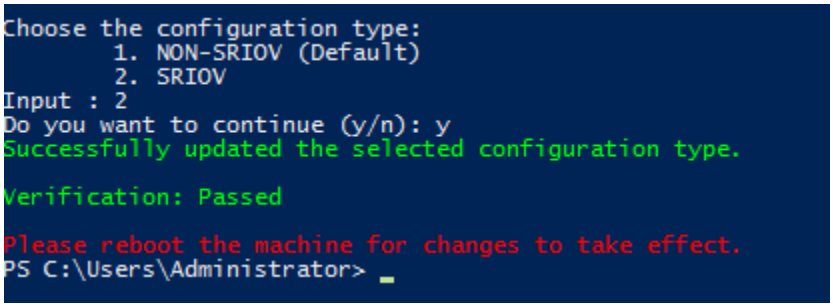


Figure 39 - Setting the configuration type

v. Reboot system for changes to take effect.

2.2. Host Configuration

i. Create a vSwitch with SR-IOV enabled using Chelsio adapter:

a) Open *Hyper-V Manager* application.

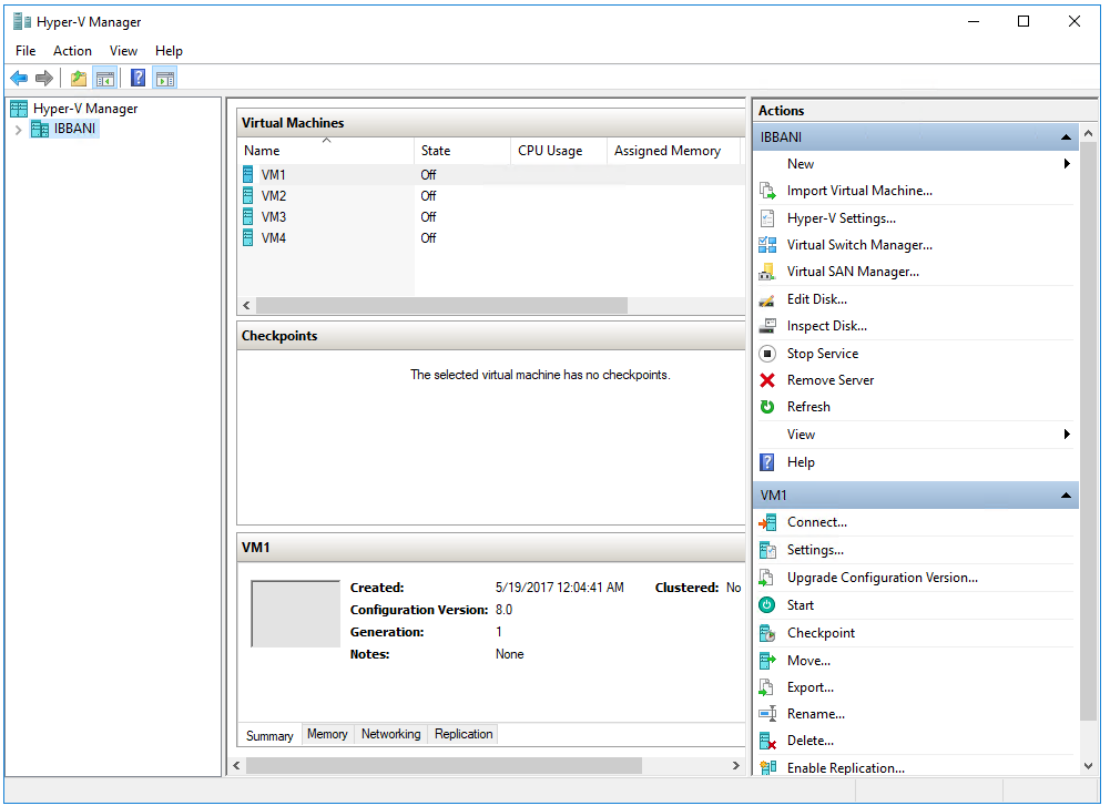


Figure 40 - Hyper-V Manager

b) Click the *Virtual Switch Manager* link under *Actions*.

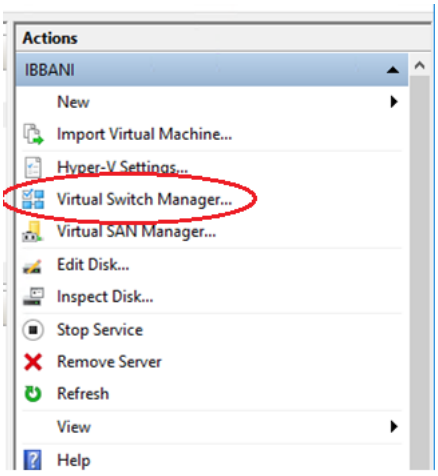


Figure 41 - Opening Virtual Switch Manager

c) Select *External* as the type of virtual switch to create. This will provide the VMs access to the external network. Click *Create Virtual Switch*.

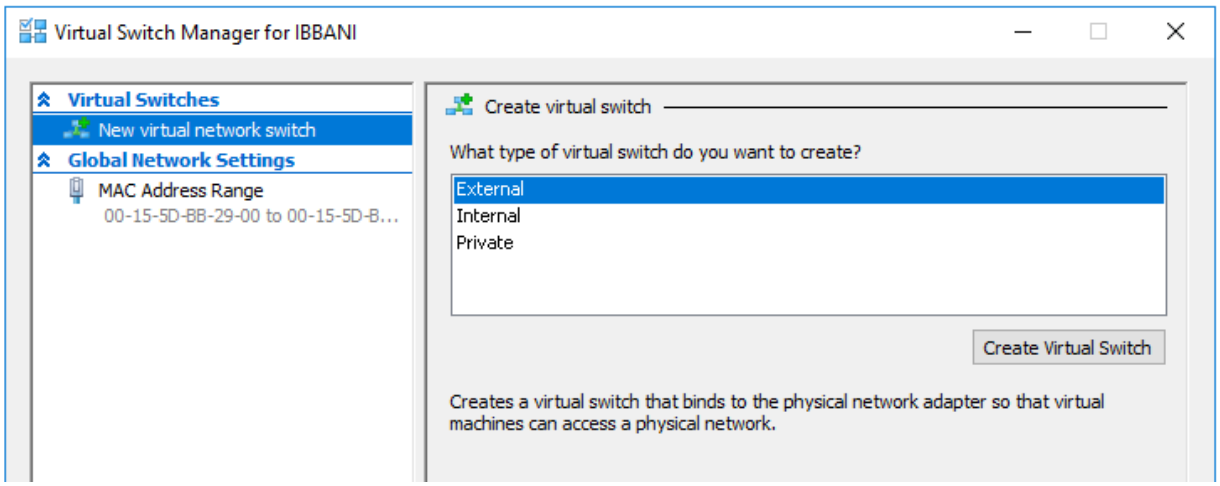


Figure 42 - Creating Virtual Switch

d) Provide a name for the vSwitch. Select the Chelsio adapter to connect the vSwitch to and select *Enable single-root I/O virtualization (SR-IOV)*. Click **OK**.

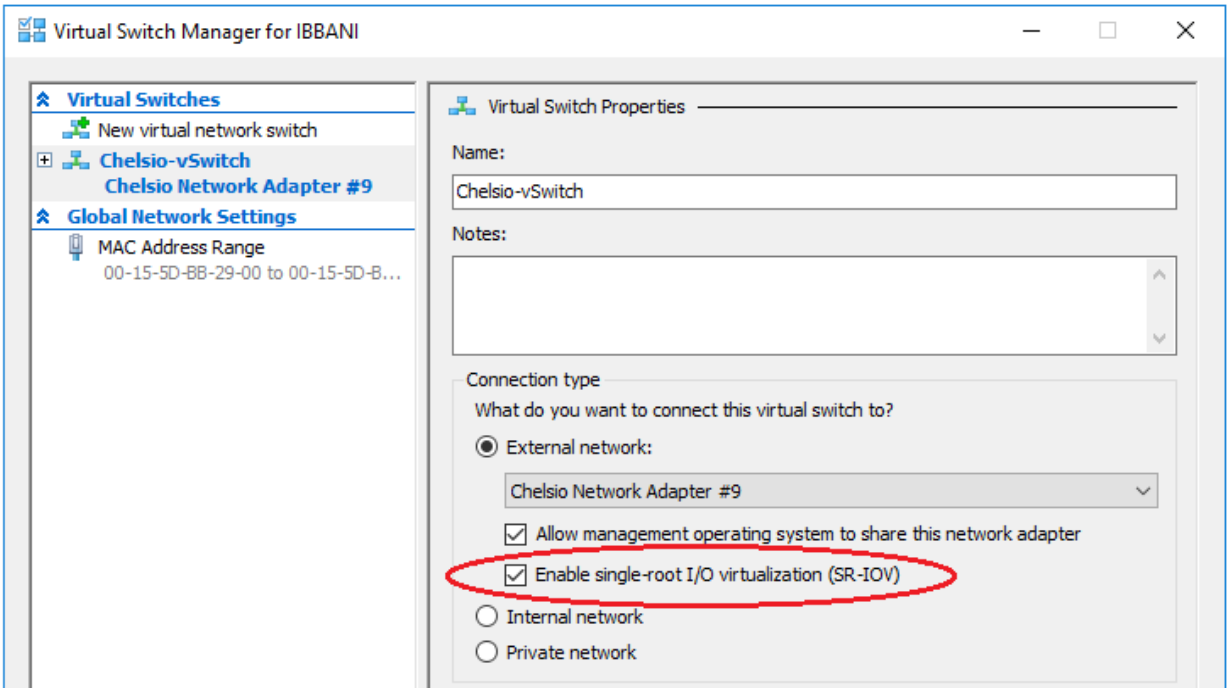


Figure 43 - Enabling SR-IOV on vSwitch

e) Click **Yes** on the prompt that appears next.

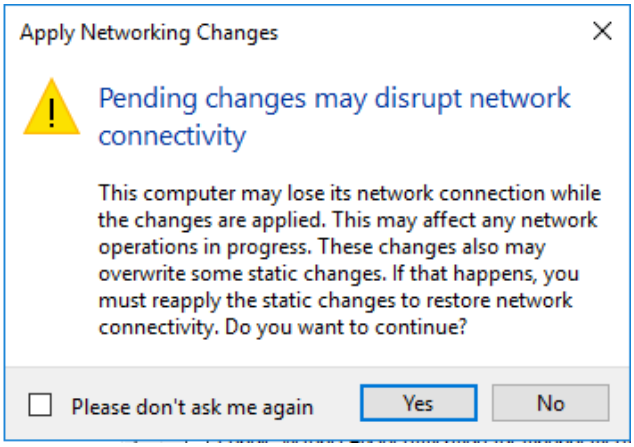


Figure 44 - Confirming Changes

ii. Assign a virtual network adapter with SR-IOV enabled to the VM.

a) Open Hyper-V Manager. Right-click on the VM and select *Settings*.

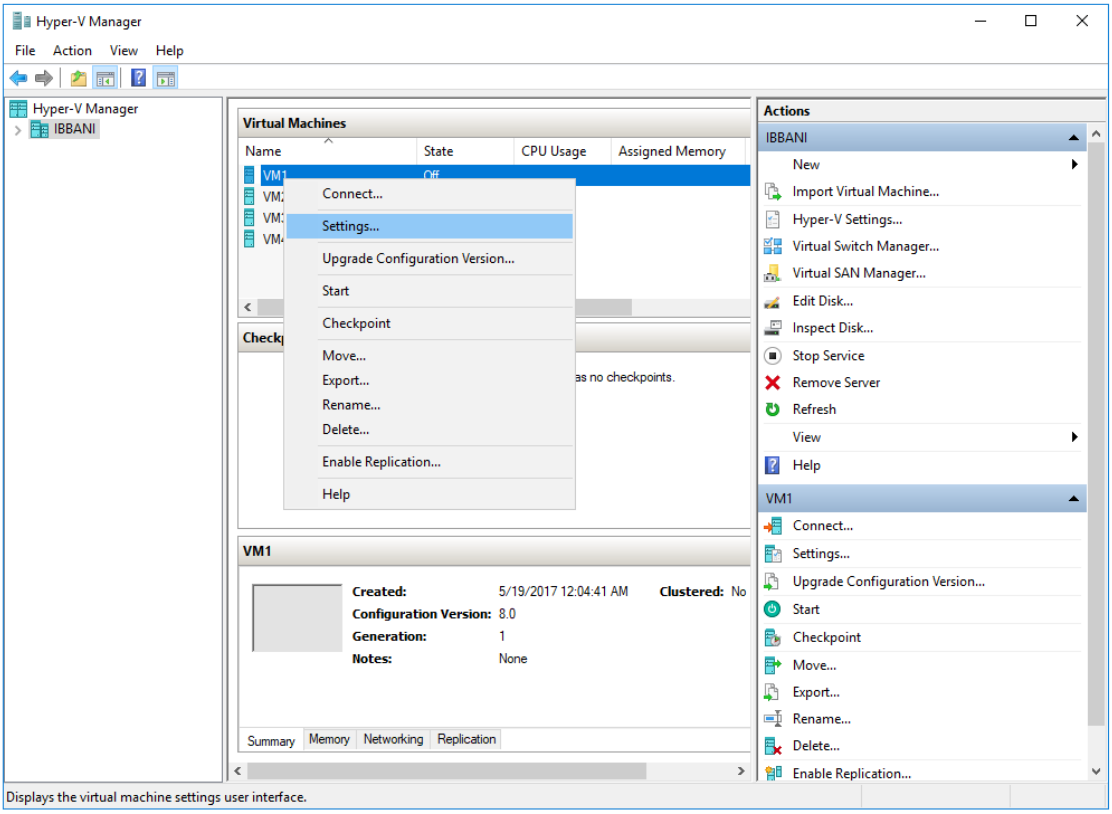


Figure 45 - Hyper-V Manager

b) Expand the *Network Adapter* link, click *Hardware Acceleration* and select *Enable SR-IOV*. Click **OK**.

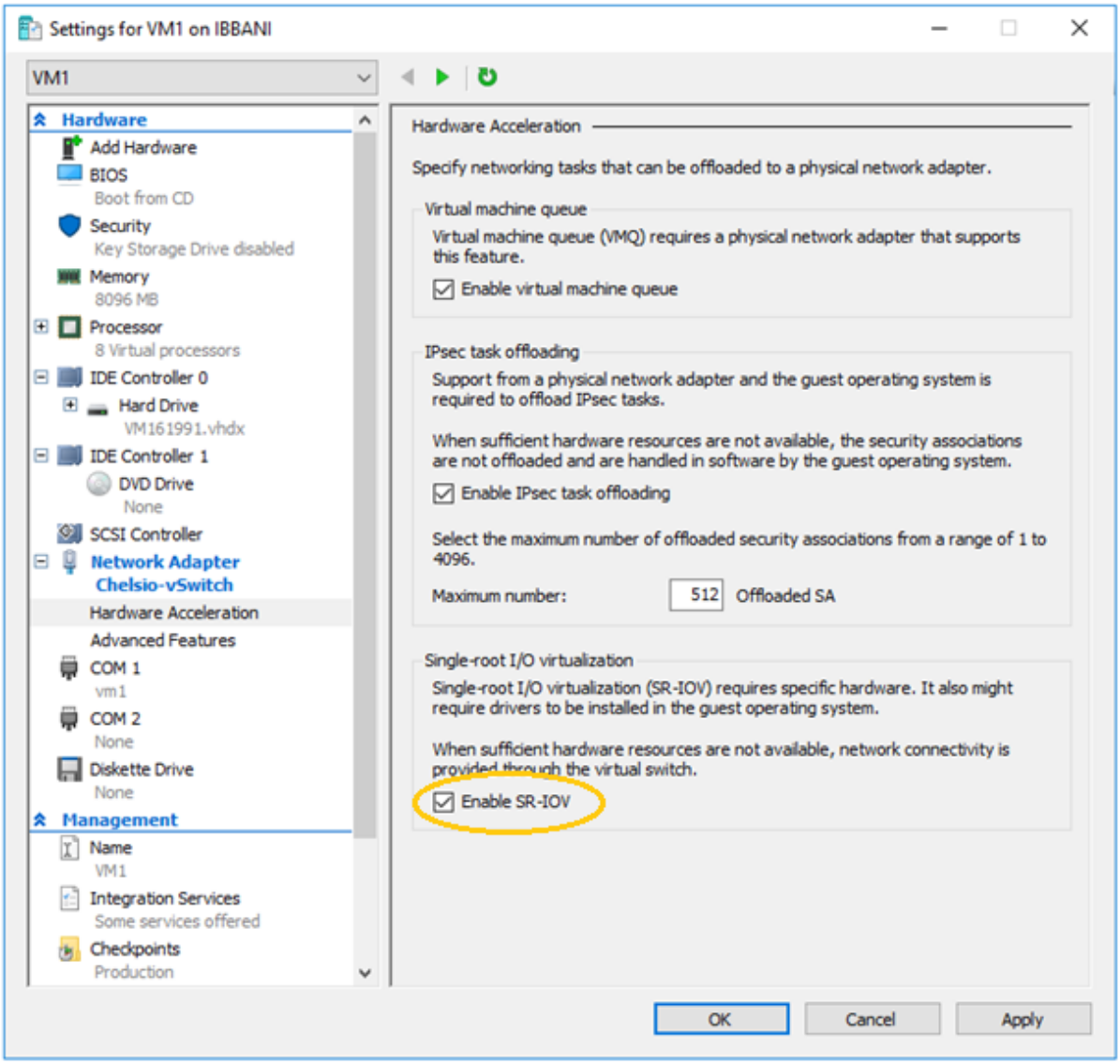


Figure 46 - Enabling SR-IOV on Virtual Network Adapter

c) Right-click on the VM and select *Start* to start the VM.

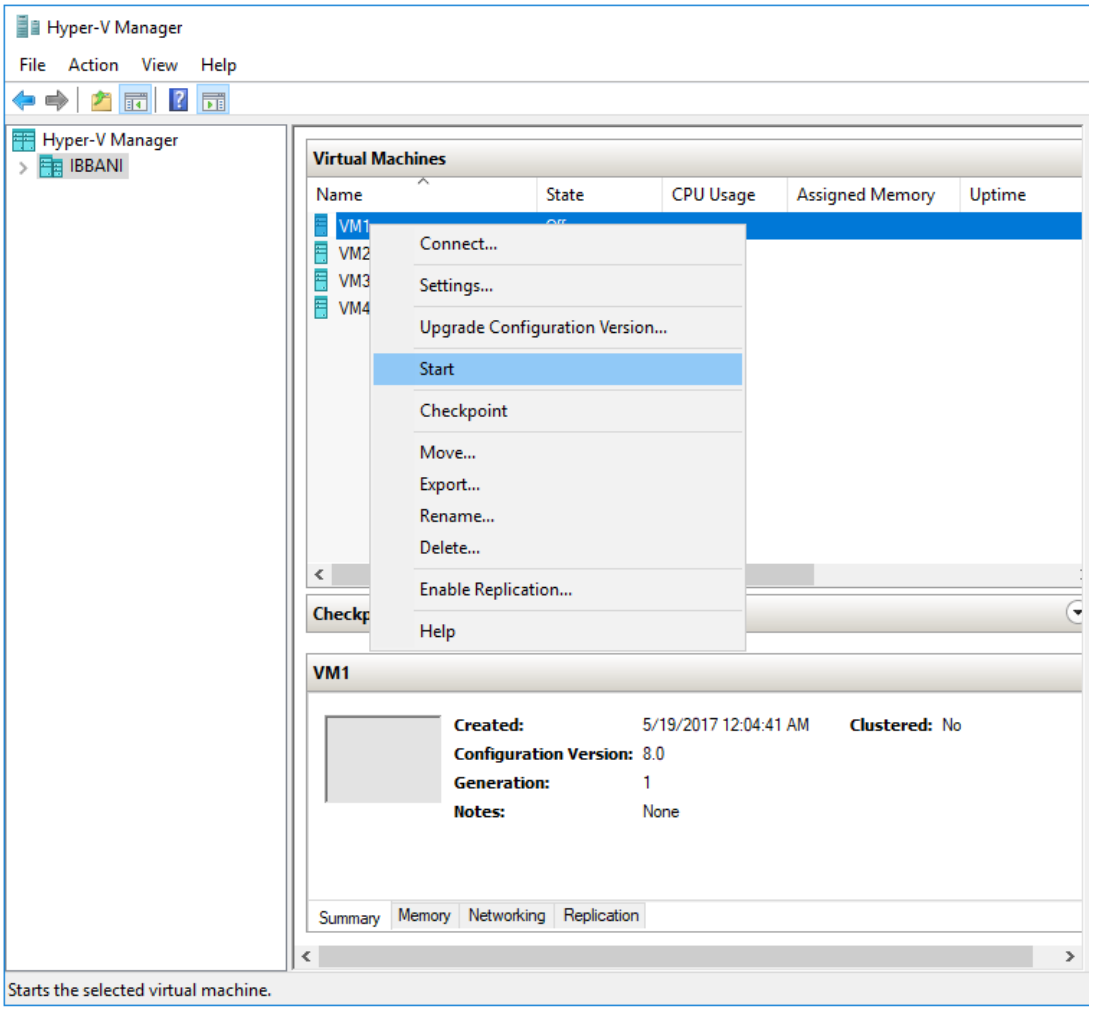


Figure 47 - Starting VM

2.3. Guest (VM) Configuration

i. Install Unified Wire

- If the guest has internet access, proceed with installation as described in the Software/Driver Installation section of the Chelsio Unified Wire chapter.
- If the guest doesn't have internet access, create an ISO image of the installer or zip package contents on the host. Open *Hyper-V Manager→VM→Settings→IDE Controller* and attach the image to the virtual CD/DVD drive of the guest. Proceed with the installation as described in the Software/Driver Installation section of the **Unified Wire** chapter.

ii. Verify if SR-IOV was enabled successfully on the host:

```
PS C:\Users\Administrator> Get-VMNetworkAdapter <vm_name> | fl
```

The *VFDataPathActive* parameter must display *True* as shown in the image below:

```
PS C:\Users\Administrator> Get-VMNetworkAdapter vm1 |fl

Name                               : Network Adapter
Id                                 : Microsoft:4C342213-5CC2-4FBD-93D7-CE036FD31CB3\C1018CF0-A3FF-4BEE-97B7-61BC46A7461C
IsLegacy                           : False
IsManagementOs                     : False
ComputerName                       : IBBANI
VMName                             : VM1
VMId                               : 4c342213-5cc2-4fbd-93d7-ce036fd31cb3
SwitchName                         : Chelsio-vSwitch
SwitchId                           : 09ab45b9-cd25-414f-8c18-01264f0b0bd2
Connected                          : True
PoolName                           :
MacAddress                         : 00155DBB291B
DynamicMacAddressEnabled           : True
AllowPacketDirect                  : False
MacAddressSpoofing                 : Off
AllowTeaming                       : Off
RouterGuard                        : Off
DhcpGuard                          : Off
StormLimit                         : 0
PortMirroringMode                  : None
IeeePriorityTag                    : Off
VirtualSubnetId                    : 0
DynamicIPAddressLimit              : 0
DeviceNaming                       : Off
VMQWeight                          : 100
VMQUsage                           : 0
IOVWeight                          : 100
IOVUsage                           : 1
IovQueuePairsRequested             : 1
IovQueuePairsAssigned              : 1
IOVInterruptModeration             : Default
PacketDirectNumProcs               : 0
PacketDirectModerationCount        : 64
PacketDirectModerationInterval     : 1000000
VrssEnabledRequested               : True
VrssEnabled                        : False
VmmqEnabledRequested               : False
VmmqEnabled                        : False
VmmqQueuePairsRequested            : 16
VmmqQueuePairs                     : 0
IPsecOffloadMaxSA                  : 512
IPsecOffloadSAUsage                : 0
VFDataPathActive                   : True
MaximumBandwidth                   :
MinimumBandwidthAbsolute           :
MinimumBandwidthWeight             :
BandwidthPercentage                : 0%
MandatoryFeatureId                 : {}
MandatoryFeatureName               : {}
Status                             : {Ok}
IPAddresses                        : {101.1.1.39, fe80::f0fa:848f:51f2:d5ec}
```

Figure 48 - Verifying if SR-IOV was enabled successfully

To uninstall Unified Wire, please refer Software/Driver Uninstallation section of the Chelsio Unified Wire chapter for step-by-step instructions.

2.4. Guest RDMA (Mode 3)

In this mode, you can run RDMA traffic on Guest (VM) using a virtual switch. This feature is enabled by default in the driver, but must be enabled on the *Microsoft Hyper-V Virtual Ethernet* adapter. Follow the steps mentioned below to enable the feature:

i. If you haven't done already, run the Chelsio Unified Wire Installer which will install NDIS function driver.
ii. Configure Chelsio adapter as described in the Enabling SR-IOV section.
iii. Enable SR-IOV support on the host as described in the Host Configuration section.

iv. Enable Guest RDMA on VM using the following command on the host:

```
PS C:\Users\Administrator> Set-VMNetworkAdapterRdma -VMNetworkAdapterName
<adapter_name> -VMName <vm_name> -RdmaWeight 100
```

Note Make sure that VM is turned off before running the above command.

Example:

```
PS C:\Users\Administrator> Set-VMNetworkAdapterRdma -VMNetworkAdapterName "Network Adapter" -VMName VM1 -RdmaWeight 100
PS C:\Users\Administrator>
```

Figure 49 - Enabling Guest RDMA on VM

To enable Guest RDMA on all VM network adapters, run the following command on the host:

```
PS C:\Users\Administrator> Set-VMNetworkAdapterRdma -VMNetworkAdapterName *
-VMName <VMName> -RdmaWeight 100
```

v. Verify if Guest RDMA was enabled successfully using the *Get-VMNetworkAdapterRdma* command. The *Rdmaweight* parameter should show 100.

```
PS C:\Users\Administrator> Get-VMNetworkAdapterRdma vm1

RdmaWeight     : 100
ParentAdapter : VMNetworkAdapter (Name = 'Network Adapter', VMName = 'VM1') [VMId = '4c342213-5cc2-4fbd-93d7-ce036fd31cb3']
IsTemplate     : False
CimSession     : CimSession: .
ComputerName   : IBBANI
IsDeleted      : False
```

Figure 50 - Verifying if Guest RDMA was enabled successfully

vi. In *Hyper-V Manager*, right-click on the VM and select *Start* to start the VM.

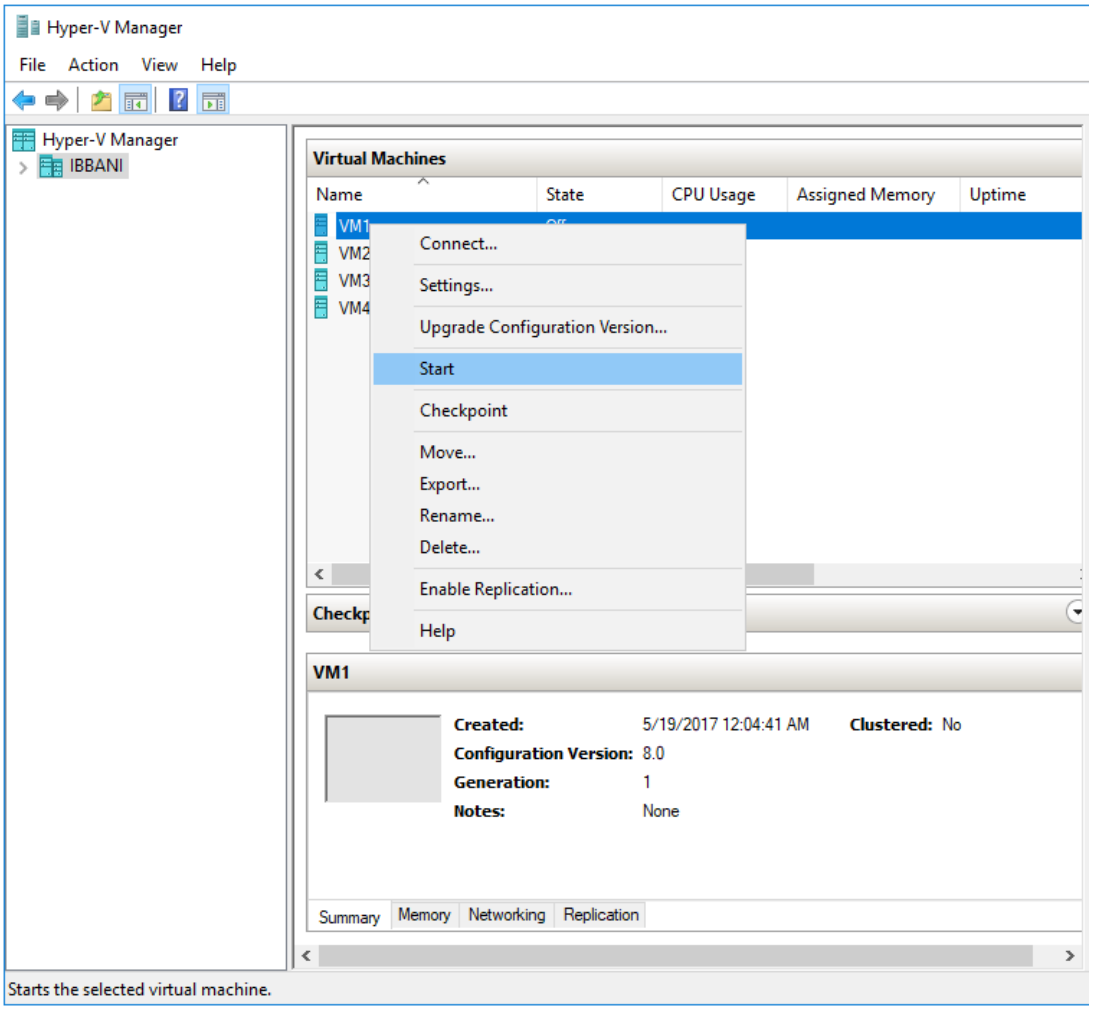


Figure 51 - Starting VM

vii. If you haven't done already, run the Chelsio Unified Wire Installer on VM which will install NDIS function driver.

viii. Open **Device Manager** on VM, click on **Network Adapters**, right click on **Microsoft Hyper-V Network Adapter** and select **Properties.**

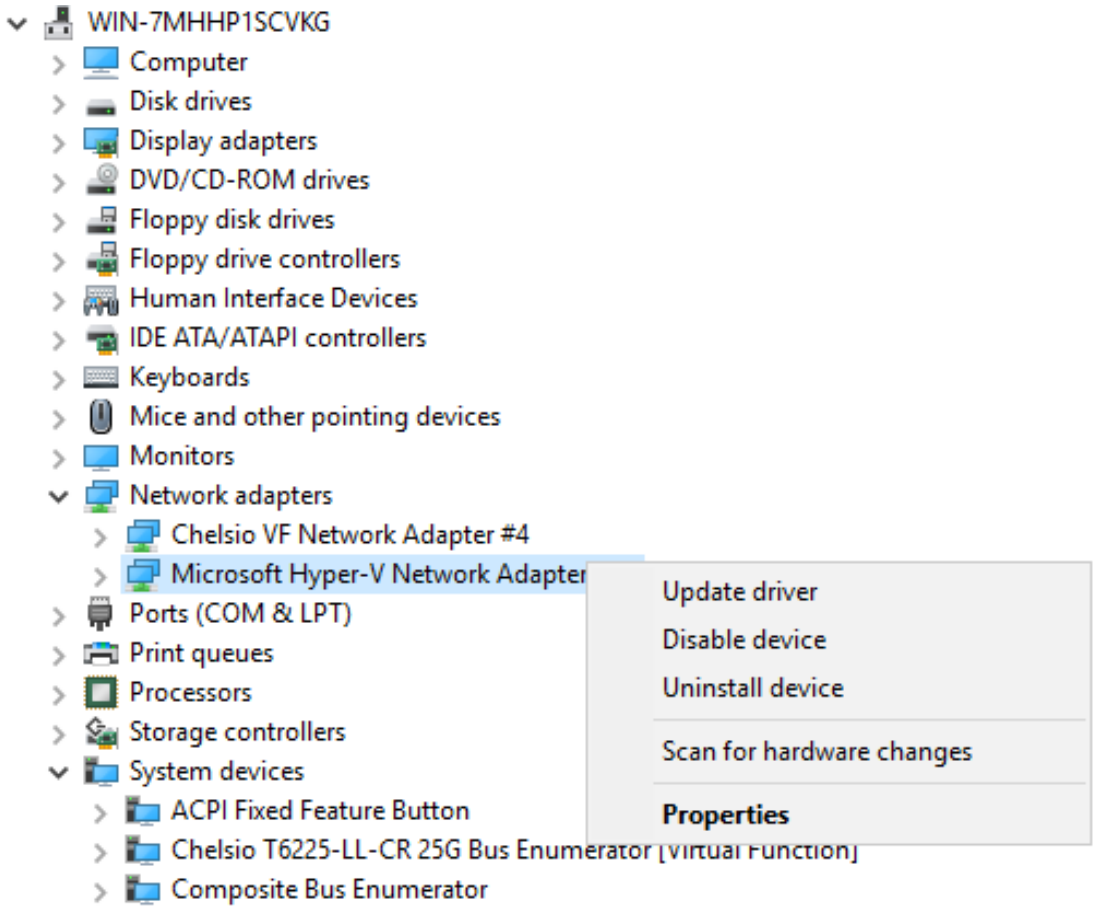


Figure 52 - Device Manager on VM

ix. In the **Advanced** tab, select the *Network Direct (RDMA)* property from the list and enable it. Click **OK**.

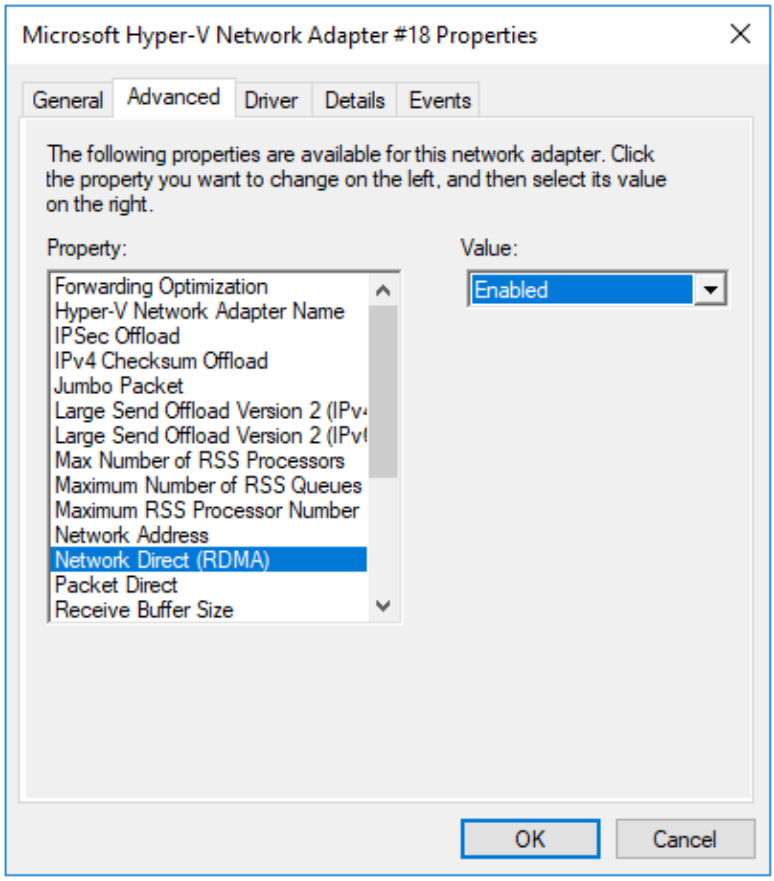


Figure 53 - Enabling Guest RDMA on Microsoft Hyper-V Network Adapter

x. To verify if RDMA was enabled successfully, open PowerShell with administrative privileges and run *Get-NetAdapterRdma*, *Get-SmbClientNetworkInterface* and *Get-SmbServerNetworkInterface* commands. You should see a similar output:

```
PS C:\Users\Administrator> Get-NetAdapterRdma

Name                      InterfaceDescription                    Enabled
----                      --------------------                    -------
Ethernet 21               Microsoft Hyper-V Network Adapter #18   True
Ethernet 22               Chelsio VF Network Adapter #4           True


PS C:\Users\Administrator> Get-SmbClientNetworkInterface

Interface Index RSS Capable RDMA Capable Speed   IpAddresses                                   Friendly Name
--------------- ----------- ------------ -----   -----------                                   -------------
12              True        True         25 Gbps {fe80::f0fa:848f:51f2:d5ec, 101.1.1.39}       Ethernet 21


PS C:\Users\Administrator> Get-SmbServerNetworkInterface

Scope Name Interface Index RSS Capable RDMA Capable Speed   IpAddress
---------- --------------- ----------- ------------ -----   ---------
*          12              True        True         25 Gbps fe80::f0fa:848f:51f2:d5ec
*          12              True        True         25 Gbps 101.1.1.39
```

Figure 54 - Verifying if Guest RDMA was enabled successfully

Note If the output of any one of the three commands is not *true*, then disable and enable the physical Chelsio network adapter on the host and try again.

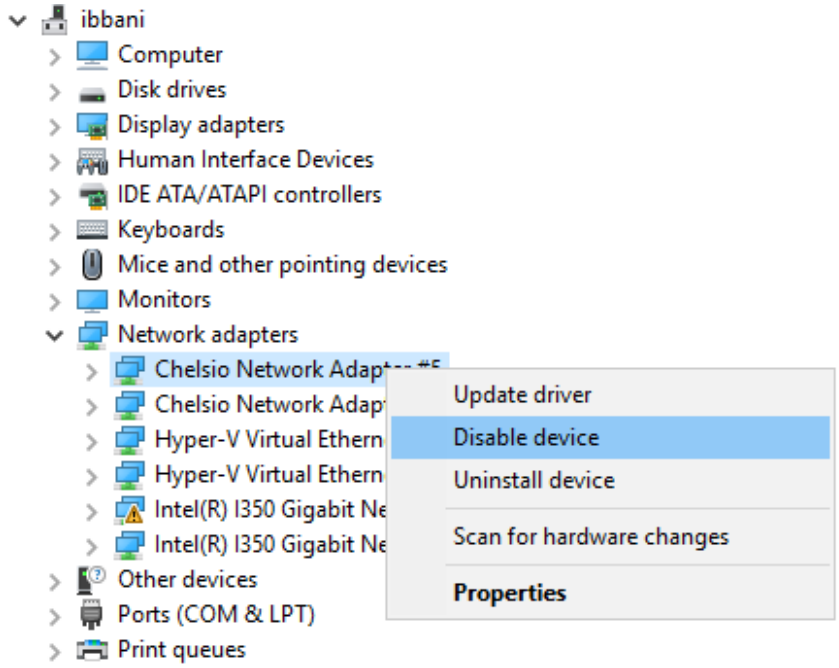


Figure 55 - Troubleshooting

You should be able to run RDMA traffic on the VM successfully now. Use a network traffic monitoring tool (like Windows Performance Monitor) to verify.

VII. iSCSI Storport Miniport

1. Introduction

The Chelsio Terminator series of Adapters support iSCSI acceleration and iSCSI Direct Data Placement (DDP) where the hardware handles the expensive byte touching operations, such as CRC computation and verification, and direct DMA to the final host memory destination:

- **iSCSI PDU digest generation and verification**
  On transmitting, Chelsio h/w computes and inserts the Header and Data digest into the PDUs.
  On receiving, Chelsio h/w computes and verifies the Header and Data digest of the PDUs.

- **Direct Data Placement (DDP)**
  Chelsio h/w can directly place the iSCSI Data-In or Data-Out PDU's payload into pre-posted final destination host-memory buffers based on the Initiator Task Tag (ITT) in Data-In or Target Task Tag (TTT) in Data-Out PDUs.

- **PDU Transmit and Recovery**
  On transmitting, Chelsio h/w accepts the complete PDU (header + data) from the host driver, computes and inserts the digests, decomposes the PDU into multiple TCP segments if necessary, and transmit all the TCP segments onto the wire. It handles TCP retransmission if needed.
  On receiving, Chelsio h/w recovers the iSCSI PDU by reassembling TCP segments, separating the header and data, calculating and verifying the digests, then forwarding the header to the host. The payload data, if possible, will be directly placed into the pre-posted host DDP buffer. Otherwise, the payload data will be sent to the host too.

1.1. Hardware Requirements

1.1.1. Supported Adapters

The following are the currently shipping Chelsio Adapters that are compatible with the iSCSI Storport Miniport:

- T62100-CR
- T62100-LP-CR
- T6225-CR
- T6225-LL-CR
- T520-BT
- T520-CR
- T580-CR
- T580-LP-CR
- T520-LL-CR
- T540-CR

1.2. Software Requirements

1.2.1. Windows Requirement

Currently iSCSI Storport Miniport driver is available for Windows versions.

- Server 2016
- 10 AU Client
- Server 2012 R2

Other versions have not been tested and are not guaranteed to work.

2. Software/Driver Configuration and Fine-tuning

2.1. Configuring iSCSI Initiator

2.1.1. Configuring IP

There are two ways to assign IP to Chelsio iSCSI interface:

- **DHCP**: NIC driver must be installed (see Software/Driver Installation). Upon installing the driver, DHCP will be enabled by default and the NIC interface will be assigned an IP automatically. The same IP can be used by the iSCSI interface as well.
- **Static**: To assign a static IP to the iSCSI interface use the Device Manager, as explained below:

1. Open **Device Manager**, right click on **Chelsio iSCSI Initiator** under **Storage controllers** and click on **Properties**.

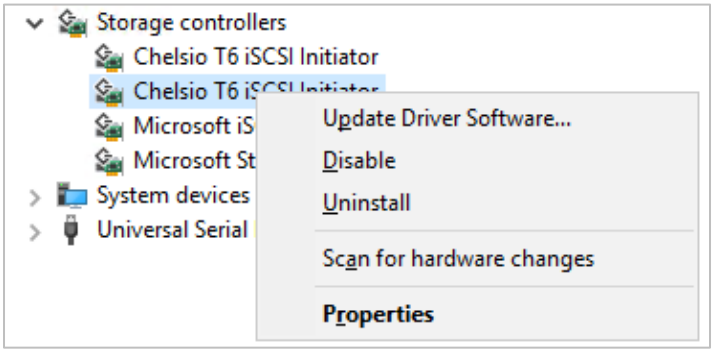


Figure 56 - Device Manager

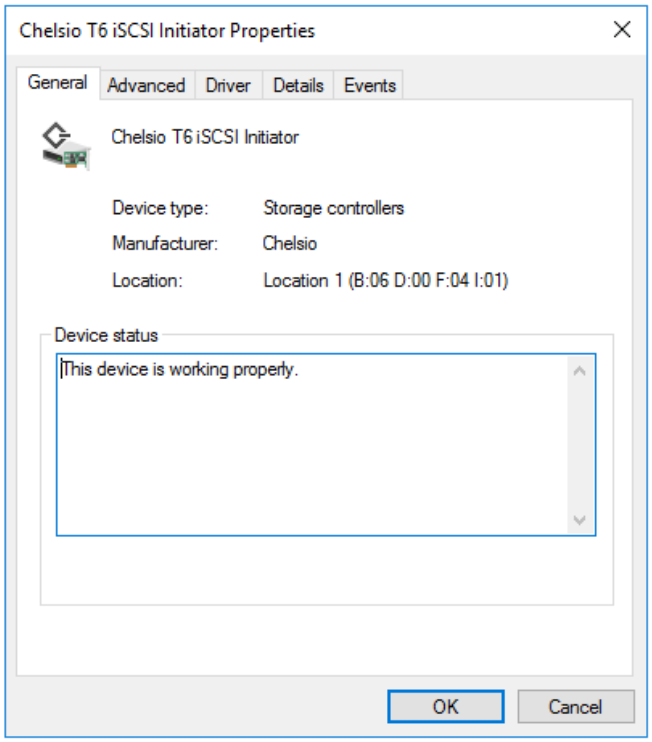


Figure 57 - Chelsio adapter physical port properties

2. Click on **Advanced** tab and select IP option and add IP. Similarly add subnet mask and default gateway.

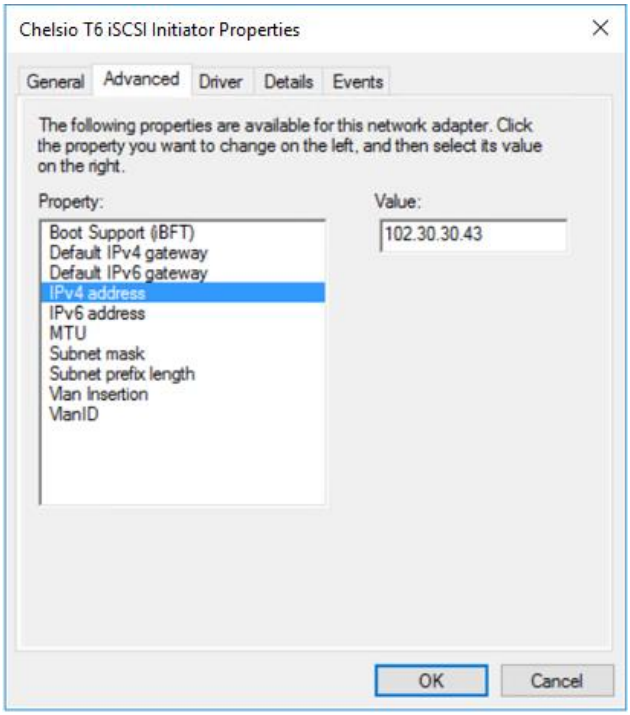


Figure 58 - Assigning IPv4 address

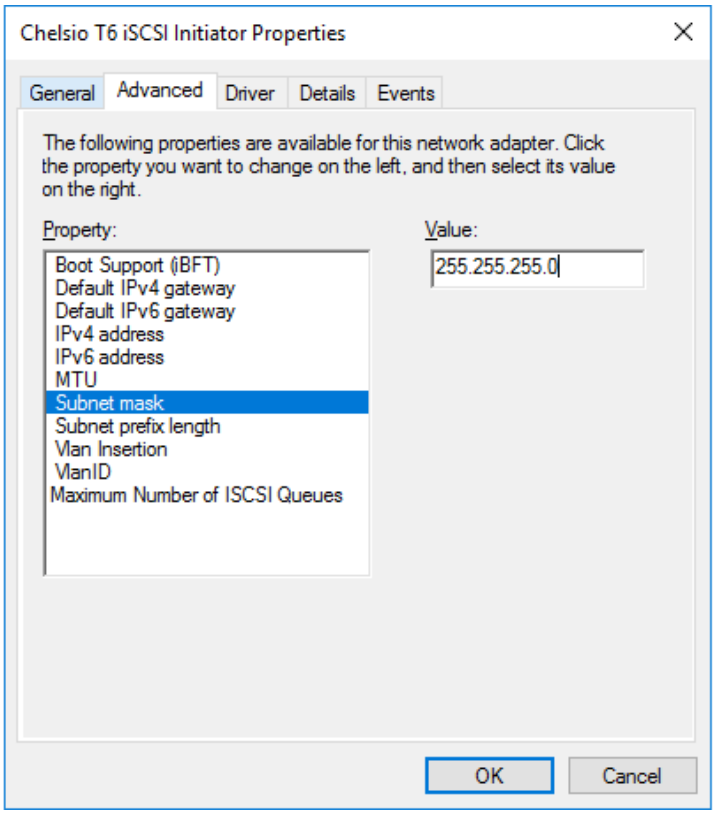


Figure 59 - Adding Subnet mask

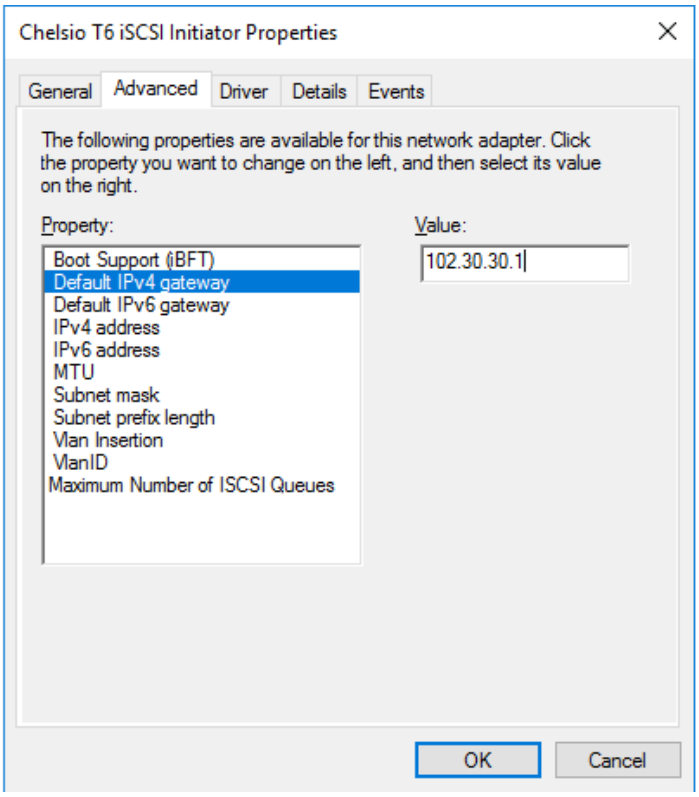


Figure 60 - Adding Default IPv4 gateway

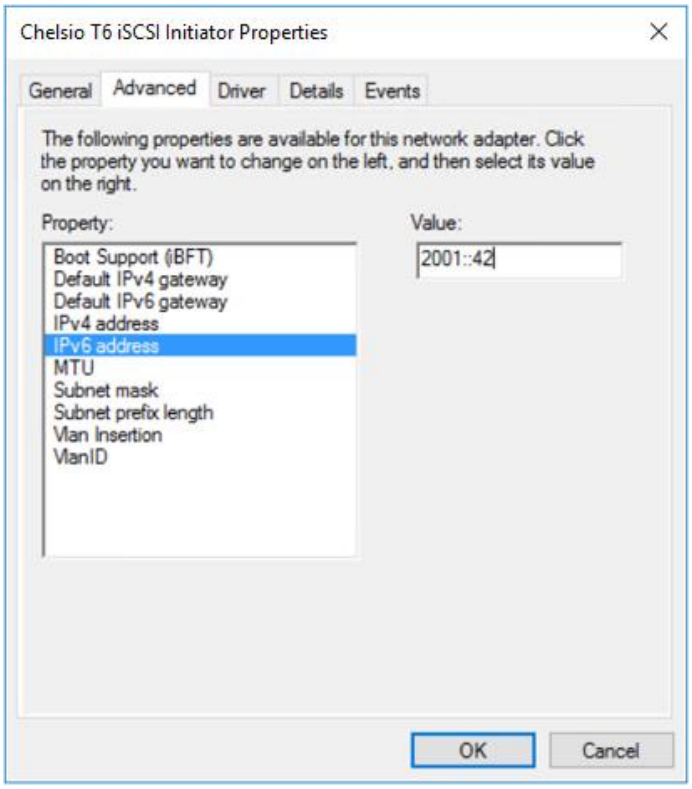


Figure 61 - Assigning IPv6 address

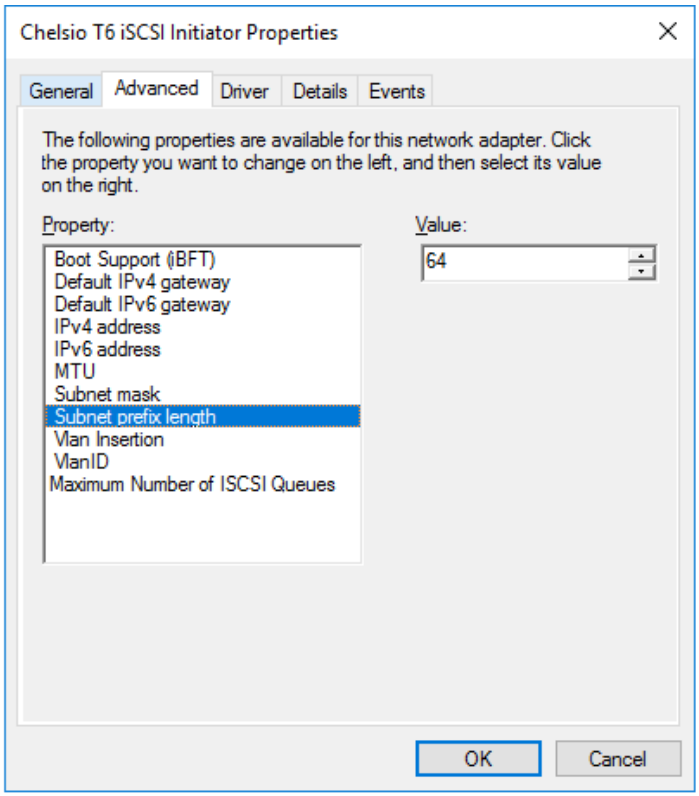


Figure 62 - Adding Subnet prefix length

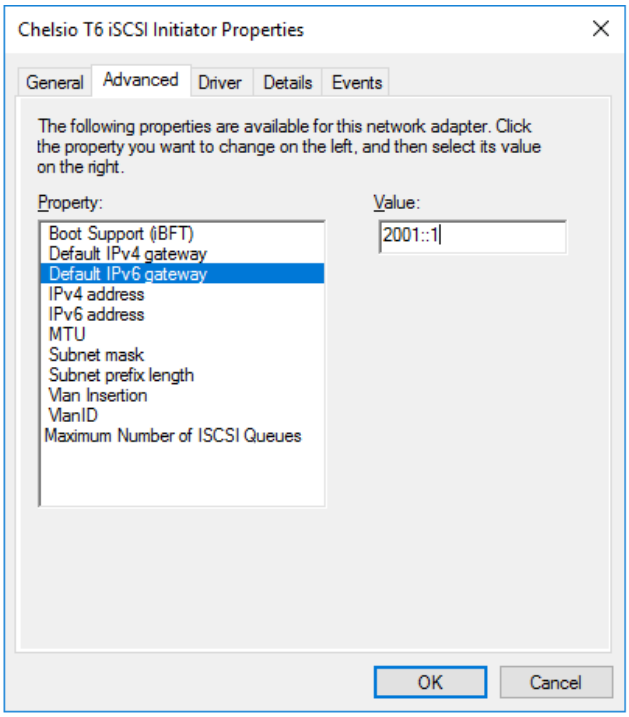


Figure 63 - Adding Default IPv6 gateway

2.1.2. Configuring Optional Parameters

Configure optional parameters like MTU, VLAN, etc., based on your requirement.

- Select MTU and assign value as per requirement.

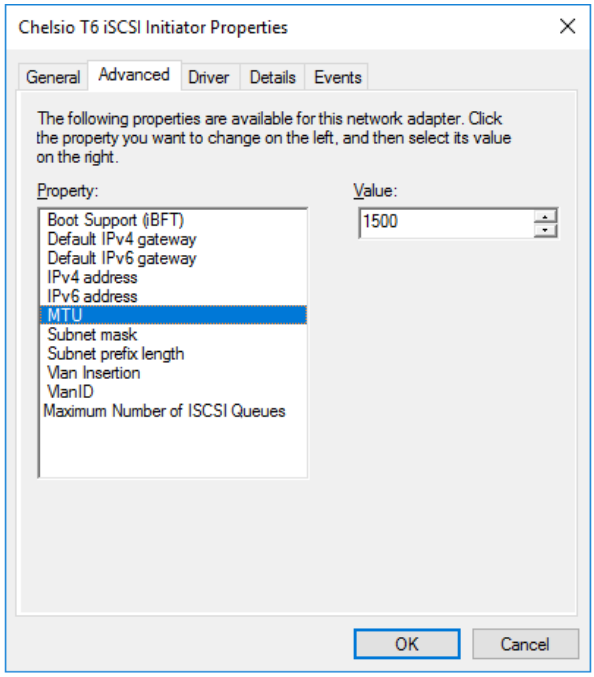


Figure 64 - Setting MTU

- VLAN is disabled by default, so to enable, click on “Vlan Insertion” and select value as “Enabled”.

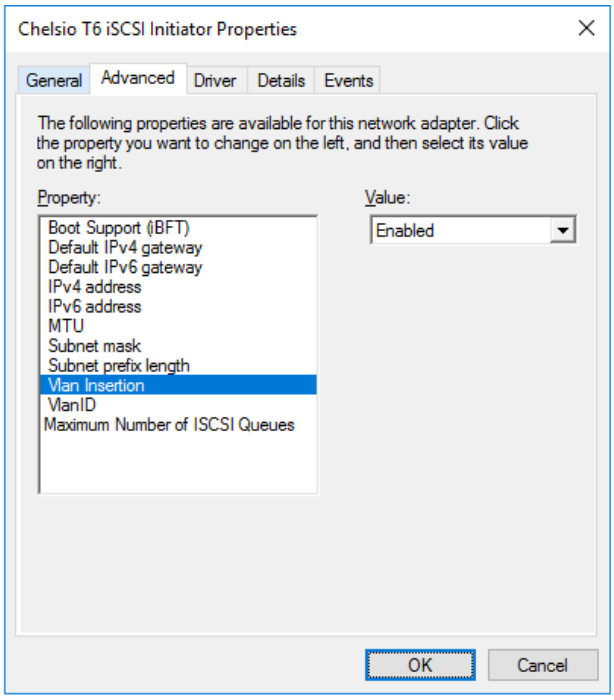


Figure 65 - Enabling VLAN

- Assign the required VLAN ID and click **OK.**

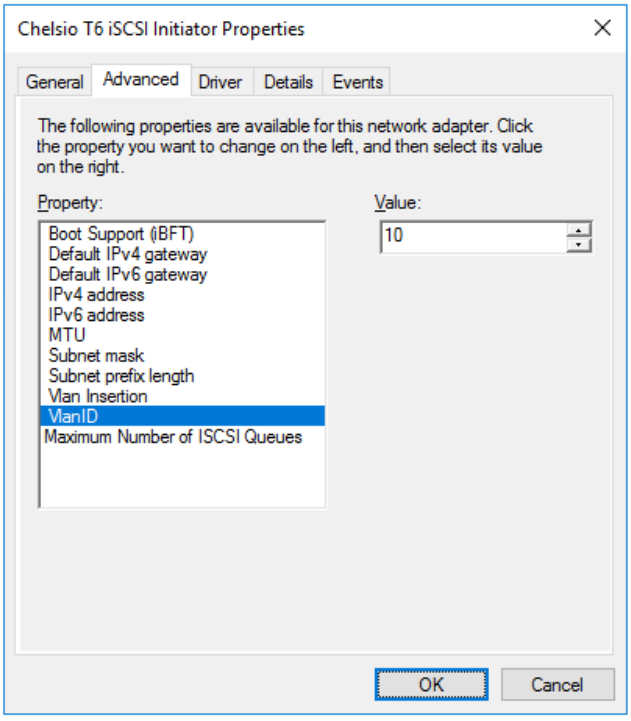


Figure 66 - Assigning VLAN Id

Repeat the above-mentioned steps to assign these driver properties on other ports.

2.2. iSCSI Target Discovery and Login

Before target discovery, make sure an IP address, either static or using DHCP, is assigned to the respective Chelsio iSCSI node (see previous section).

Note Connecting to more than 64 targets using single port is currently not supported.

Following are the set of instructions to discover the target and login to it using the Chelsio iSCSI interface:

1. To start the Initiator configuration, go to Control Panel and click on the **iSCSI Initiator** option**.**

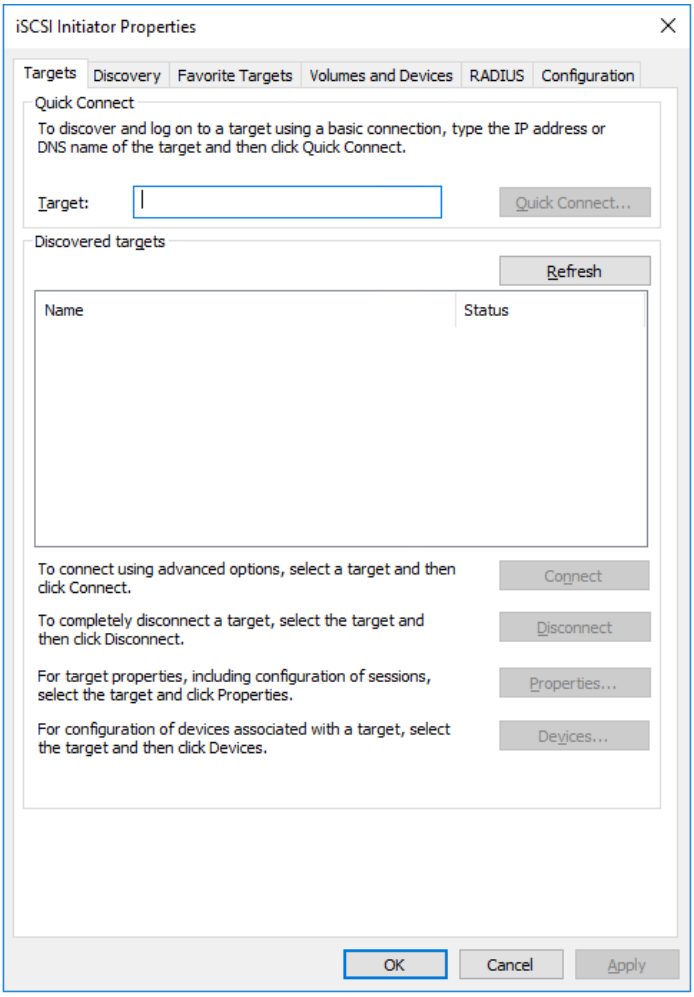


Figure 67 - iSCSI Initiator Properties: Targets tab

2. Choose the **Discovery** tab and click on **Discover Portal**.

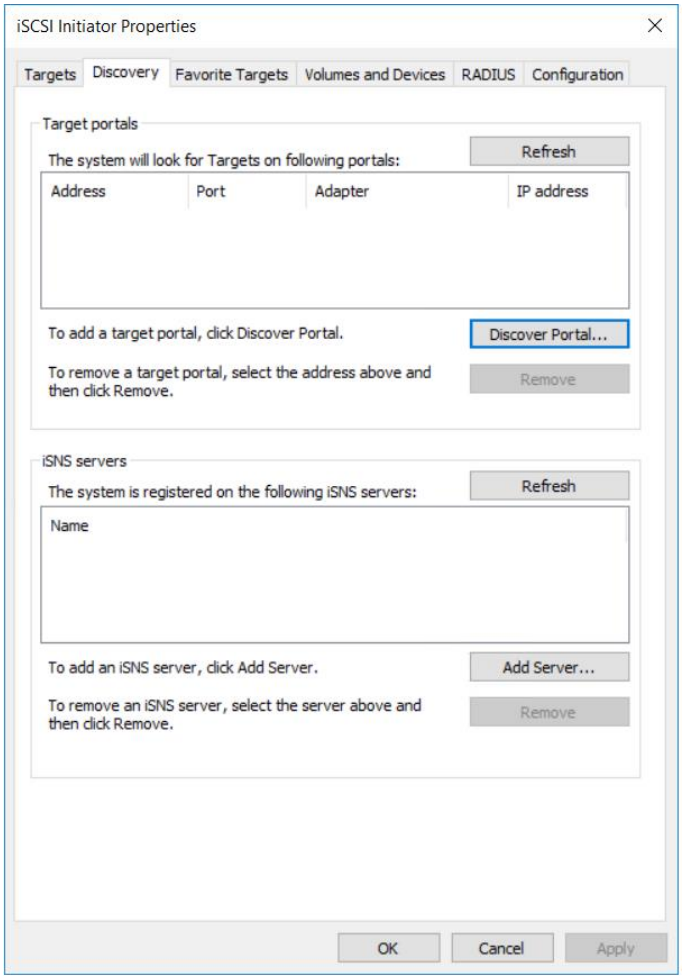


Figure 68 - iSCSI Initiator Properties: Discovery tab

3. Click on **Advanced** button.

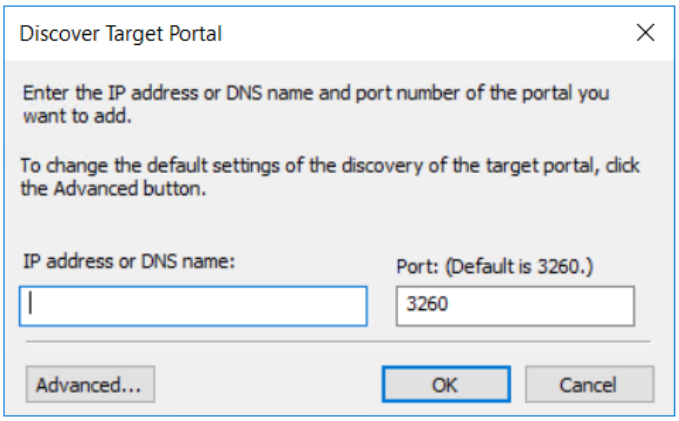


Figure 69 - Discovery Target Portal window

4. In the **Advanced Settings** window, select **Chelsio iSCSI interface** as the local adapter.

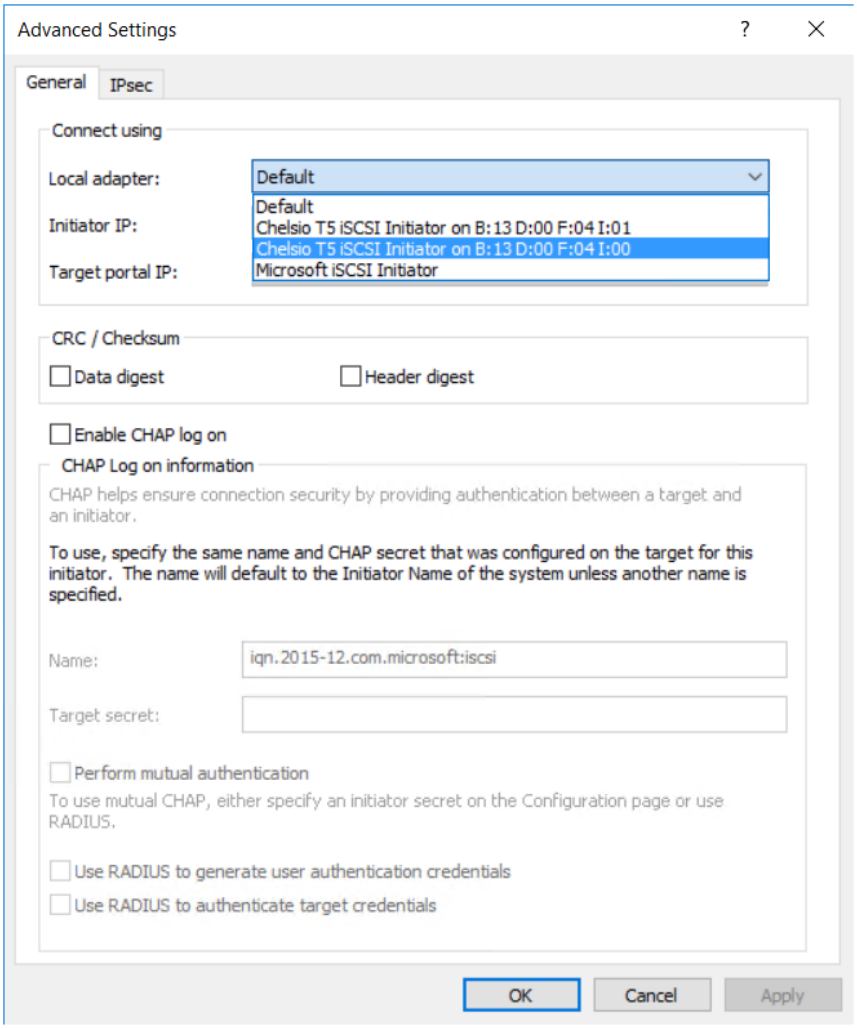


Figure 70 - Discovery Target Portal Advanced Settings: Selecting Local adapter

5. Select the corresponding Chelsio iSCSI Interface IP as the Initiator IP and click **OK**.

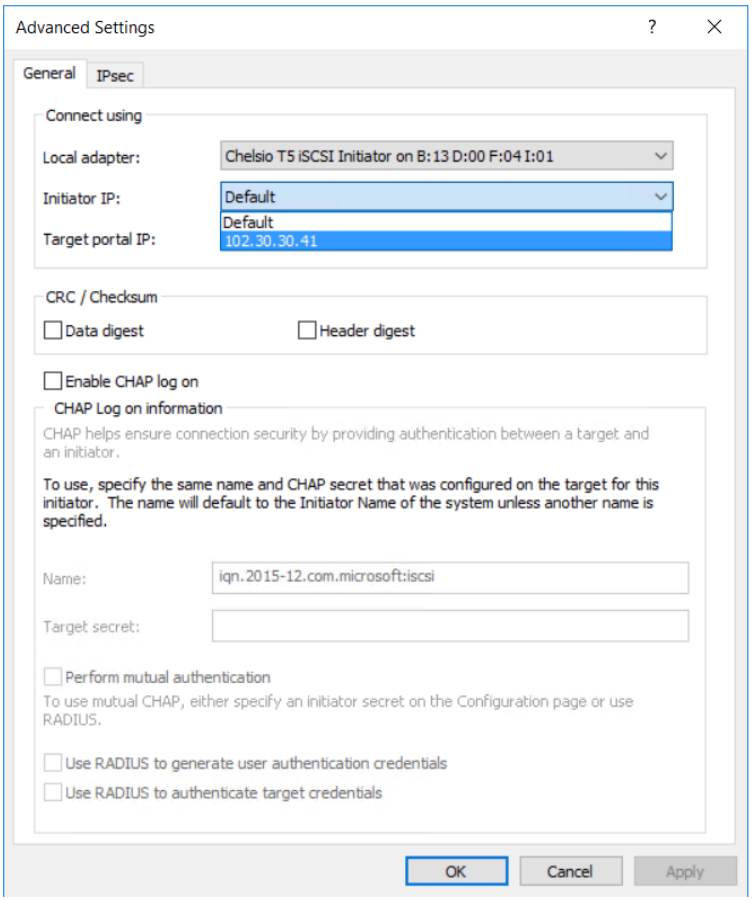


Figure 71 - Discovery Target Portal Advanced Settings: Specifying Initiator IP

6. Enter the IP address (DNS name not supported) of the target machine and the corresponding port number and click **OK** on the **Discover Target Portal** Window. Note that the default port number for iSCSI traffic is 3260.

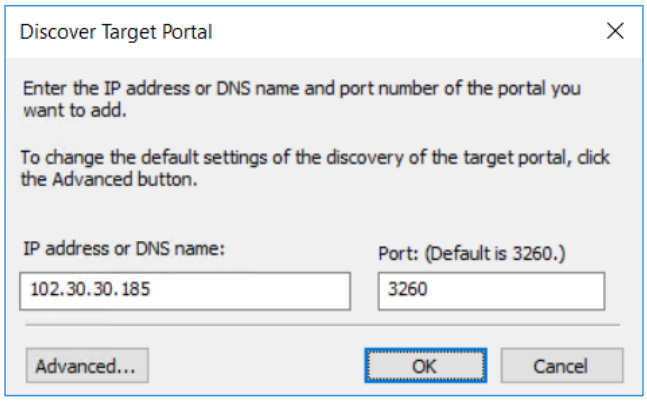


Figure 72 - Adding Target portal

7. Once target portal is added, details like target and initiator machine IP, Port number and Chelsio iSCSI interface IP will be displayed.

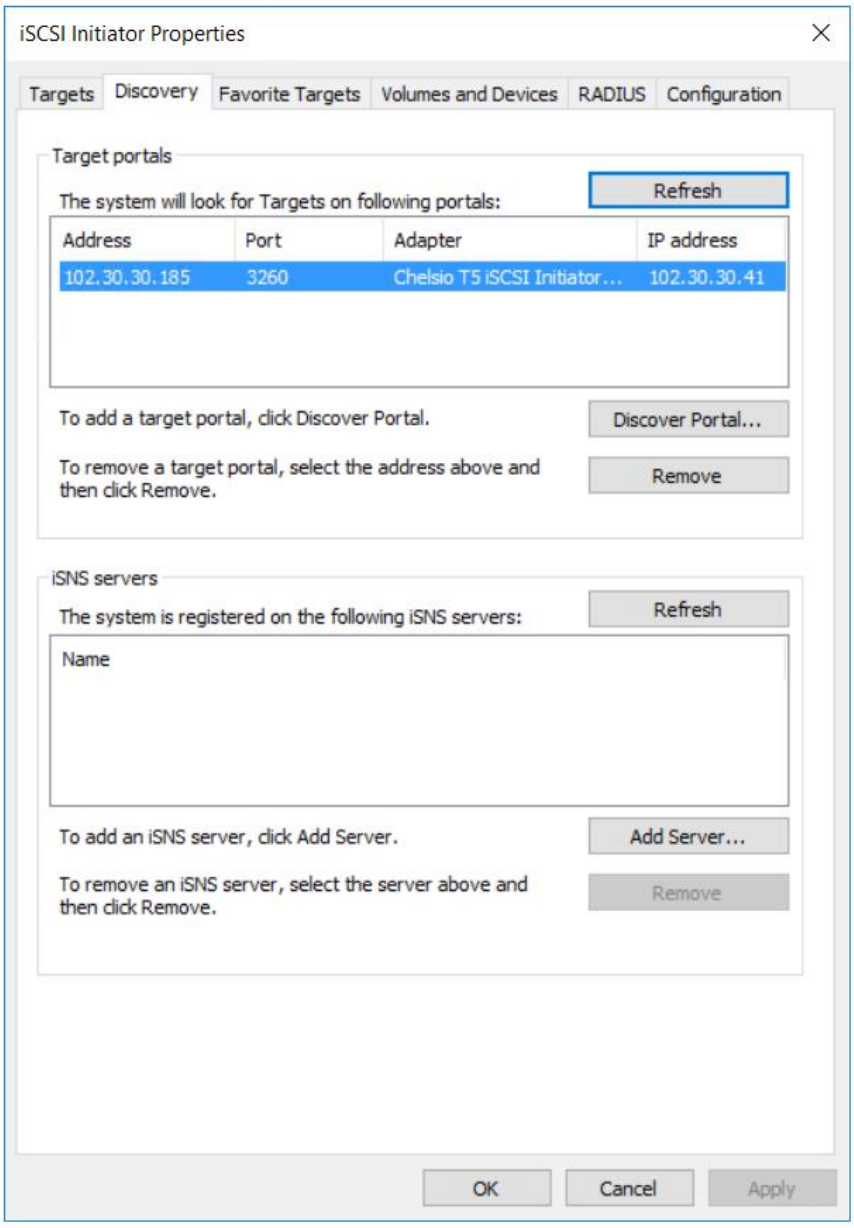


Figure 73 - Target portal added

8. Click on the **Targets** tab to see the list of targets available, choose a particular target and click on **Connect**.

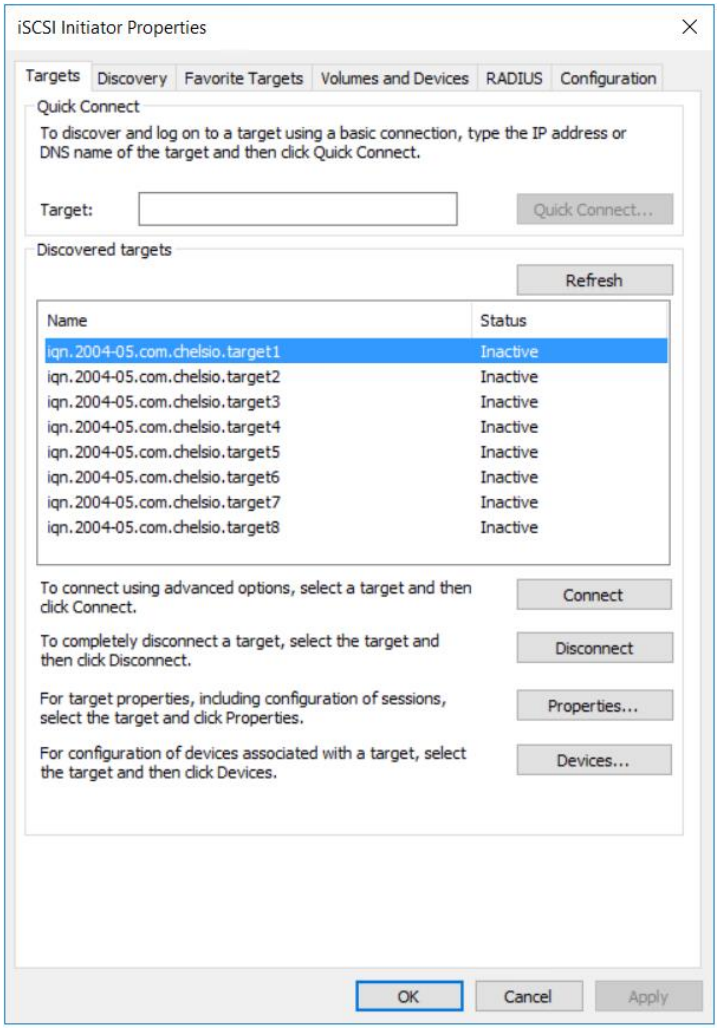


Figure 74 - Targets tab displaying list of available targets

9. A window pops up showing the Target Name, Click **OK**.

Figure 75 - Connecting to Target

10. After logging in to the target, the state of the target will change from *Inactive* to *Connected*.

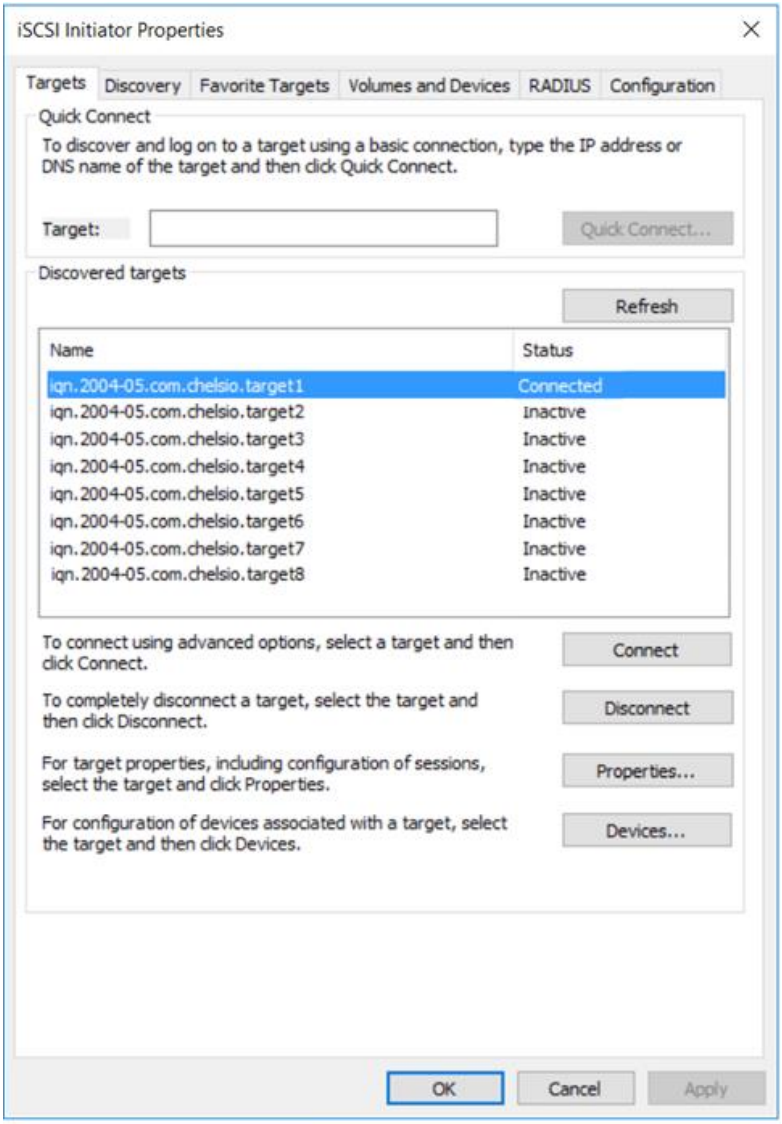


Figure 76 - iSCSI target connected

11. After successful login, go to **Disk Management** and make the respective volume online.

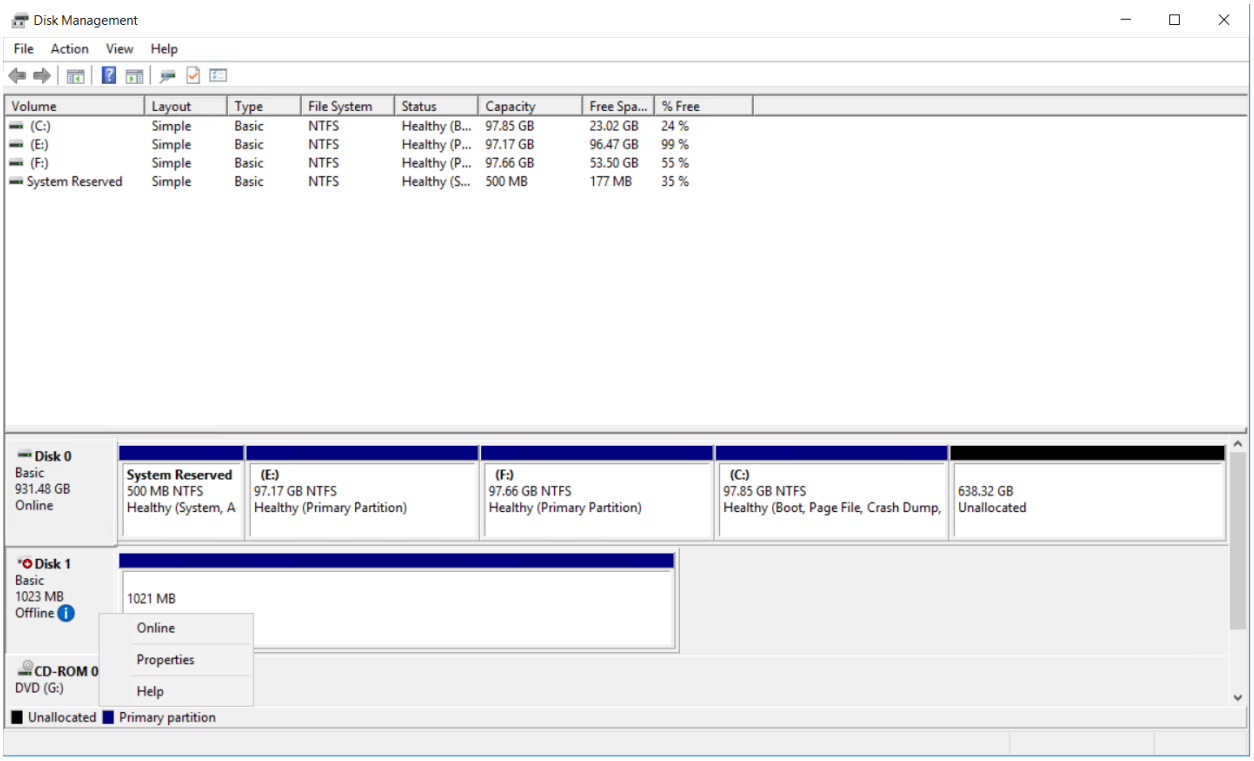


Figure 77 - Disk Management: making volume online

VIII. iSER Initiator

1. Introduction

The iSCSI Extensions for RDMA (iSER) protocol is a translation layer for operating iSCSI over RDMA transports, such as iWARP/Ethernet or InfiniBand.

Note *This is an alpha release.*

1.1. Hardware Requirements

1.1.1. Supported Adapters

The following are the currently shipping Chelsio adapters that are compatible with the iSER Initiator driver:

- T62100-CR
- T62100-LP-CR
- T6225-CR
- T6225-LL-CR
- T520-BT
- T520-CR
- T580-CR
- T580-LP-CR
- T520-LL-CR
- T540-CR

1.2. Software Requirements

1.2.1. Windows Requirement

Currently iSER Initiator driver is available for the following Windows versions.

- Server 2016
- 10 AU Client
- Server 2012 R2

Other versions have not been tested and are not guaranteed to work.

2. Software/Driver Configuration and Fine-tuning

2.1. Configuring iSER Target

The iSER target must be configured before establishing connection. For more information, please refer Unified Wire for Linux documentation available at the Chelsio Download Center.

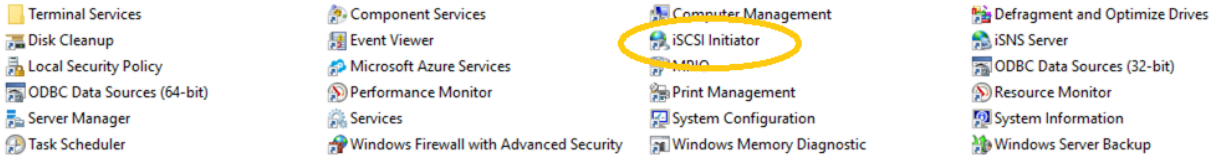
2.2. Configuring iSER Initiator

The iSER interface uses the same IP as the NIC interface. If not already done, assign an IP to Chelsio interface as described in the Assigning IP Address section, before proceeding.

2.3. iSER Target Discovery and Login

The following example describes the method to discover and login to iSER target using the Chelsio iSER interface:

i. To start the initiator configuration, go to **Control Panel** and click **iSCSI Initiator** in **Administrative Tools.**



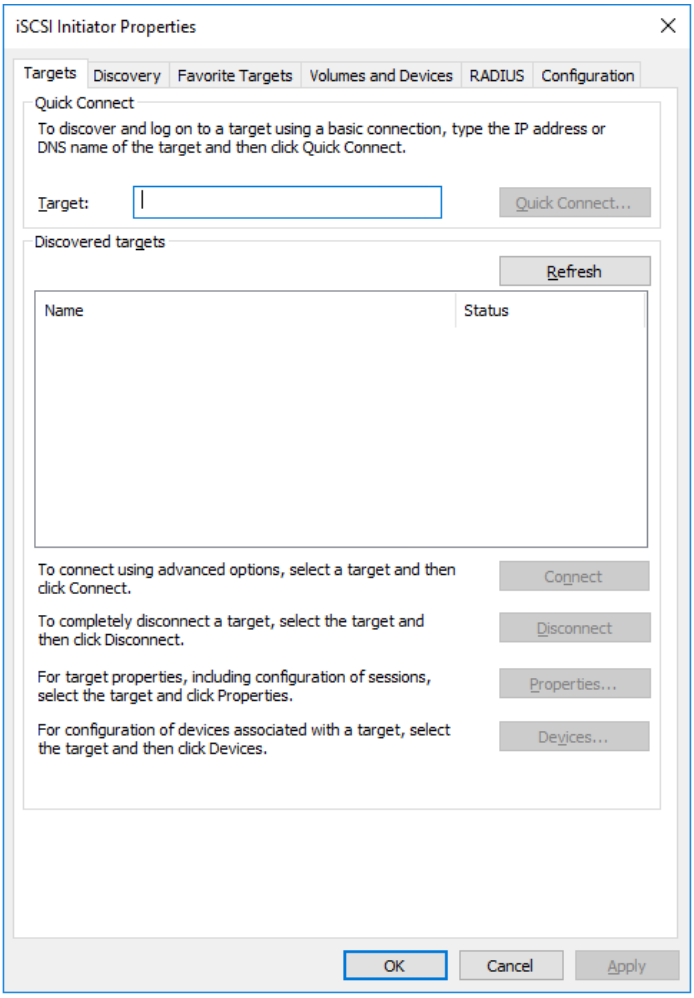


Figure 78 - iSCSI Initiator Properties: Targets tab

ii. Click the **Discovery** tab and then **Discover Portal...**

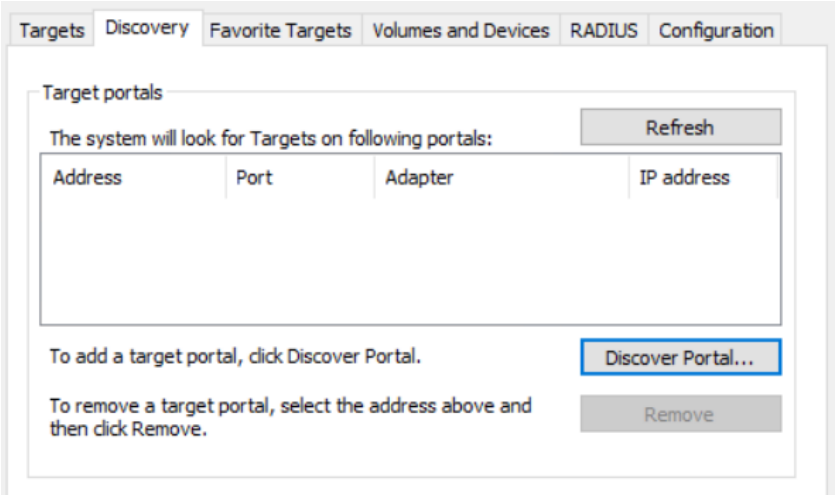


Figure 79 - iSCSI Initiator Properties: Discovery tab

iii. Click on **Advanced…**

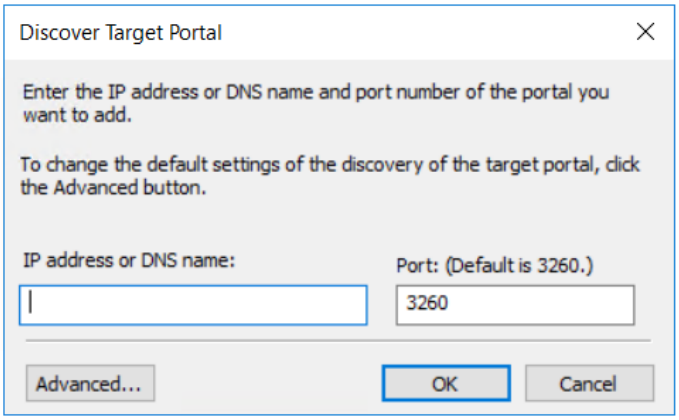


Figure 80 - Discovery Target Portal window

iv. In the **Advanced Settings** window, select **Chelsio iSER Initiator** as the local adapter, the corresponding Chelsio interface IP as the initiator IP and click **OK**.

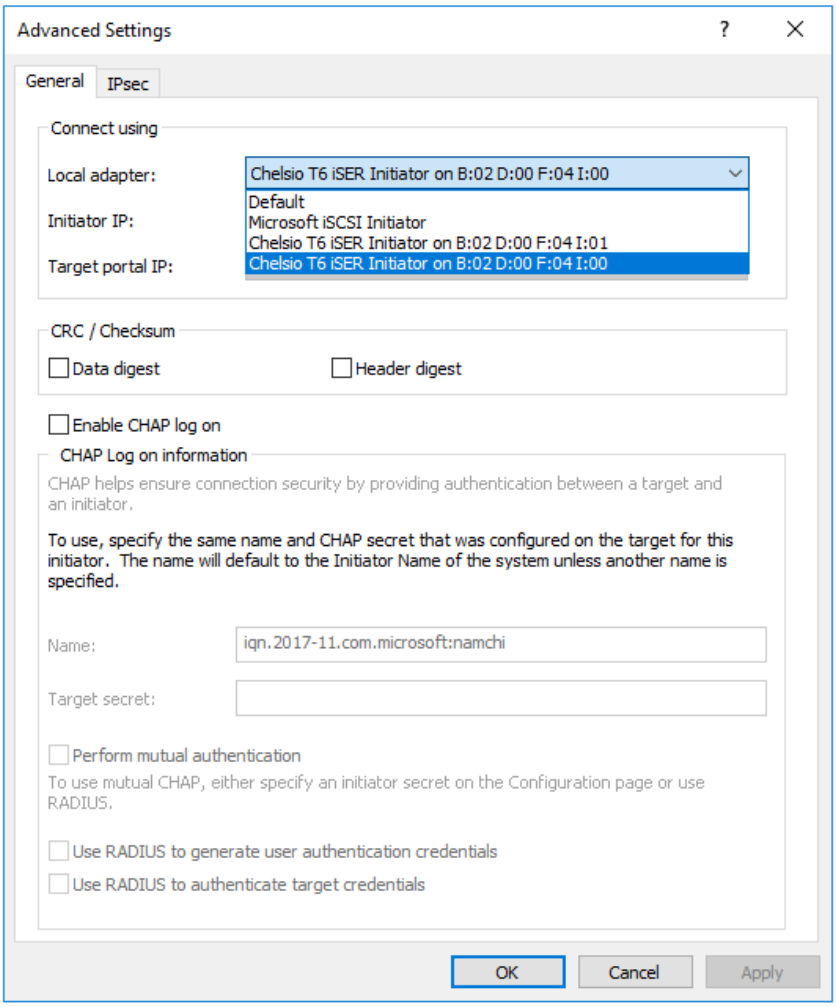


Figure 81 - Discovery Target Portal Advanced Settings: Selecting Local adapter

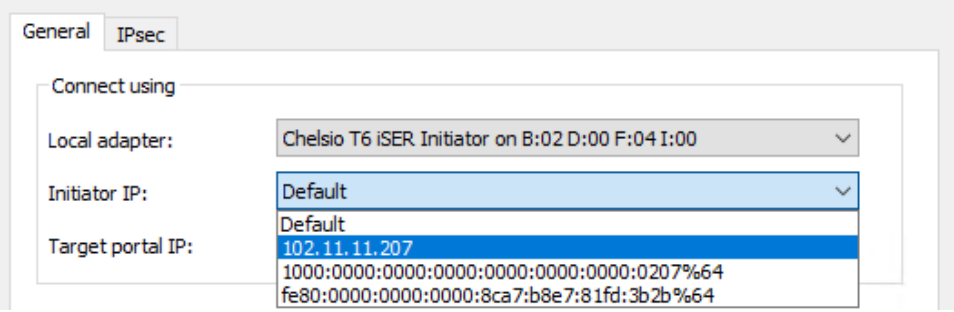


Figure 82 - Discovery Target Portal Advanced Settings: Specifying Initiator IP

v. On the **Discover Target Portal** Window, enter the IP address (DNS name not supported) of the target machine and the corresponding port number, and click **OK**. Note that the default port number for iSER traffic is 3260.

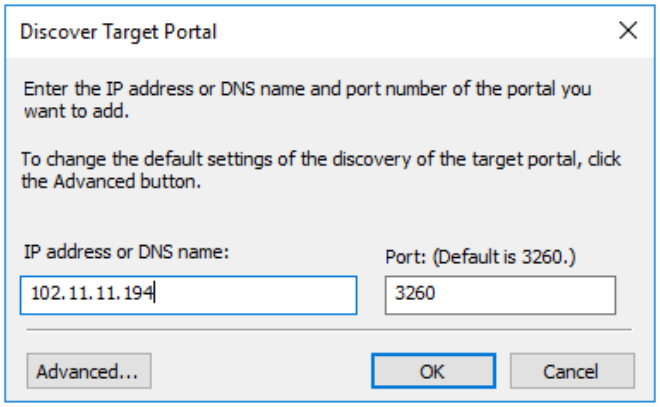


Figure 83 - Adding Target portal

vi. Once target portal is added, details like target and initiator machine IP, port number and Chelsio interface IP will be displayed.

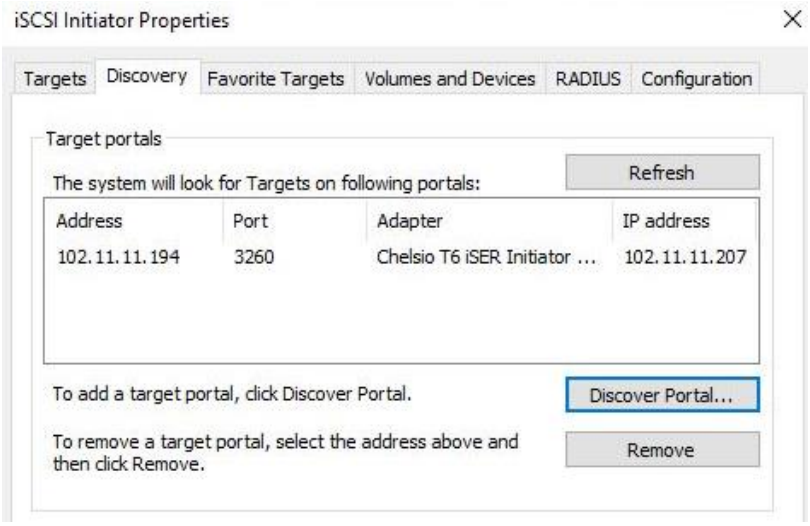


Figure 84 - Target portal added

vii. Click the **Targets** tab to see the list of targets available. Choose a target and click **Connect**.

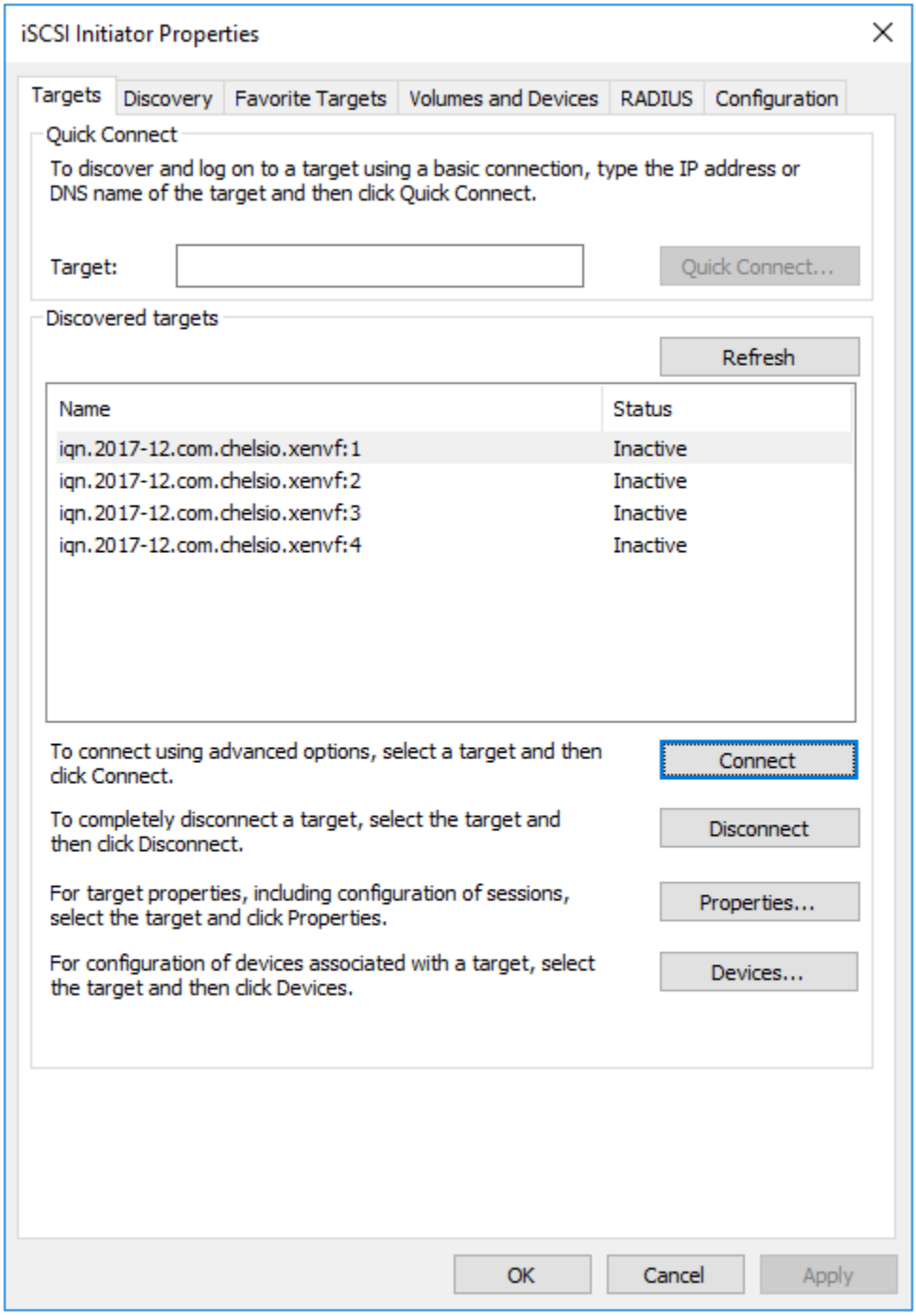


Figure 85 - Targets tab displaying list of available targets

viii. **Connect To Target** window pops up displaying the Target name.

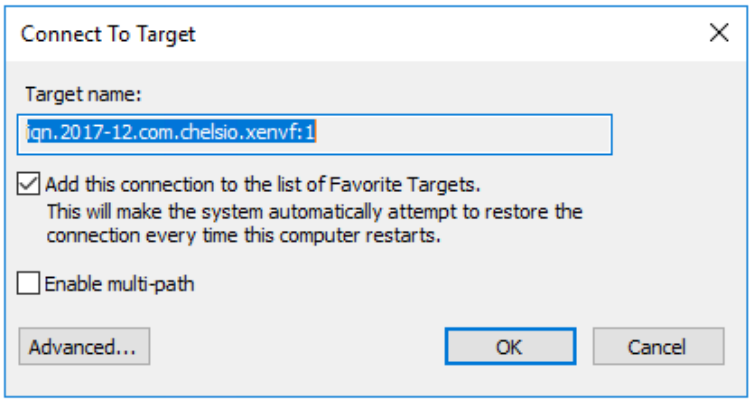


Figure 86 - Connecting to Target

ix. Click **Advanced..**, select Chelsio iSER interface and corresponding Initiator and Target portal IPs. Click **OK**.

Figure 87 - Advanced Settings

x. Click **OK** to close the **Connect To Target** window.

xi. After successful login, target's status will change from *Inactive* to *Connected*.

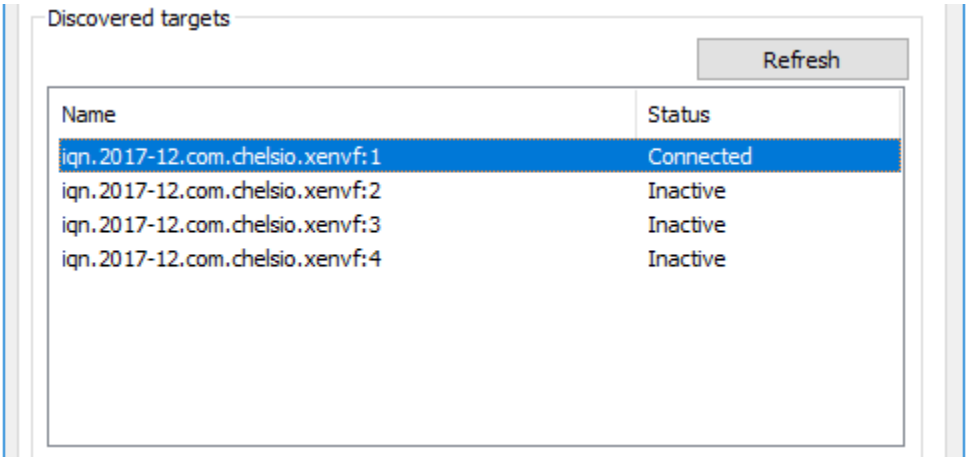


Figure 88 - iSER target connected

xii. The discovered LUN will appear as an offline disk in the **Disk Management** application. Right-click the disk and select *Online*.

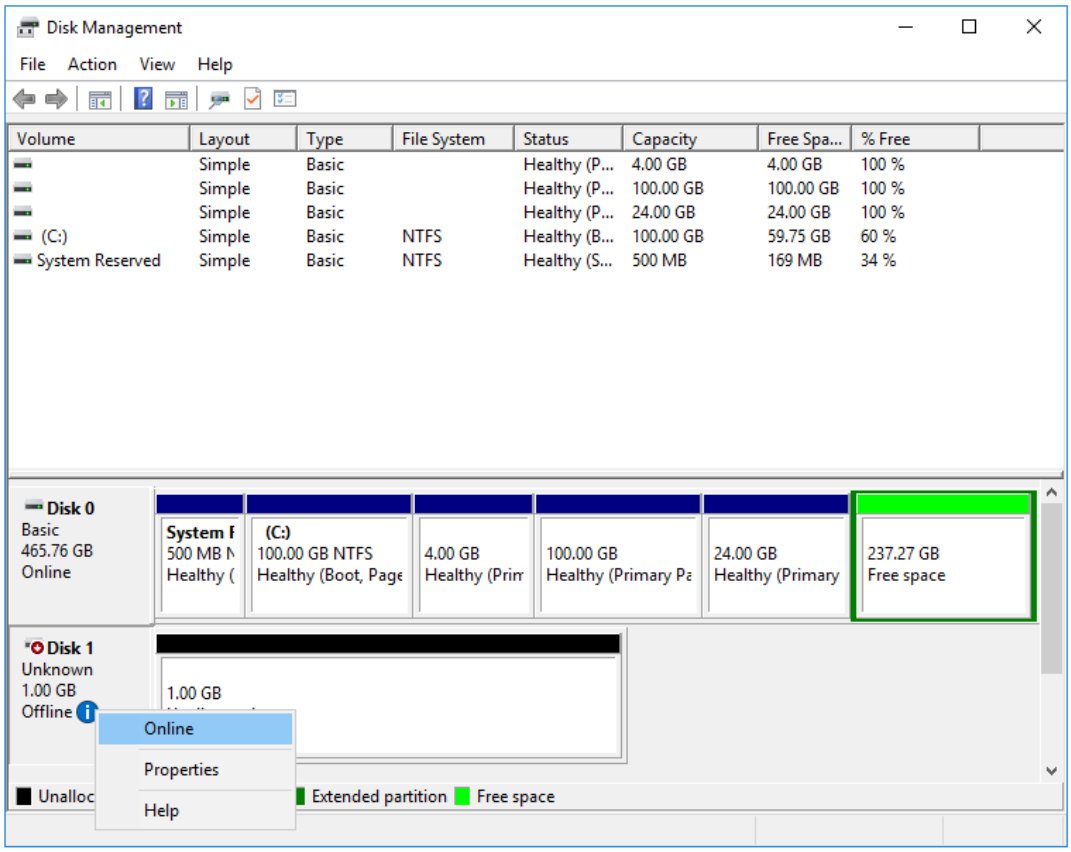


Figure 89 - Disk Management: making volume online

IX. Data Center Bridging (DCB)

1. Introduction

Data Center Bridging (DCB) refers to a set of bridge specification standards, aimed to create a converged Ethernet network infrastructure shared by all storage, data networking and traffic management services. An improvement to the existing specification, DCB uses priority-based flow control to provide hardware-based bandwidth allocation and enhances transport reliability.

One of DCB's many benefits includes low operational cost, due to consolidated storage, server and networking resources, reduced heat and noise, and less power consumption.
Administration is simplified since the specifications enable transport of storage and networking traffic over a single unified Ethernet network.

Note *Only IEEE configuration supported.*

1.1. Hardware Requirements

1.1.1. Supported Adapters

The following are the currently shipping Chelsio adapters that are compatible with Chelsio's DCB feature:

- T62100-LP-CR
- T6225-CR
- T580-CR
- T580-LP-CR
- T540-CR
- T520-CR
- T520-LL-CR
- T520-BT

1.2. Software Requirements

1.2.1. Windows Requirements

Currently Chelsio's DCB feature is available for the following Windows versions:

- Server 2016
- 10 AU Client
- Server 2012 R2

Other versions have not been tested and are not guaranteed to work.

2. Software/Driver Configuration and Fine-tuning

Chelsio network adapters work seamlessly with any legacy switches and do not require the complexity of configuring Data Center Bridging (DCB) protocols either on adapter or network switch. In case DCB is already configured on the switch, the adapter will automatically negotiate the PFC settings configured on the switch.

2.1. Network QoS (optional)

Network Quality of Service (QoS) is an advanced Windows feature that can be used to distribute bandwidth between different kinds of outgoing traffic. This feature ensures efficient usage of resources and minimizes the impact of bandwidth congestion. The bandwidth percentage can be configured either on switch or the host.

2.1.1. Enabling QoS on Host

Network QoS is disabled by default. Use **NDIS Miniport Driver Parameters (**Device manager → Chelsio Network Adapter Properties → Advanced Tab → NDIS QoS) to enable it.

Note *Chelsio QoS requires a minimum of 8 cores to work. In case of multiple sockets, where the number of cores per socket is less than 8, Chelsio adapters must be configured to utilize all the available cores before enabling QoS.*

Run the following command on all Chelsio ports:

```
PS C:\Users\Administrator> Set-NetAdapterRss -Name <interface name> -
MaxProcessorGroup 3
```

2.1.2. Configuring Bandwidth Allocation

- **Switch**

Here is an example of setting bandwidth allocation on a Dell Force10 switch:

i. Log in to the switch and enter the configuration mode:

```
Login: admin
Password:
Force10#enable
Force10#configure
```

ii. Enable DCB and LLDP:

```
Force10(conf)#dcb enable
Force10(conf)#protocol lldp
Force10(conf-lldp)#no disable
```

iii. Set DCBx version to *auto*:

```
Force10(conf-lldp)#dcbx version auto
```

iv. Specify the intervals between hello packets:

```
Force10(conf-lldp)#hello 5
```

v. Create a DCB map. For example, here we are configuring 2 groups/classes:
- Group 0 has BW of 80% with *pfc* enabled.
- Group 1 had BW of 20% with *pfc* disabled.
- Assigning priority groups: Priority 5-7 are under group 0 and 0-4 are under group 1.

```
Force10(conf)#dcb-map win_dcb
Force10(conf-dcbmap-win_dcb)# priority-group 0 bandwidth 80 pfc on
Force10(conf-dcbmap-win_dcb)# priority-group 1 bandwidth 20 pfc off
Force10(conf-dcbmap-win_dcb)# priority-pgid 1 1 1 1 1 0 0 0
```

vi. Enter interface configuration mode and configure protocol LLDP:

```
Force10(conf)#interface tengigabitethernet 0/17
Force10(conf-if-te-0/17)#protocol lldp
Force10(conf-if-te-0/17-lldp)#no disable
```

vii. Set DCBx version to *auto*:

```
Force10(conf-if-te-0/17-lldp)#dcbx version auto
```

viii. Apply the DCB-MAP created:

```
Force10(conf-if-te-0/17)#dcb-map win_dcb
```

Now the host adapter will honor the bandwidth allocation settings configured on the switch.

- **Host**

Here is an example of setting bandwidth allocation on the host:

i. Open PowerShell with administrative privileges and run the following command:

```
PS C:\Users\Administrator> Install-WindowsFeature -Name "Data-Center-
Bridging" -IncludeAllSubFeature -IncludeManagementTools
```

ii. Create new policy:

```
PS C:\Users\Administrator> New-NetQosPolicy -Name rdma -PriorityValue 5 -
NetDirectPortMatchCondition 445
```

iii. Create new rule with bandwidth percentage:

```
PS C:\Users\Administrator> New-NetQosTrafficClass -Name rdma -Priority 5 -
Algorithm ETS -BandwidthPercentage 80 -Verbose
```

The above command allocates 80% of the bandwidth to the policy named 'rdma'.

To change the percentage of an existing ETS rule:

```
PS C:\Users\Administrator> Set-NetQosTrafficClass -Name rdma -Priority 5 -
Algorithm ETS -BandwidthPercentage 10 -Verbose
```

X. Appendix

Chelsio End-User License Agreement (EULA)

Installation and use of the driver/software implies acceptance of the terms in the Chelsio End-User License Agreement (EULA).

IMPORTANT: PLEASE READ THIS SOFTWARE LICENSE CAREFULLY BEFORE DOWNLOADING OR OTHERWISE USING THE SOFTWARE OR ANY ASSOCIATED DOCUMENTATION OR OTHER MATERIALS (COLLECTIVELY, THE "SOFTWARE"). BY CLICKING ON THE "OK" OR "ACCEPT" BUTTON YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, CLICK THE "DO NOT ACCEPT" BUTTON TO TERMINATE THE INSTALLATION PROCESS.

1. License. Chelsio Communications, Inc. ("Chelsio") hereby grants you, the Licensee, and you hereby accept, a limited, non-exclusive, non-transferable license to install and use the Software with one or more Chelsio network adapters on a single server computer for use in communicating with one or more other computers over a network. You may also make one copy of the Software in machine readable form solely for back-up purposes, provided you reproduce Chelsio's copyright notice and any proprietary legends included with the Software or as otherwise required by Chelsio.

2. Restrictions. This license granted hereunder does not constitute a sale of the Software or any copy thereof. Except as expressly permitted under this Agreement, you may not:

(i) reproduce, modify, adapt, translate, rent, lease, loan, resell, distribute, or create derivative works of or based upon, the Software or any part thereof; or

(ii) make available the Software, or any portion thereof, in any form, on the Internet. The Software contains trade secrets and, in order to protect them, you may not decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human-perceivable form. You assume full responsibility for the use of the Software and agree to use the Software legally and responsibly.

3. Ownership of Software. As Licensee, you own only the media upon which the Software is recorded or fixed, but Chelsio retains all right, title and interest in and to the Software and all subsequent copies of the Software, regardless of the form or media in or on which the Software may be embedded.

4. Confidentiality. You agree to maintain the Software in confidence and not to disclose the Software, or any information or materials related thereto, to any third party without the express written consent of Chelsio. You further agree to take all reasonable precautions to limit access of the Software only to those of your employees who reasonably require such access to perform their employment obligations and who are bound by confidentiality agreements with you.

5. Term. This license is effective in perpetuity, unless terminated earlier. You may terminate the license at any time by destroying the Software (including the related documentation), together with all copies or modifications in any form. Chelsio may terminate this license, and this license shall be deemed to have automatically terminated, if you fail to comply with any term or condition of this Agreement. Upon any termination, including termination by you, you must destroy the Software (including the related documentation), together with all copies or modifications in any form.

6. Limited Warranty. If Chelsio furnishes the Software to you on media, Chelsio warrants only that the media upon which the Software is furnished will be free from defects in

material or workmanship under normal use and service for a period of thirty (30) days from the date of delivery to you.

CHELSIO DOES NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR ANY PART THEREOF. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, CHELSIO MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, AND HEREBY DISCLAIMS ALL OTHER WARRANTIES, INCLUDING, BUT NOT LIMITED TO, NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.

7. Remedy for Breach of Warranty. The sole and exclusive liability of Chelsio and its distributors, and your sole and exclusive remedy, for a breach of the above warranty, shall be the replacement of any media furnished by Chelsio not meeting the above limited warranty and which is returned to Chelsio. If Chelsio or its distributor is unable to deliver replacement media which is free from defects in materials or workmanship, you may terminate this Agreement by returning the Software.

8. Limitation of Liability. IN NO EVENT SHALL CHELSIO HAVE ANY LIABILITY TO YOU OR ANY THIRD PARTY FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, HOWEVER CAUSED, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE LICENSE OR USE OF THE SOFTWARE, INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR LOSS OF ANTICIPATED PROFITS, EVEN IF CHELSIO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL CHELSIO'S LIABILITY ARISING OUT OF OR RELATED TO THE LICENSE OR USE OF THE SOFTWARE EXCEED THE AMOUNTS PAID BY YOU FOR THE LICENSE GRANTED HEREUNDER. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY.

9. High Risk Activities. The Software is not fault-tolerant and is not designed, manufactured or intended for use or resale as online equipment control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the Software could lead directly to death, personal injury, or severe physical or environmental damage. Chelsio specifically disclaims any express or implied warranty of fitness for any high risk uses listed above.

10. Export. You acknowledge that the Software is of U.S. origin and subject to U.S. export jurisdiction. You acknowledge that the laws and regulations of the United States and other countries may restrict the export and re-export of the Software. You agree that you will not export or re-export the Software or documentation in any form in violation of applicable United States and foreign law. You agree to comply with all applicable international and national laws that apply to the Software, including the U.S.

Export Administration Regulations, as well as end-user, end-use, and destination restrictions issued by U.S. and other governments.

11. Government Restricted Rights. The Software is subject to restricted rights as follows. If the Software is acquired under the terms of a GSA contract: use, reproduction or disclosure is subject to the restrictions set forth in the applicable ADP Schedule contract. If the Software is acquired under the terms of a DoD or civilian agency contract, use, duplication or disclosure by the Government is subject to the restrictions of this Agreement in accordance with 48 C.F.R. 12.212 of the Federal

Acquisition Regulations and its successors and 49 C.F.R. 227.7202-1 of the DoD FAR Supplement and its successors.

12. General. You acknowledge that you have read this Agreement, understand it, and that by using the Software you agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between Chelsio and you, and supersedes any proposal or prior agreement, oral or written, and any other communication between Chelsio and you relating to the subject matter of this Agreement. No additional or any different terms will be enforceable against Chelsio unless Chelsio gives its express consent, including an express waiver of the terms of this Agreement, in writing signed by an officer of Chelsio. This Agreement shall be governed by California law, except as to copyright matters, which are covered by Federal law. You hereby irrevocably submit to the personal jurisdiction of, and irrevocably waive objection to the laying of venue (including a waiver of any argument of forum non conveniens or other principles of like effect) in, the state and federal courts located in Santa Clara County, California, for the purposes of any litigation undertaken in connection with this Agreement. Should any provision of this Agreement be declared unenforceable in any jurisdiction, then such provision shall be deemed severable from this Agreement and shall not affect the remainder hereof. All rights in the Software not specifically granted in this Agreement are reserved by Chelsio. You may not assign or transfer this Agreement (by merger, operation of law or in any other manner) without the prior written consent of Chelsio and any attempt to do so without such consent shall be void and shall constitute a material breach of this Agreement.

Should you have any questions concerning this Agreement, you may contact Chelsio by writing to:

Chelsio Communications, Inc.
209 North Fair Oaks Avenue,
Sunnyvale, CA 94085
U.S.A