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

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
- RDMA/NVGRE concurrent (Mode 2)
- iWARP (ND)
- NDIS SR-IOV
- Guest RDMA (Mode 3)
- VMMQ/vRSS
- iSCSI Storport Miniport
- Data Center Bridging (DCB)
- Storage Replica (SR)
- Unified Wire Manager (UM)
- UM for Win PE

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.*

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, UM installer, UM for Win PE application, firmware binaries, adapter configuration binaries and support documentation (*docs* folder).

- ### Nano Server Installer

The Nano Server Installer 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.
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

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

3.1.1. Unified Wire Installer

Download and install the latest Microsoft Visual C++ Redistributable Packages for Visual Studio (x86 and x64).

3.1.2. NDIS SR-IOV

Enable SR-IOV in the machine.

3.1.3. Unified Wire Manager (Installer & Zip Package)

The Unified Wire installer has been designed to install Unified Wire Manager (UM) along with driver components by default, hence no separate installation is required. In case of zip package, you will have to manually run the UM installer located in *ChelsioUwire-x.x.x.xx/UM* folder (See Unified Wire Manager chapter for for more information). Based on the Windows version running on the system, the three UM components, i.e. *Management Agent*, *Client* and *Station* will be installed. If an older version of UM exists, the installer will upgrade it to the version provided in the package.

⦿ **Management Agent**

If you wish to install Management Agent, please make sure that **Microsoft .Net Framework 3.5** is installed before proceeding with the installation.

⦿ **Management Station**

If you wish to install Management Station, please make sure that the following requirements are met before proceeding with the installation:

i.   Install **Python 2.6.6 (32-bit).** (Download from here)
ii. Ensure that the path to python binary (typically *C:\Python26*), is added to PATH system variable.
iii. Install **Apache HTTP Server 2.2** with SSL. (Download from here)
iv. If the Apache Server is running, it should be stopped before starting the installation process.

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.
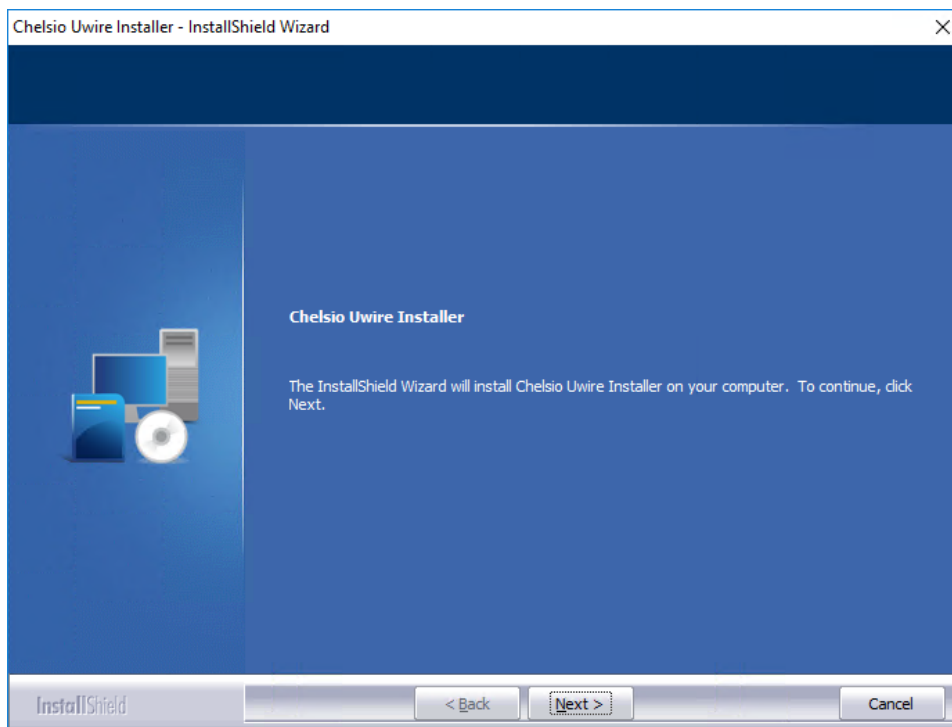
Figure 1 - Unified Wire installer welcome window

iii. Select the radio button **I accept the terms of the license agreement** and click **Next**.
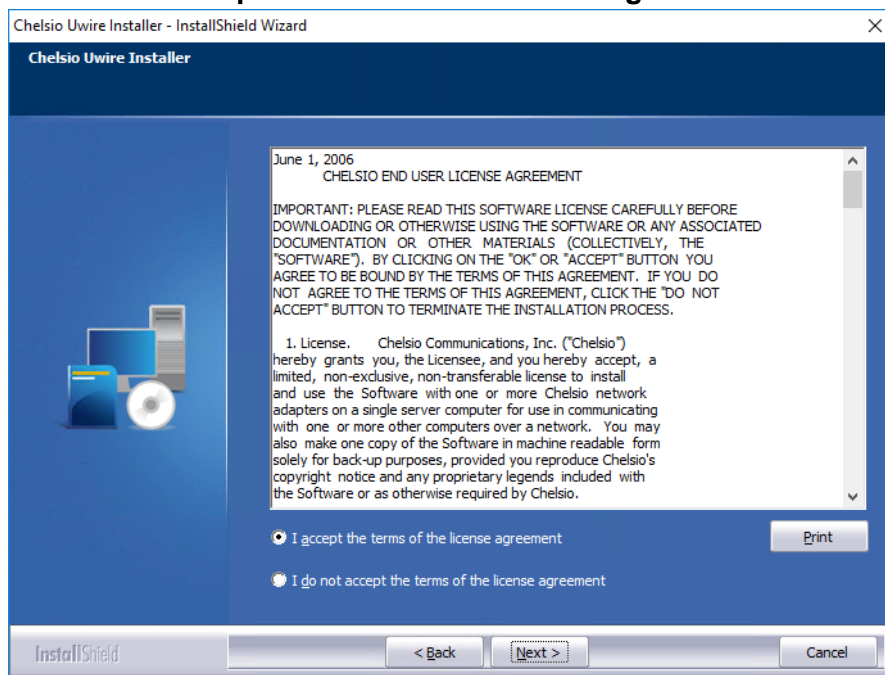
Figure 2 - Chelsio EULA window

iv. Now, either select **Complete** for complete package installation or else select **Custom** radio button to customize the installation.
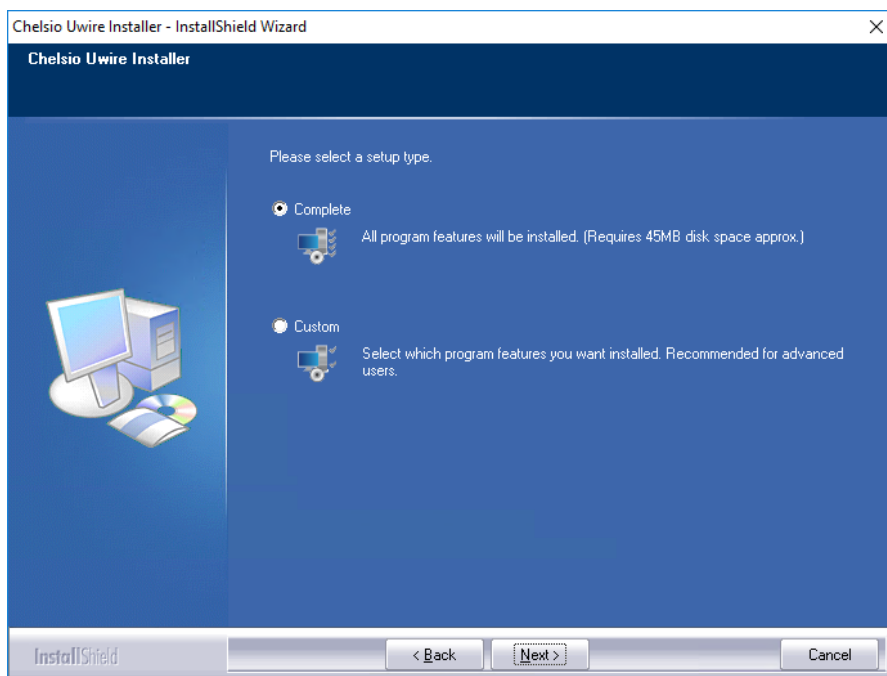
Figure 3 - Select setup (installation) type

If you select **Custom**, you can choose whether to install iSCSI Storport Miniport driver or not. Please note that Chelsio Ethernet Driver (NIC and VBD) is required for iSCSI Storport Miniport to work.
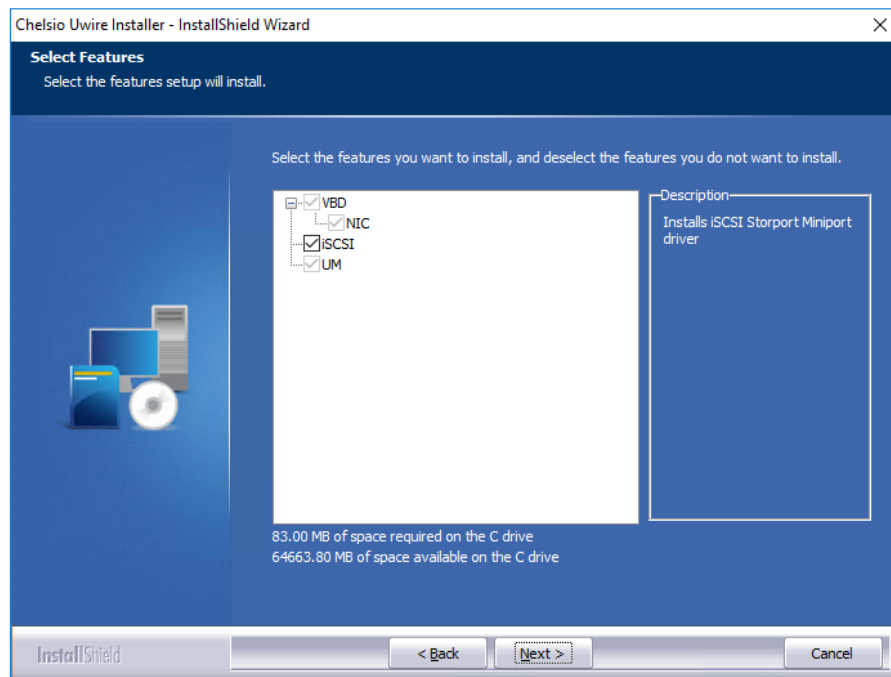
Figure 4 - Customizing the installation

v.   Click **Install** to start the installation.
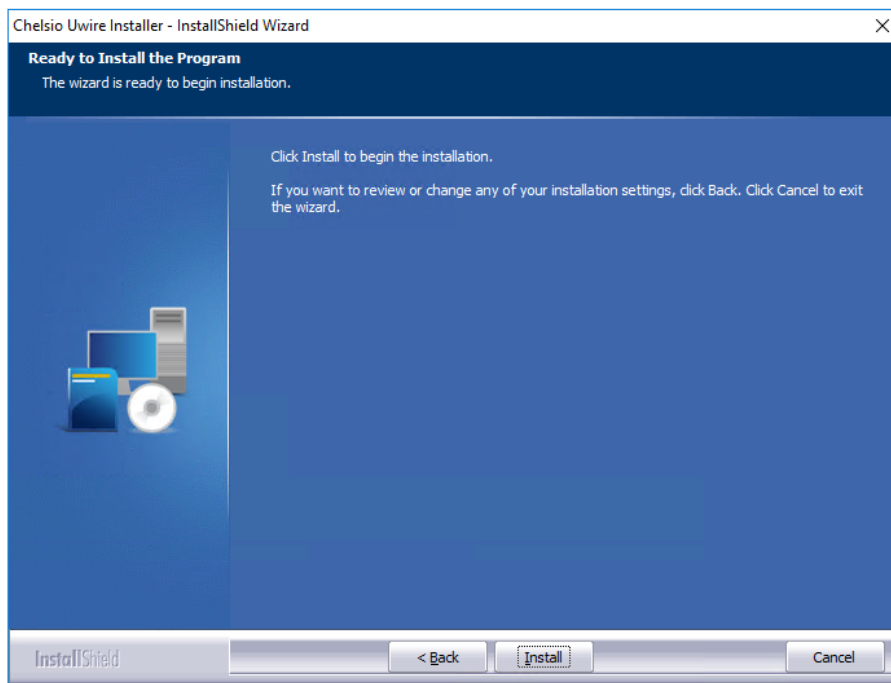
Figure 5 - Start installation

vi. Selected driver components will now be installed.
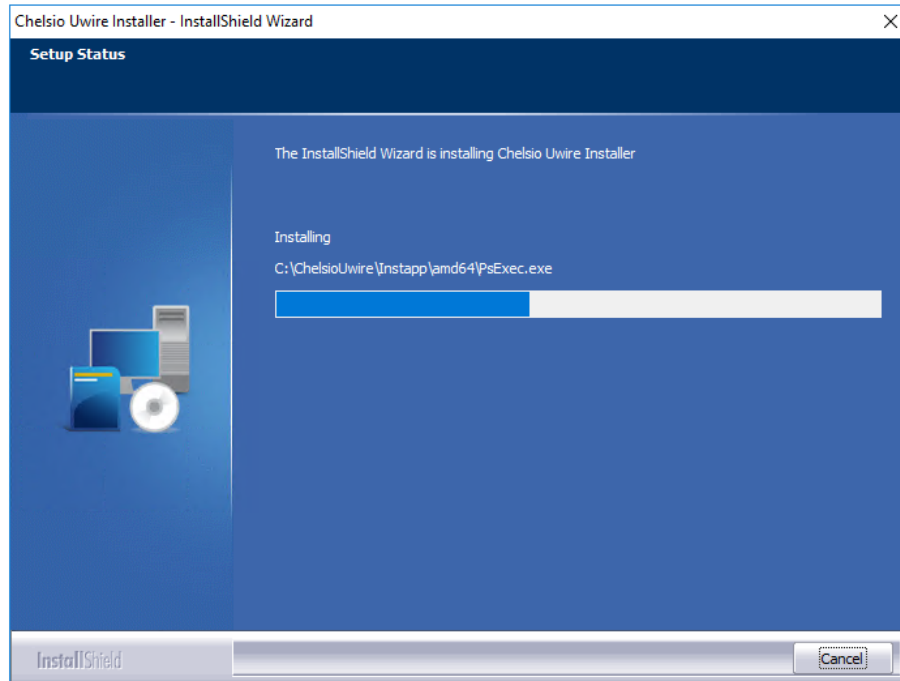
Figure 6 - Installation in progress

vii. After successful installation of driver(s), the Unified Wire Manager installer will be invoked. Click **Next** for the Chelsio End User License Agreement Window.
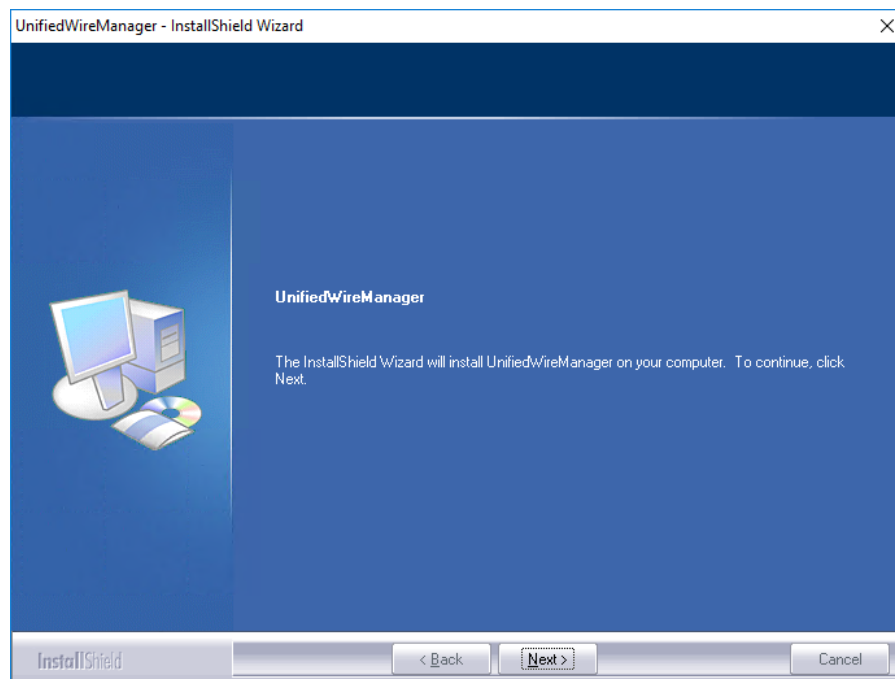
Figure 7 - UM installer welcome window

viii. Select the radio button **I accept the terms of the license agreement** and click **Next**.
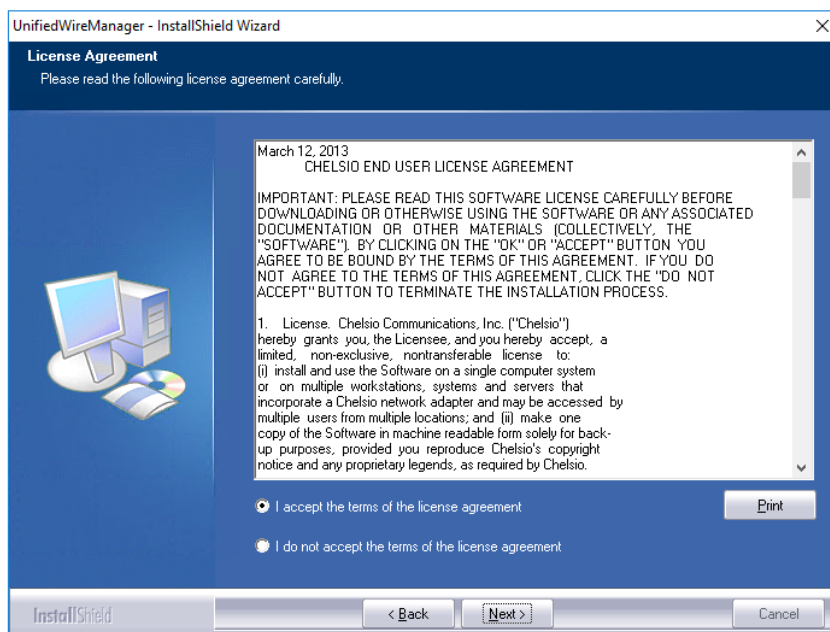
Figure 8 - Chelsio EULA window

ix. The next window will display the prerequisites for various UM components. Ensure that they are met before proceeding. Click **Next**.
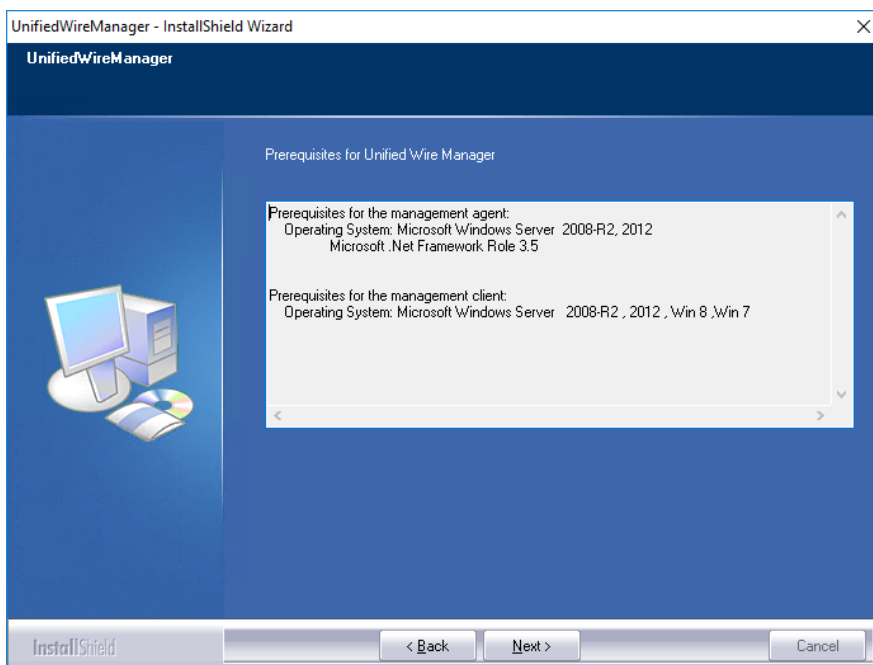
Figure 9 - UM prerequisites window

ⓘ Note *If prerequisites mentioned are not met, installation of UM will fail. Driver installation will however continue.*

x. Now, either select **Complete** for complete package installation or else select **Custom** radio button to customize the installation. The **Custom** option provides the option to change the installation path and which UM components to install. Click **Next**.

> *ⓘ Note*   *If Management Station is selected, please make sure that all related prerequisites are met before proceeding (See* Prerequisites*) or else the* component will be skipped during installation.*
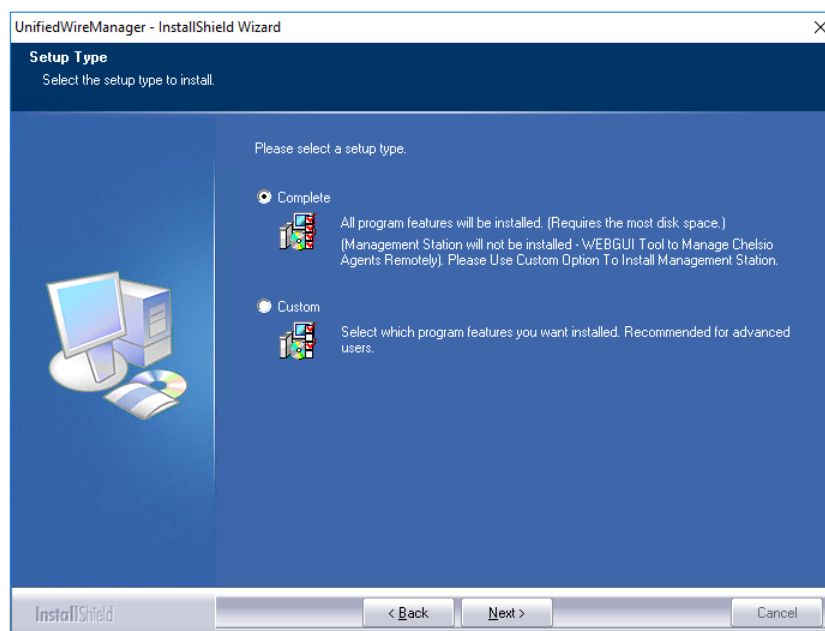
Figure 10 - Select setup (installation) type
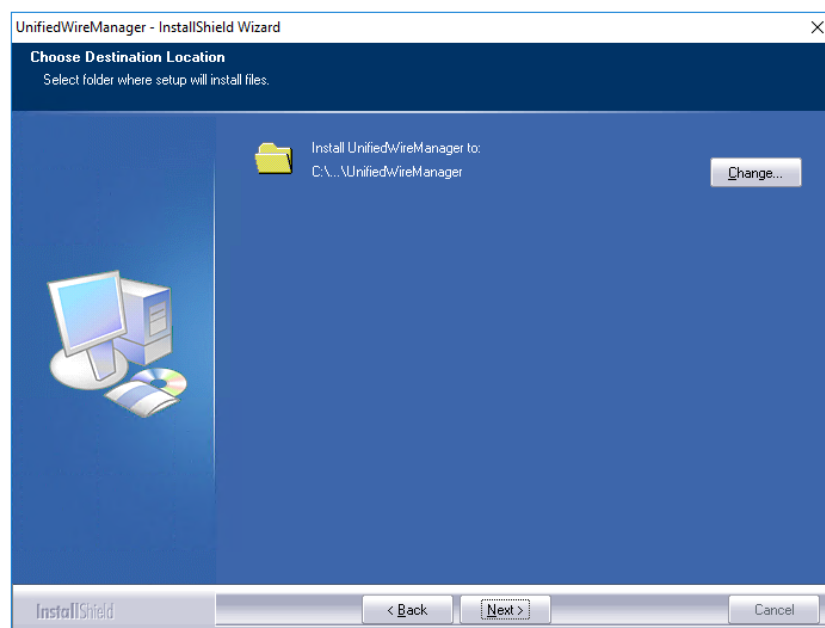
Figure 11 - Changing UM installation path
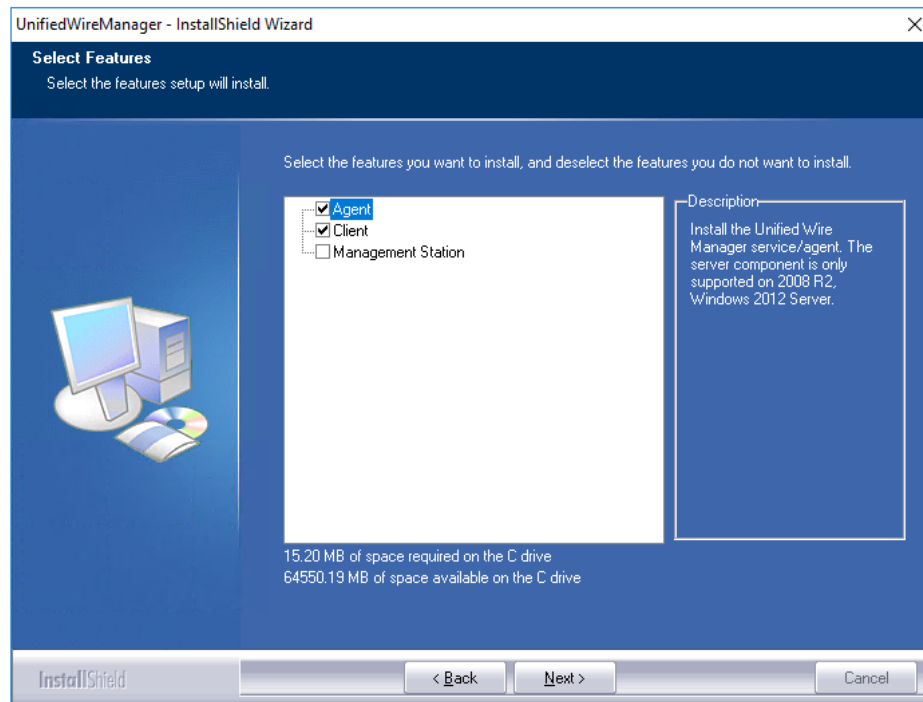
Figure 12 - Selecting UM components to install

xi. Click **Install** to start the installation. Unified Wire Manager will now be installed with the selected options.
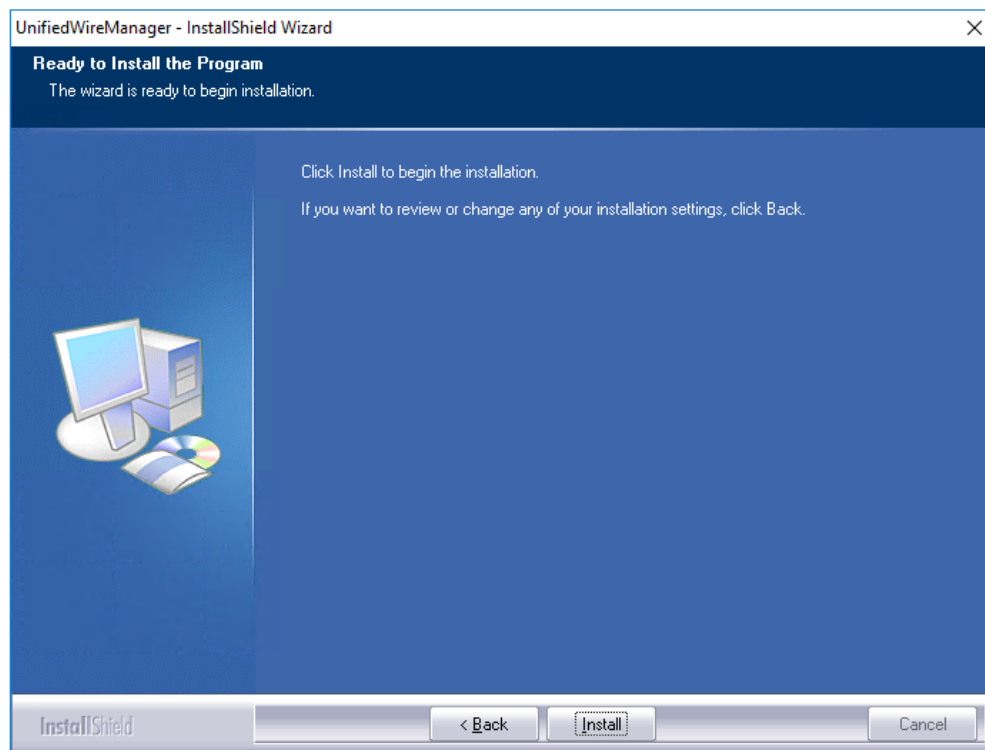
Figure 13 - Starting UM installation

xii. Click **Finish** to exit from the UM Installer.
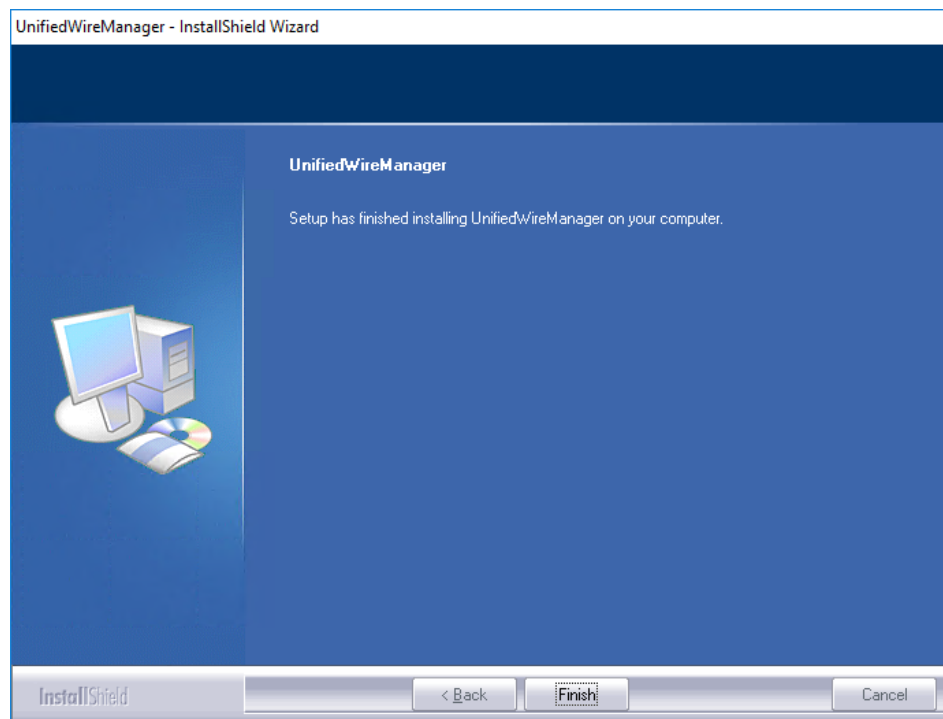
Figure 14 - Finishing UM installation

xiii. Click **Finish** to exit from the Unified Wire Installer.
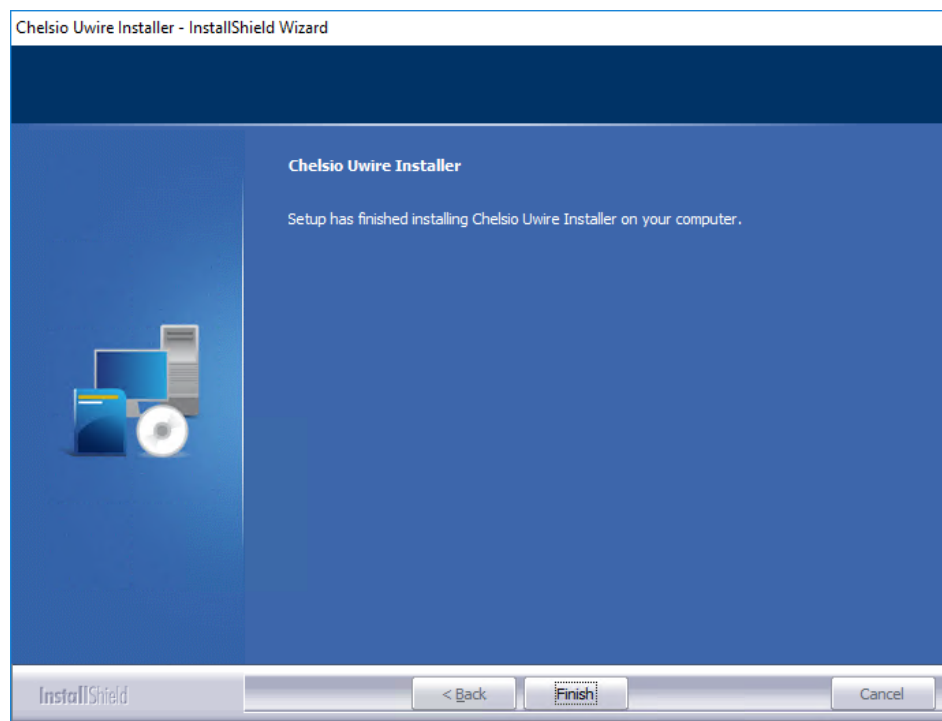
Figure 15 - Finishing Unified Wire installation

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 the drivers and UM Agent.

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. Nano Server Installer

Follow the steps mentioned below to install Unified Wire on a Nano Server machine:

i.   Download and unzip the driver package *Chelsio-NANO-installer-x.x.x.xx.zip*

ii. Connect to Nano Server machine and copy the package contents.

iii. Open PowerShell with administrative privileges and change your working directory to *Chelsio-NANO-installer-x.x.x.xx*

iv. Run the installer

```
[nanomachine-ip]: PS D:\Chelsio-NANO-installer-x.x.x.xx> .\install.ps1 -in
all
```

A log file, *Chelsio-installer.log*, containing installation summary will be created in the same folder.

i Note *In case of non-WHQLed drivers, if Nano Server is freshly installed, the installer will enable Test Signing. Reboot the machine and run the installer again.*

To know more about Installer parameters and options, view the help:

```
[nanomachine-ip]: PS D:\Chelsio-NANO-installer-x.x.x.xx> .\install.ps1 -h
```

3.4. 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.

3.4.1. 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.

3.4.2. 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** file and click **Open** and then **OK.** 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.

- **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.

3.4.3. 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

3.4.4. Nano Server: Driver Installation

Before installing Nano Server, the Chelsio driver components need to be added to the boot image file. The following steps explain step-by-step procedure to add Chelsio VBD, NDIS and Null drivers to the image file:

i. Copy *NanoServer.wim* image file from Windows Server 2016 (build>=10586) ISO to a desired folder on your hard drive.

ii. Open a command prompt with administrative privileges (elevated command prompt).

iii. Determine the index number of image file.

```
C:\Windows\system32>dism /get-wiminfo /wimfile:<path>\NanoServer.wim
```

iv. Mount the image file:

```
C:\Windows\system32>dism /mount-wim /wimfile:<path>\NanoServer.wim
/index:<index_number> /mountdir:<mount_directory>
```

v. Copy the Microsoft OEM driver set package from *NanoServer\Packages* folder in the ISO to your hard drive and add it to the image file:

```
C:\Windows\system32>dism /image:<path>\<mount_directory> /add-package
/packagepath:<path>\Microsoft-NanoServer-OEM-Drivers-Package.cab
```

vi. Add Chelsio drivers to the image file:

```
C:\Windows\system32> dism /image:<path>\<mount_directory> /add-driver
/driver:<path>\ChelsioUwire-x.x.x.xx\chvbdx64.inf
```

```
C:\Windows\system32> dism /image:<path>\<mount_directory> /add-driver
/driver:<path>\ChelsioUwire-x.x.x.xx\chnetx64.inf
```

```
C:\Windows\system32> dism /image:<path>\<mount_directory> /add-driver
/driver:<path>\ChelsioUwire-x.x.x.xx\chnulx64.inf
```

ⓘ Note   *In case of non-WHQLed drivers, run the above commands with the /forceunsigned option. Here's an example for the first command:*

```
C:\Windows\system32> dism /image:<path>\<mount_directory> /add-driver
/driver:<path>\ChelsioUwire-x.x.x.xx\chvbdx64.inf /forceunsigned
```

vii. Commit and save the image file:

```
C:\Windows\system32>dism /unmount-wim /mountdir:<mount_directory> /commit
```

viii. Finally, add the image file to WDS server and install Nano Server (with Chelsio drivers) using PXE boot.

ix. The Nano Server machine will reboot.

> ℹ️ Note *In case of non-WHQLed drivers, run the following command to enable test signing and reboot the machine again:*

```
[nanomachine-ip]: PS D:\> bcdedit /set testsigning on
```

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.

For Nano Server Installer, the binaries will be present in *Chelsio-NANO-installer-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..**
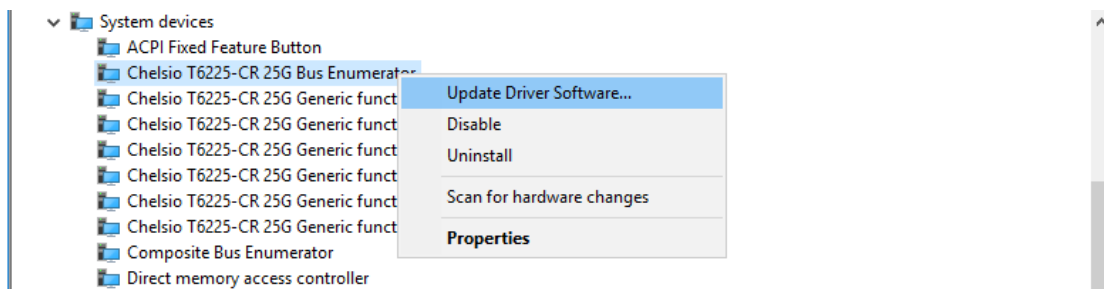
Figure 16 – Updating VBD driver

iii. Select **Search automatically for updated driver software**.
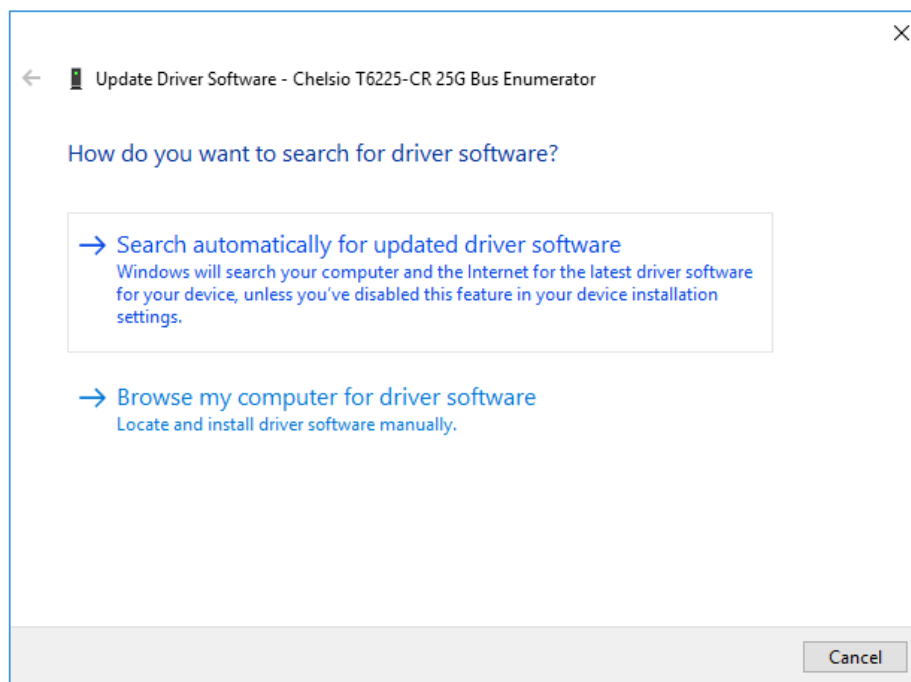
Figure 17 – Searching for driver using Windows update

iv. Windows will search and install the latest driver. Click **Close**.
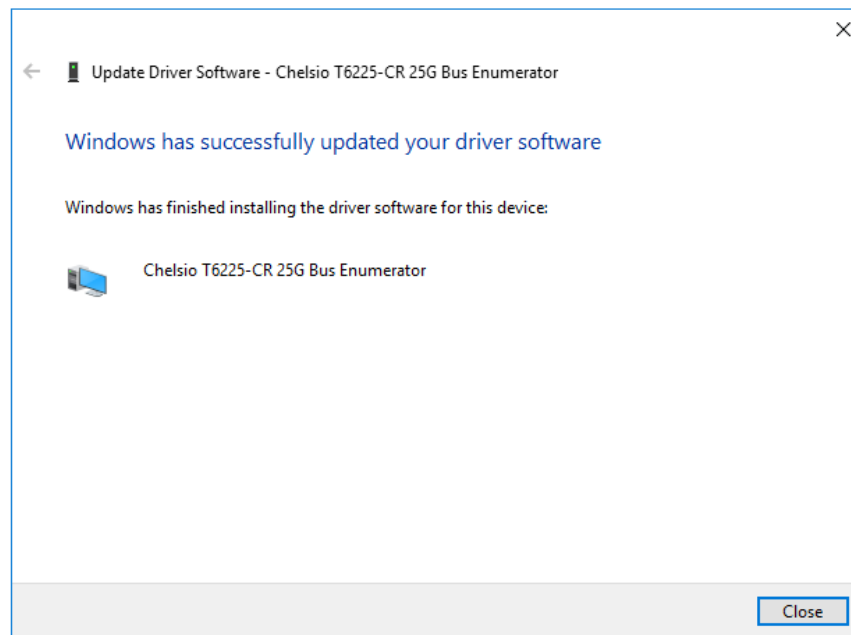
Figure 18 - Driver update successful

v. Click on **Network Adapters**, right-click on **Chelsio Network Adapter** and select **Update Driver Software..**
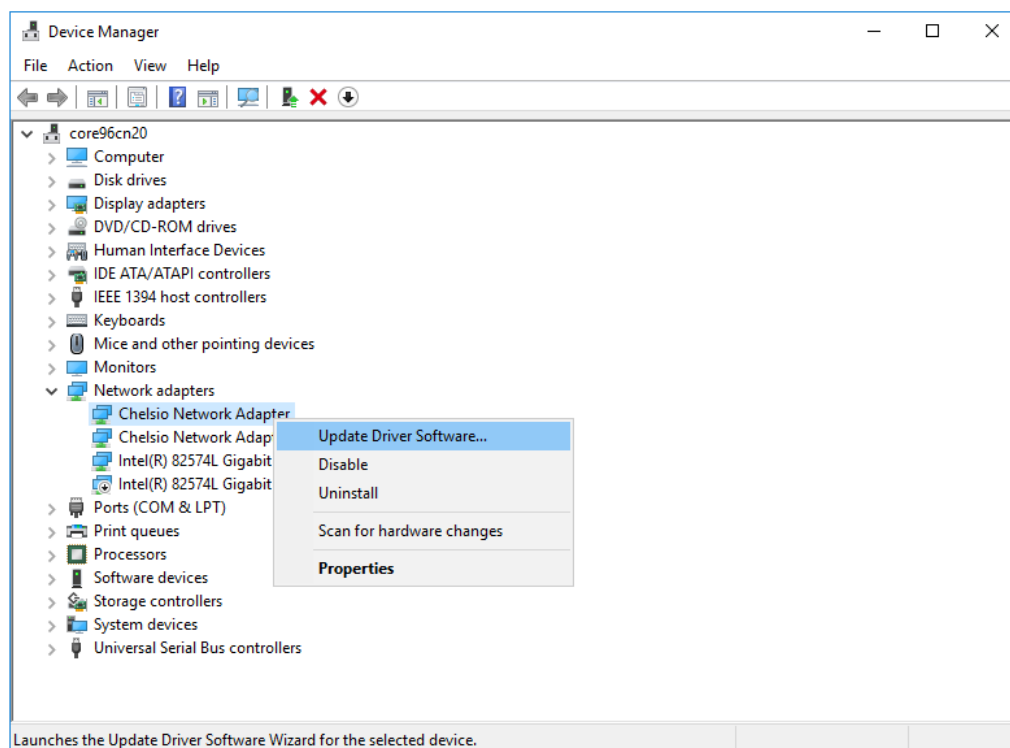
Figure 19 - Updating NDIS driver

vi. Select **Search automatically for updated driver software**
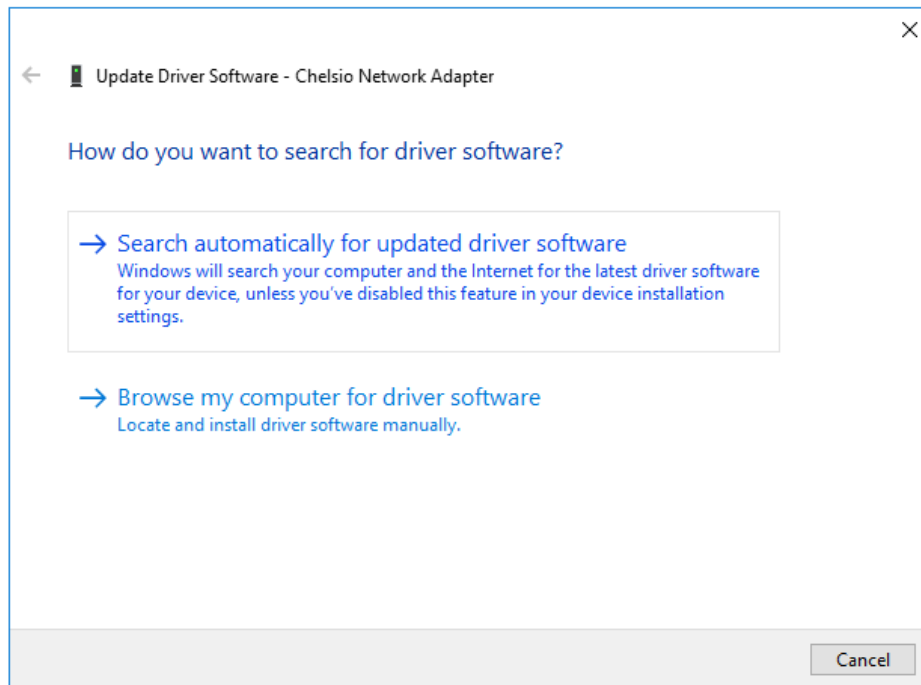
Figure 20- Searching for driver using Windows update

vii. Windows will search and install the latest driver. Click **Close**.
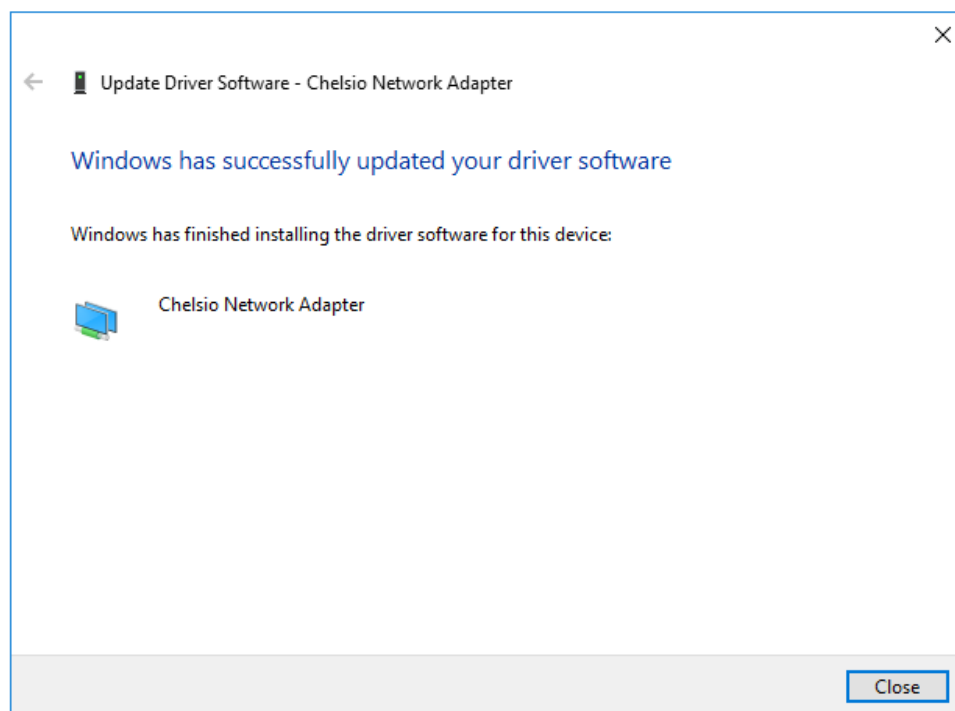
Figure 21 - Driver update successful

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.*
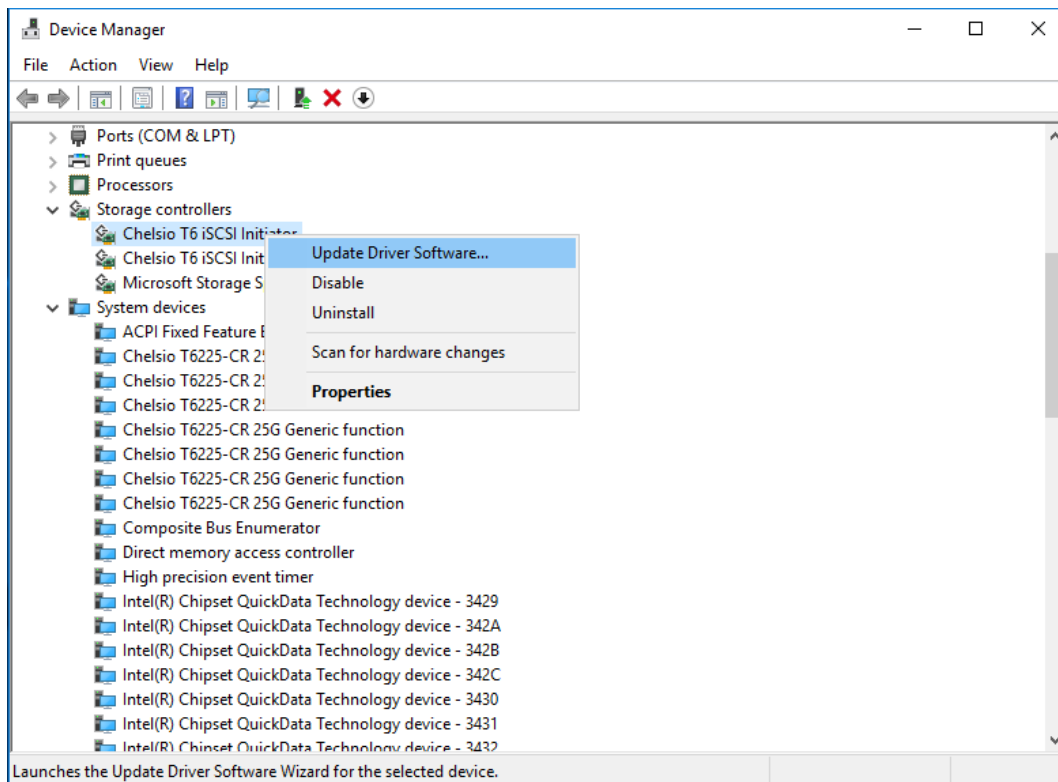
Figure 22 - Updating iSCSI Storport Miniport driver

ix. Windows will search and install the latest driver. Click **Close**.
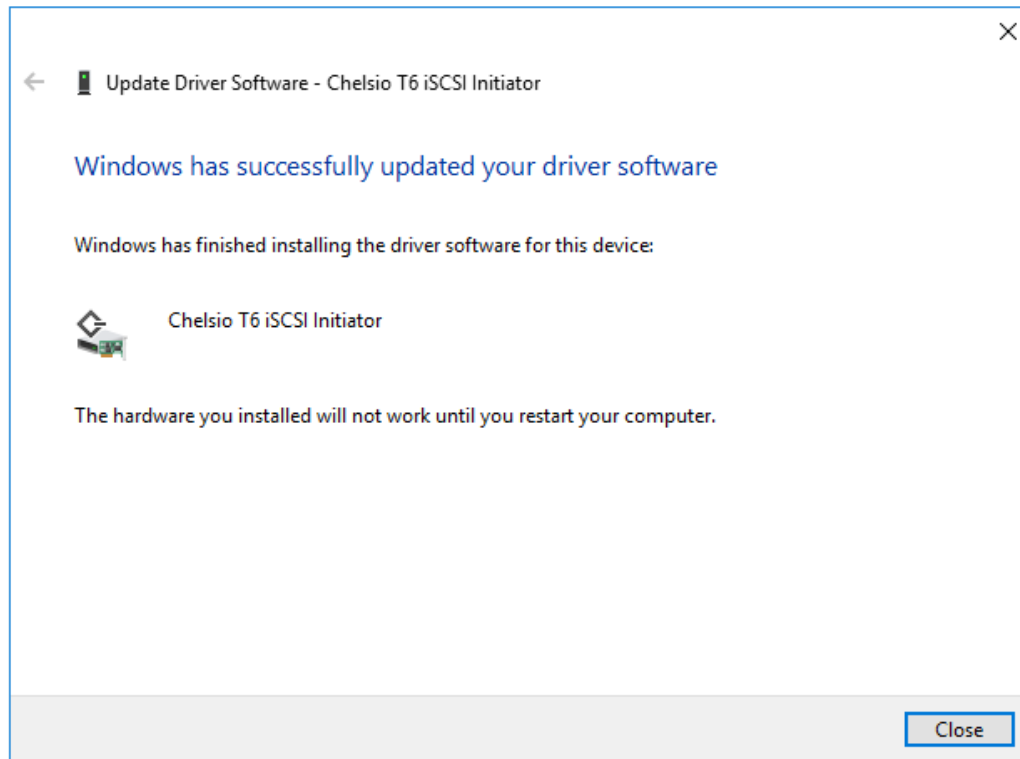
Figure 23 - Driver update successful

x. Open the **Device Manager,** click on **Network Adapters**, right-click on **Chelsio Generic Function** and select **Update Driver Software..**
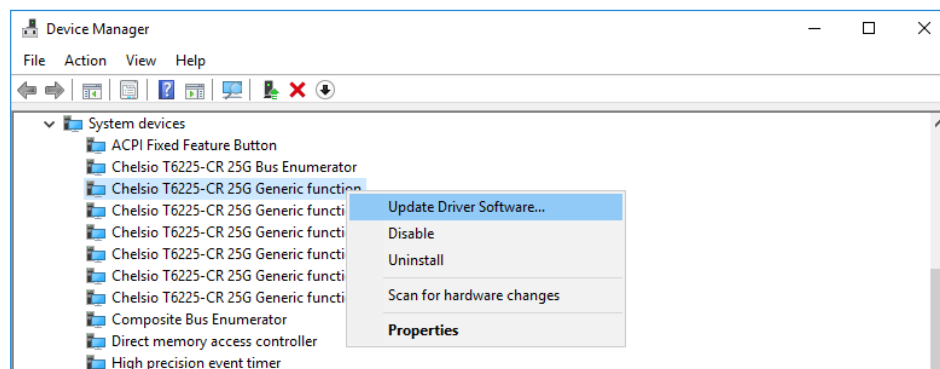
Figure 24 - Updating Generic function driver

xi. Select **Search automatically for updated driver software**
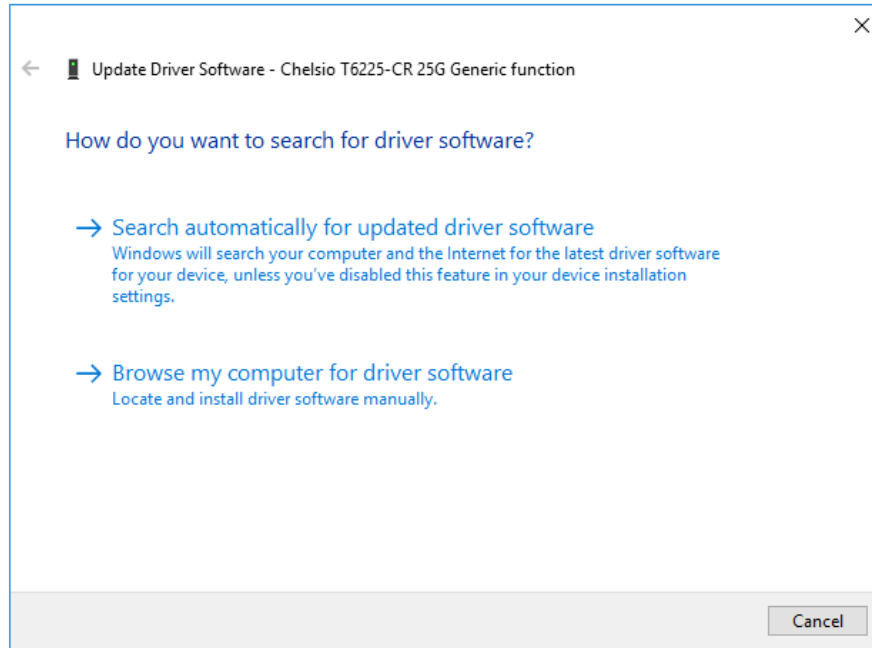
Figure 25 - Searching for driver using Windows update

xii. Windows will search and install the latest driver. Click **Close**.
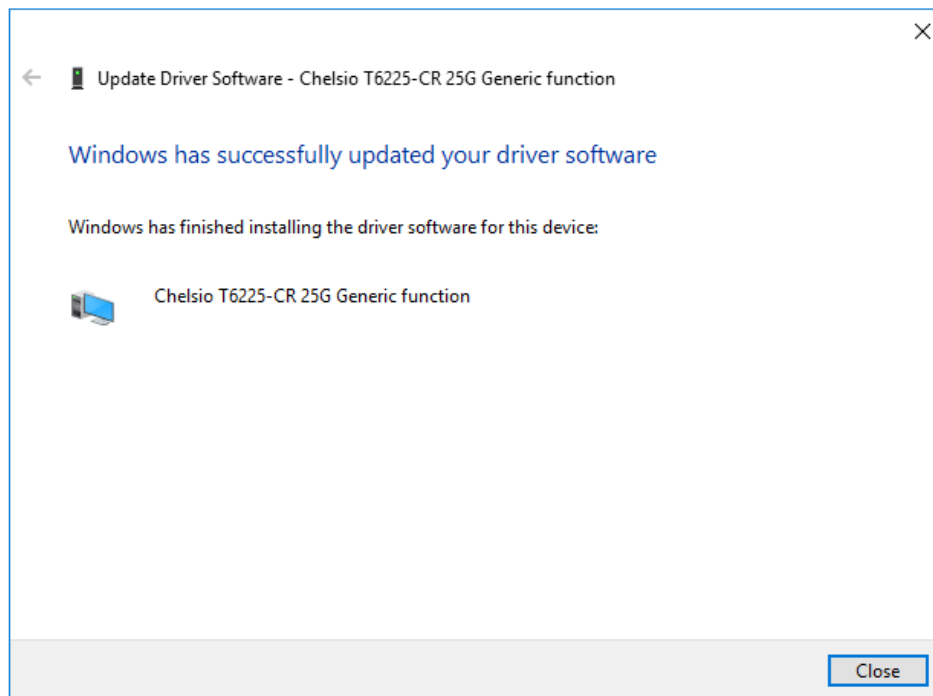
Figure 26 - Driver update successful

5.2. Unified Wire Installer

i.   Run the **ChelsioUwire-x.x.x.xx.exe** installer application.

ii. Click **Next** to start the update.
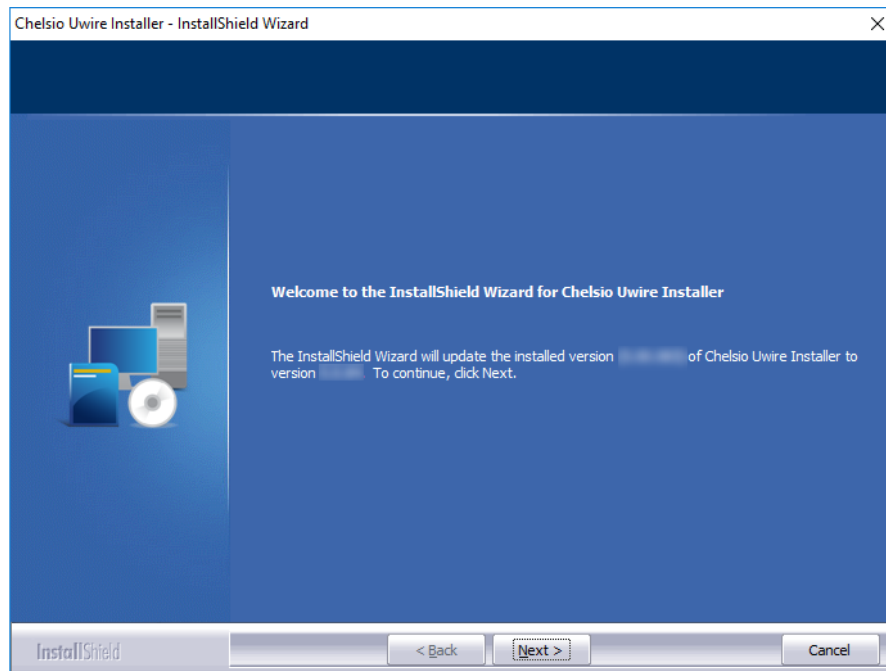
Figure 27 - Unified Wire installer welcome window

iii. Click **Finish** to exit from the Unified Wire Installer.
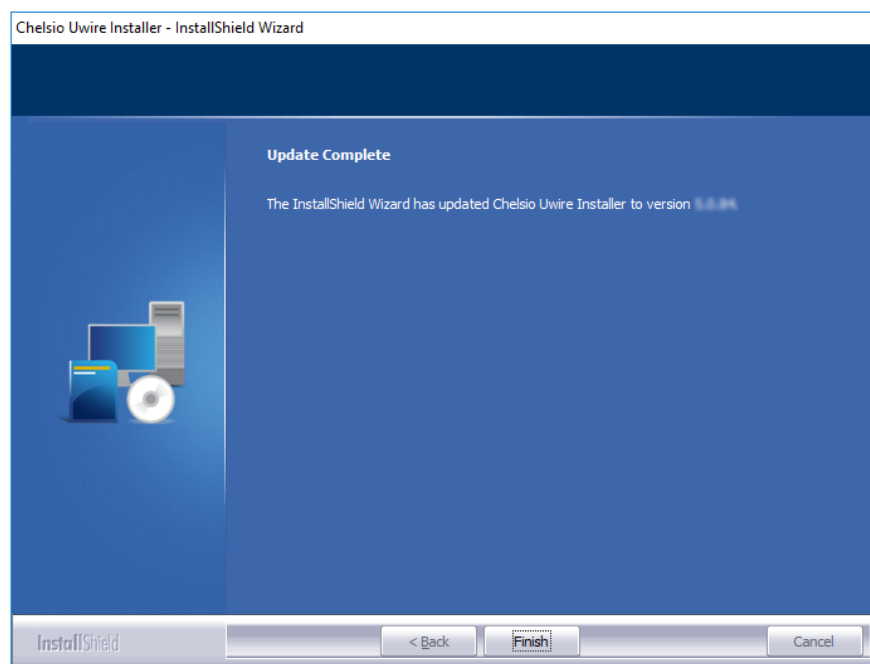
Figure 28 - Finishing Unified Wire update

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 select *Windows GUI* (option 1) as the Windows version. Hit [Enter].

Installer:

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


Figure 29 - Adapter configuration utility (Installer)

Zip Package:

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


Figure 30 - Adapter configuration utility (zip package)

iv. Enter the index of the 40G adapter for which the configuration needs to be updated. Hit [Enter].

Figure 31 - Selecting adapter

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

Figure 32 - Port settings

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

Figure 33 - Operation mode

vii. Open **Device Manager**, click on **System Devices** and right-click on the 40G Chelsio adapter selected in step (iv).

viii. Select **Disable** and then **Enable** for changes to take effect.

6.1.1. Nano Server Machine (Nano Server Installer & Zip Package)

i.   Connect to Nano Server Machine.

ii. In case of Nano Server Installer, the configuration script and cxgbnano.exe utility will be copied to *<system_drive>\Windows\System32* during installation.
If you are using the zip package, copy *cxgbnano.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*.

iii. Run the adapter configuration script and select *Windows Nano Server* (option 2) as the Windows version. Hit [Enter].

Nano Server Installer:

```
[nanomachine-ip]: PS D:\> chelsio_adapter_config.ps1
```
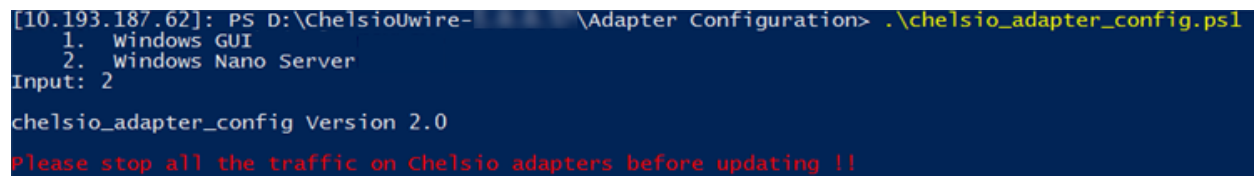
```
[10.193.187.62]: PS D:\>chelsio_adapter_config.ps1
     1.   Windows GUI
     2.   Windows Nano Server
Input: 2

chelsio_adapter_config Version 2.0

Please stop all the traffic on Chelsio adapters before updating !!
```

Figure 34 - Adapter configuration utility (Nano Server Installer)

Zip Package:

```
[nanomachine-ip]: PS D:\ChelsioUwire-x.x.x.xx\Adapter Configuration>
.\chelsio_adapter_config.ps1
```
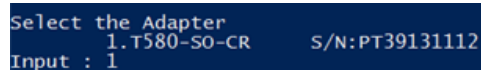
```
[10.193.187.62]: PS D:\ChelsioUwire-         \Adapter Configuration> .\chelsio_adapter_config.ps1
     1.   Windows GUI
     2.   Windows Nano Server
Input: 2

chelsio_adapter_config Version 2.0

Please stop all the traffic on Chelsio adapters before updating !!
```

Figure 35 - Adapter configuration utility (zip package)

iv. Enter the index of the 40G adapter for which the configuration needs to be updated

```
Select the Adapter
     1.T580-SO-CR      S/N:PT39131112
Input : 1
```

Figure 36 - Selecting adapter1

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

Figure 37 - Port settings

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

Figure 38 - Operation mode

vii. Determine the Instance ID of the adapter using the *Get-PnpDevice* command:

Figure 39 - Adapter instance IDs

viii. Use the Instance ID obtained from the previous step to disable and enable the adapter for changes to take effect.

Figure 40 - Committing changes

6.2. Assigning IP address

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. Mass Deployment (Installer)

Using Unified Wire Installer's **Mass Deployment** feature, you can flash or erase Option ROM on multiple systems (nodes) simultaneously. The Installer utilizes **Unified Wire Manager's** Agent and Client components to implement this feature. The Client component must be installed on the host machine and the Agent component on the remote nodes.

7.1. Pre-deployment Configuration

A configuration file containing directives is required by the Installer during installation, flashing and erasing option ROM. Follow the steps mentioned below to generate and configure the file:

i.   Copy the Unified Wire Installer (ChelsioUwire-x.x.x.xx.exe) to a shared location accessible to all the nodes.

ii. Open Windows PowerShell with administrative privileges (*right-click and select Run as Administrator*) and run the following command:

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

The above command will generate a sample configuration file, *RemoteMachineconfig.txt*. It contains the following directives:

```
StartShareMachineInfo
ExecutablePath:\\<Machine-IP>\<Shared directory>\ChelsioUwire-x.x.x.xx.exe
User:<user>
Password:<password>
EndShareMachineInfo

StartOfCluster
<user>:<Password>
<IPAddress1>
<IPAddress2>
<IPAddress3>
<IPAddressN>
EndOfCluster
```

```
StartOfNonCluster

<IPAddress1>:<user1>:<Password1>

<IPAddress2>:<user2>:<Password2>

<IPAddress3>:<user3>:<Password3>

<IPAddressN>:<userN>:<PasswordN>

EndOfNonCluster
```

iii. Open the configuration file and provide the following values:
   a. Enter absolute path of the shared location where installer is copied to, for the *ExecutablePath* parameter.
   b. Provide user credentials for the *User* and *Password* parameters of the machine where Installer was copied to.
   c. Provide IP addresses of remote nodes between the *StartOfCluster* and *EndOfCluster* tags. Enter each node's IP address per line. If same user credentials are set for all the nodes, enter them after the *StartOfCluster* tag, separated by a colon.

   E.g.:

```
StartOfCluster
admin:pass123
10.193.184.63
10.193.184.62
.
.
EndOfCluster
```

   For machines with different user credentials, enter each node's IP address and corresponding user credentials per line between the *StartOfNonCluster* and *EndOfNonCluster* tags in the following format.

   E.g.:

```
10.193.184.76:admin:pass789
10.193.184.78:admin:pass_456
```

iv. Save the file to a desired location.
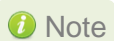
> ⓘ Note   *For successful deployment of UM components and flashing/erasing option ROM, please ensure that the host machine and remote nodes are in the same domain.*

7.2. Flashing Option ROM

> ⚠ **Important** *Unified Wire Installer will flash Option ROM onto the first Chelsio adapter present in remote node (adapter with index 0. See **Configuring using UM CLI** to determine adapter index). Hence, ensure that you either use a non-Chelsio adapter for corporate/private network or ensure that the Chelsio adapter used is not installed as first on remote nodes.*

Option ROM can be flashed using two methods. Both methods involve deployment of UM components.

- **Deploying components and Flashing Option ROM together.**
- **Deploying components and Flashing Option ROM separately.**

> ℹ **Note**
> - *Both methods mentioned above will flash option ROM only onto the first Chelsio adapter present in the remote node. If you plan to flash more adapters in the same node or add new nodes to the cluster, they will have to be added as members to a group and flashed using UM's CLI component. See **Configuring using UM CLI** for instructions.*
> - *Flashing option ROM on remote nodes with inbox NDIS drivers will fail. Hence, please ensure that all nodes are updated to the latest version using the Unified Wire Installer.*

7.2.1. Deploying UM components and Flashing Option ROM together

i.   Run the following command to invoke the UM installer.

```
PS C:\Users\Administrator> .\ChelsioUwire-x.x.x.xx.exe -action
flashoptionrom -config <config_file.txt>
```

ii. Click the **Next** button for the Chelsio End User License Agreement Window.
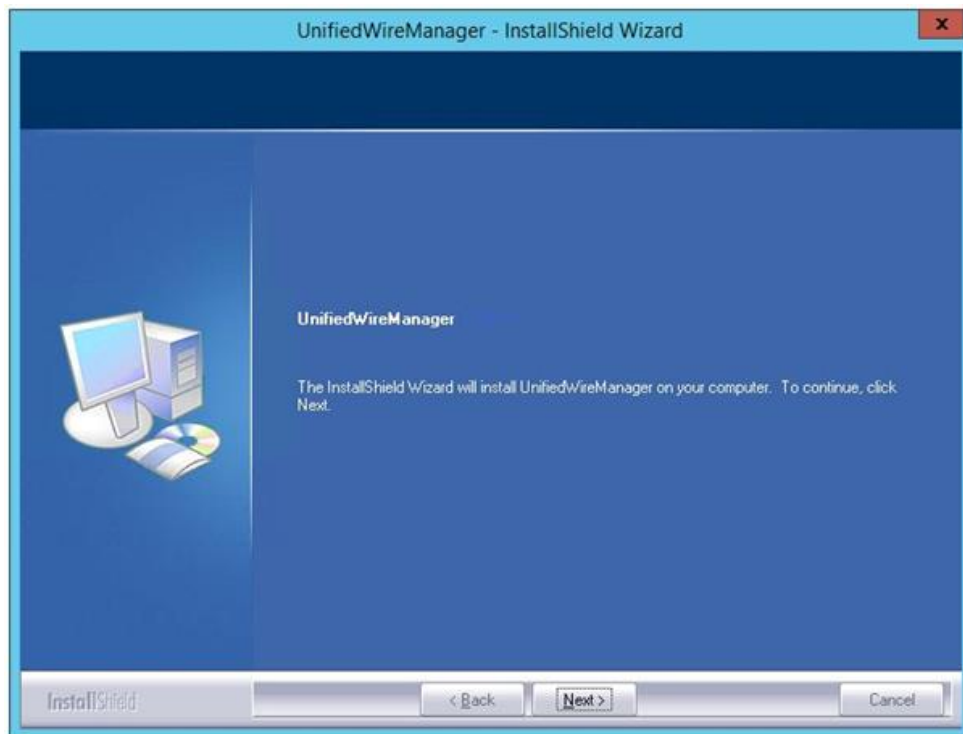
Figure 41 - UM Installer welcome window

iii. Select the radio button **I accept the terms of the license agreement** and click **Next**.
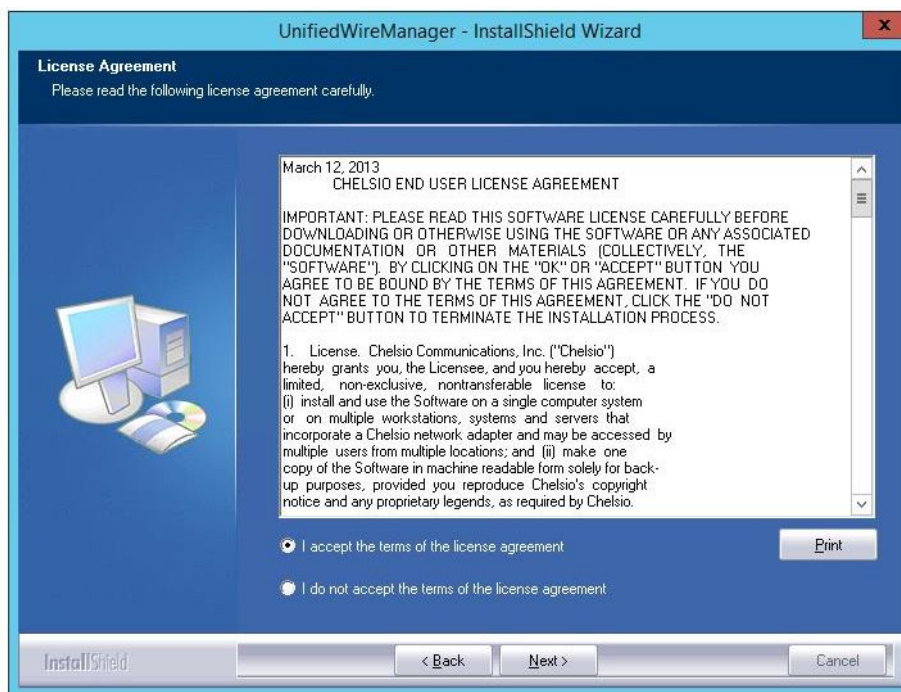
Figure 42 - Chelsio EULA window

iv. The next window will display the pre-requisites for various UM components. Ensure that they are met before proceeding. Click **Next**.
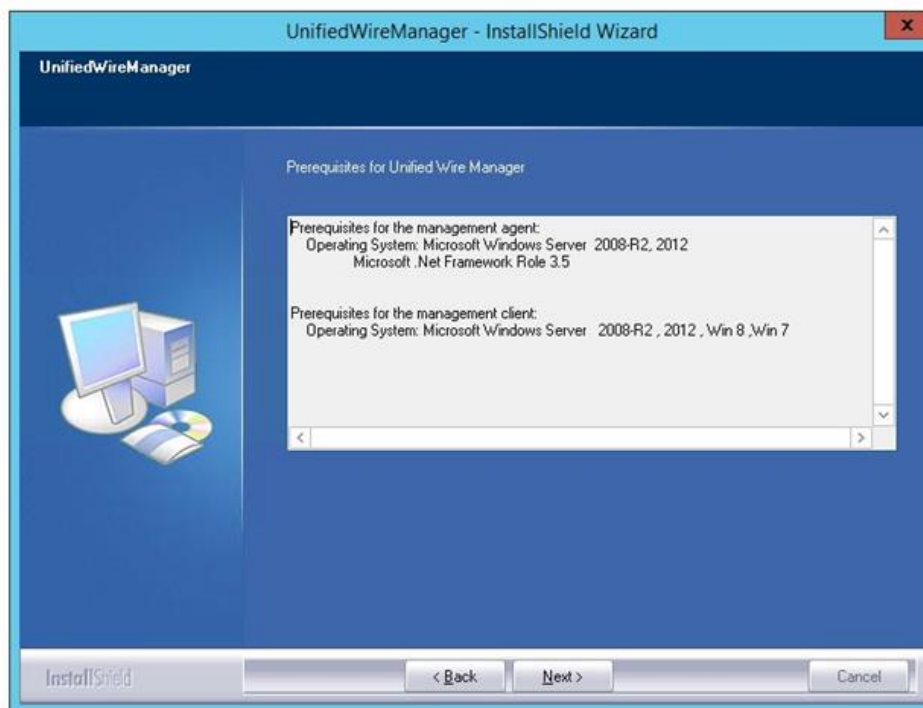
Figure 43 - UM prerequisites window

v. Now, select **Custom** radio button to customize the installation. Click **Next**.
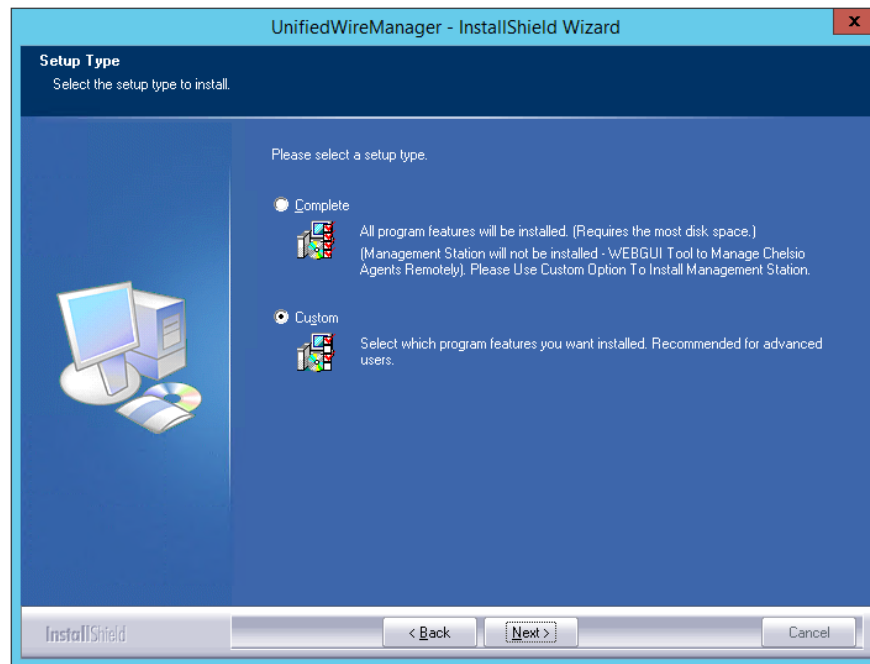
Figure 44 - UM prerequisites window

vi. The next window will display the location where UM will be installed by default. You can change the location by using the **Change** button or click **Next** to continue with the default path.
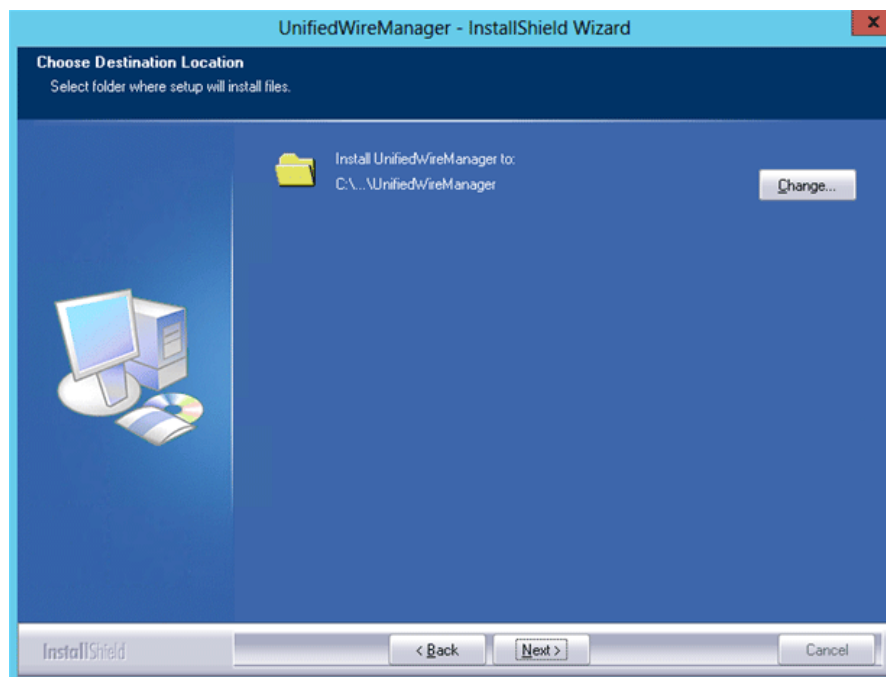
Figure 45 - Changing UM installation path

vii. Next, ensure that only **Client** is selected under features to be installed. Click **Next**.
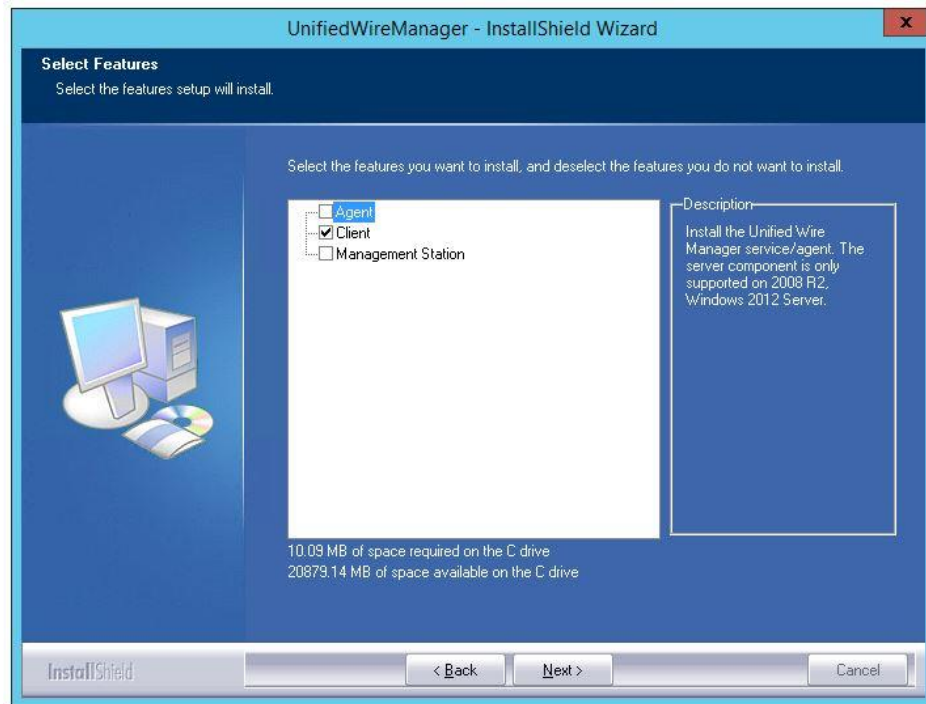
Figure 46 - Selecting UM components

viii. Click **Finish** to complete Client installation.
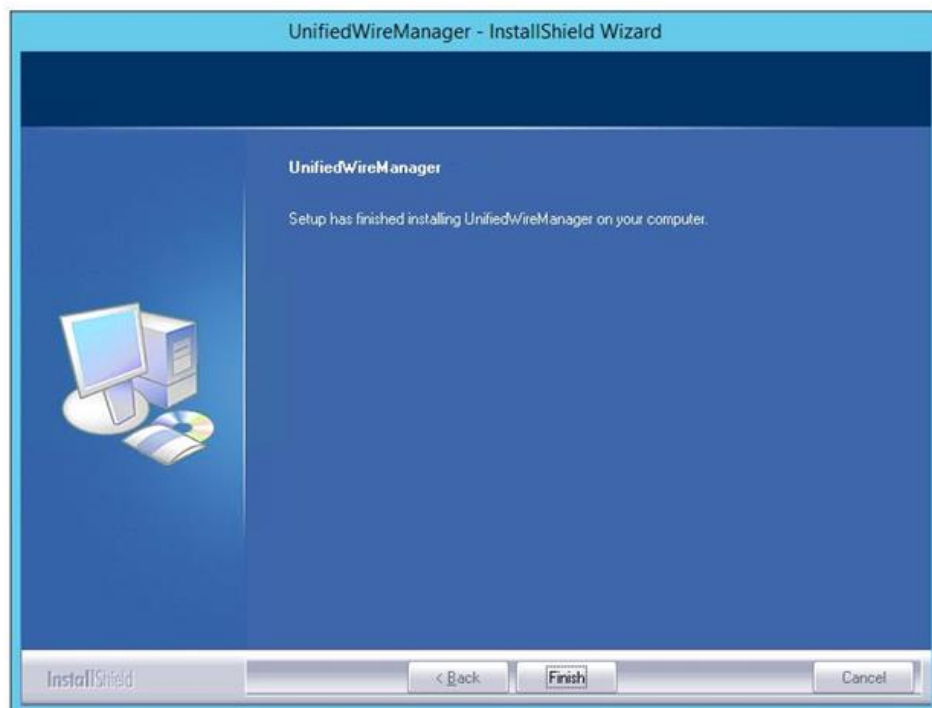
Figure 47 - Finishing UM installation

ix. Now, installation of UM Agent component on remote nodes will commence.
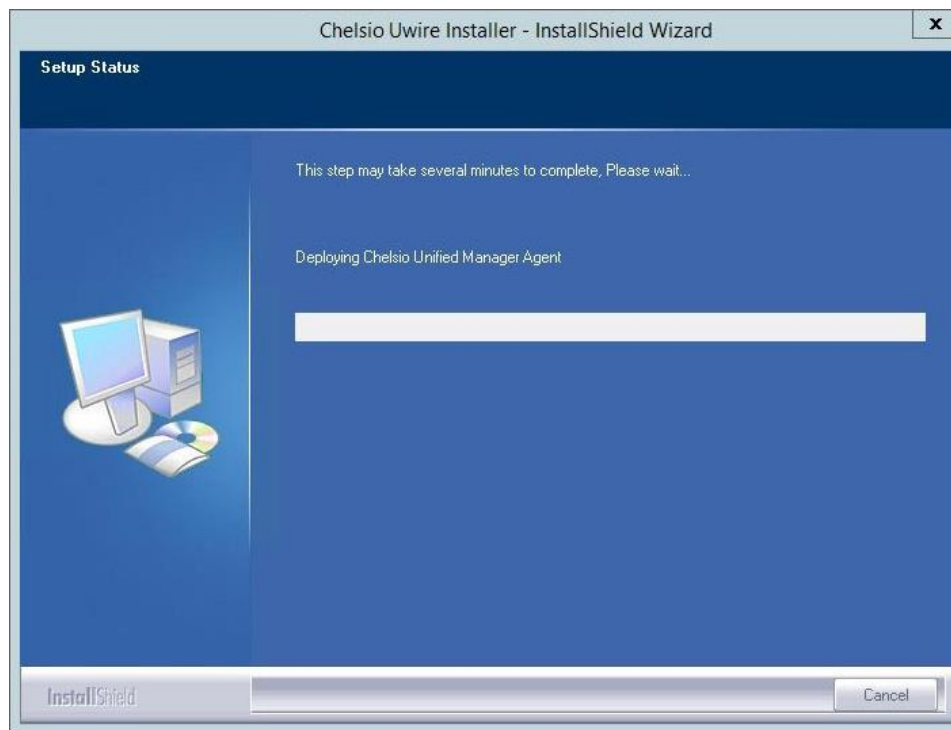
Figure 48 - Starting Agent installation

x. Click **Agree** on the **PsExec License Agreement** window that appears. This window will appear only during first installation.
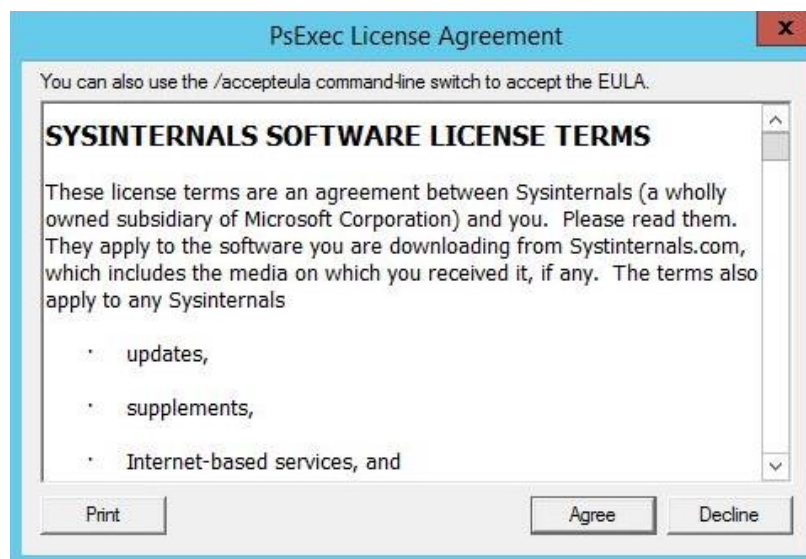
Figure 49 - PsEXec license agreement

xi. UM Agents will now be installed on remote nodes. A default group **OptionROM** will be created and all the nodes on which UM Agent was successfully installed, will be added to it as *members*.

Figure 50 - Agent installation on remote nodes

⚠ **Important** *Please ensure that the **OptionROM** group created here is not deleted. Also, UM Client component should not be un-installed. Doing so will cause flashing/erasing Option ROM to fail on remote nodes. UM components will have to be deployed again to fix this issue.*

xii. Option ROM will now be flashed onto adapters on remote nodes present in the OptionROM group.

Figure 51 - Flashing option ROM

xiii. After completion, a log file *MassDeploy.log* containing the summary of the process will be created. Click **OK** on the dialog box that appears to exit the installer and view the log.
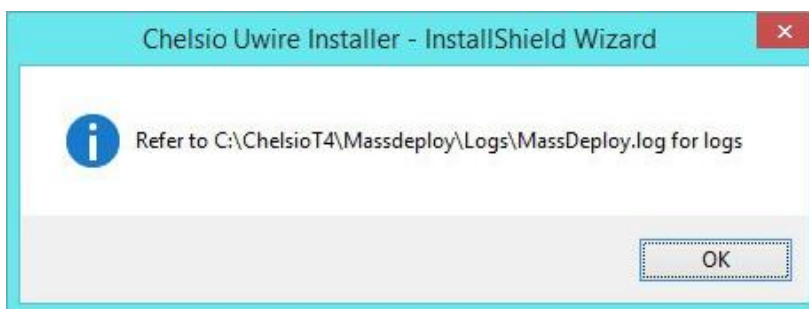
Figure 52 - Finishing option ROM flashing

7.2.2. Deploying UM components and Flashing Option ROM separately

● Deploying UM Components

i. Run the following command to invoke the Unified Wire Installer.

```
PS C:\Users\Administrator> .\ChelsioUwire-x.x.x.xx.exe -action deployagent
-config <config_file.txt>
```

ii. Follow steps **(ii)**-(**xi**) in the **Deploying UM components and Flashing Option ROM together** section.

iii. After completion, a log file *MassDeploy.log* containing the summary of the process will be created. Click **OK** on the dialog box that appears to exit the installer and view the log.
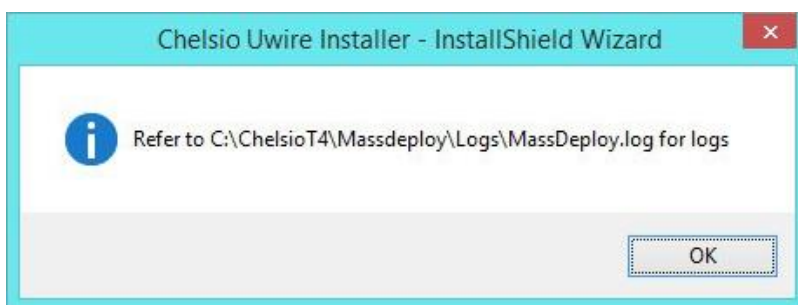
Figure 53 - Finishing UM components deployment

● Flashing Option ROM

i. Run the following command to flash Option ROM:

```
PS C:\Users\Administrator> .\ChelsioUwire-x.x.x.xx.exe –action
flashoptionrom
```

ii. Flashing Option ROM is possible only if UM Agents were installed on nodes using this installer. Click **Yes** to continue.

Figure 54 - Starting option ROM flashing

iii. Follow steps **xii** and **xiii** in the **Deploying UM components and Flashing Option ROM together** section.

> ⓘ Note *Flashing option ROM on remote nodes with inbox NDIS drivers will fail. Hence, please ensure that all nodes are updated to the latest version using the Unified Wire Installer.*

7.3. Erasing Option ROM
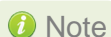
> ⓘ Note *Unified Wire Installer will erase option ROM only from the first Chelsio adapter present in the remote node. If you plan to erase from more adapters in the same node or add new nodes to the cluster, they will have to be added as members to a group and erased using UM's CLI component. See **Configuring using UM CLI** for instructions.*

i. To erase OptionROM from adapters on all the nodes, run the following command:

```
PS C:\Users\Administrator> .\ChelsioUwire-x.x.x.xx.exe -action
eraseoptionrom
```

ii. Erasing OptionROM is possible only if the adapters were flashed using this installer. Click **Yes** to continue.
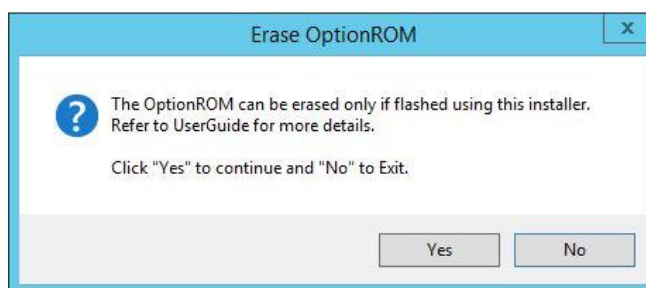
Figure 55 - Starting option ROM erasing

iii. Option ROM will now be erased from adapters on nodes present in the **OptionROM** group.

Figure 56 - Erasing option ROM on remote nodes

iv. After completion, a log file *MassDeploy.log* containing the summary of the process will be created. Click **OK** on the dialog box that appears to exit the installer and view the log.
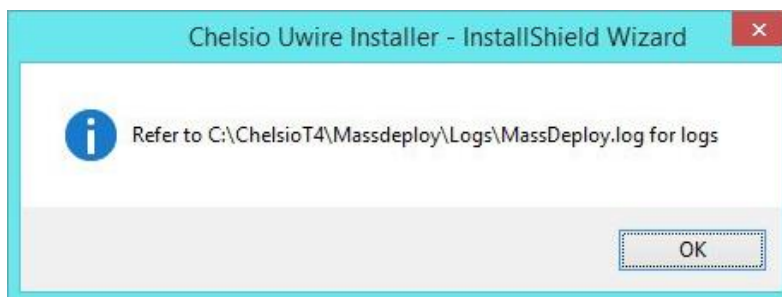
Figure 57 - Finishing option ROM erasing

7.4. Configuring using UM CLI

7.4.1. Configuring groups and members

You can manage multiple groups and members using Unified Wire Manager's CLI component, **chelsio_uwcli.**

> ⓘ Note   *To use the CLI component, you will have to change your working directory to the location where UM Client is installed. Typically, this will be **C:\Program Files (x86)\Chelsio Communications\UnifiedWireManager\client**. To avoid this and run the command from any path, you will need to reboot the host. The following examples assume that the host machine was rebooted before running commands.*

⦿ **Create Group**

To create a new group, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe –m creategroup
grpname=<new_group_name> OSType=windows GroupType=t4adapter
```

E.g.

```
PS C:\Users\Administrator> chelsio_uwcli.exe –m creategroup grpname=flash
OSType=windows GroupType=t4adapter
```

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m creategroup grpname=flash OSType=windows GroupType=t4adapter
Machine Group created successfully
```

Figure 58 - Creating group

- ### Add member to group

To add a member to a group, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m addmember
grpname=<group_name>
details="<IP_address>;<user_id>;<password>;*;*,*,<adapter_index>,*,*,*,*,*>"
```

E.g.

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m addmember grpname=OptionROM
details="10.193.185.107;administrator;cdrom888;*;*,*,0,*,*,*,*,*>"
```

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m addmember grpname=OptionROM details="10.193.185.107;administrator;cdrom@888;*;*;*;0;*;*;*;*;*"
Machine added to group successfully
```

Figure 59 - Adding member to group

- ### View group and member details

To view details of all the groups and members created, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m list
```

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m list
Machine Group Details
--------------------
Group Name       : OptionROM
OSType           : windows
Group Type       : t4adapter

        Member details
        ------------------------------------
        index                : 0
        hostname/IP address : 10.193.184.62
        user                 : administrator
        password             : cdrom@888
        pci id               : *
        ifname               : *
        port instance        : *
        adapter no           : 0
        comment              : *
        param1               : *
        param2               : *
        param3               : *
        param4               : *

        Member details
        ------------------------------------
        index                : 1
        hostname/IP address : 10.193.184.78
        user                 : administrator
        password             : cdrom@888
        pci id               : *
        ifname               : *
        port instance        : *
        adapter no           : 0
        comment              : *
        param1               : *
        param2               : *
        param3               : *
        param4               : *
```

Figure 60 - Viewing group and member details

● Delete member from a group

To delete a member from a group, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m deletemember
grpname=<group_name> index=<member_index>
```

E.g.

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m deletemember
grpname=OptionROM index=1
```

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m deletemember grpname=OptionROM index=1
System removed from Machine Group Successfully
PS C:\Users\Administrator>
```

Figure 61 - Deleting member from group

● Delete Group

To delete a group, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m deletegroup
grpname=<group_name>
```

E.g.

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m deletegroup
grpname=OptionROM
```

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m deletegroup grpname=OptionROM
Machine Group deleted successfully
```

Figure 62 - Deleting group

7.4.2. Flashing/Erasing Option ROM

Apart from Unified Wire Installer, you can also use Unified Wire Manager's CLI component (*chelsio_uwcli*) to flash or erase Option ROM on multiple nodes. This is particularly useful when you need to flash/erase Option ROM on groups other than the default **OptionROM** group.

Flashing Option ROM

To flash Option ROM using UM, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe chelsio flashrom
adapter=BULKADAPTER path=<path_to_optionrom_image_file> -B <group_name>
```

E.g.

```
PS C:\Users\Administrator> chelsio_uwcli.exe chelsio flashrom
adapter=BULKADAPTER path=C:\Users\Administrator\Desktop\cuwlbt4.bin -B
OptionROM
```


Figure 63 - Flashing option ROM using UM CLI

> ⓘ Note   *Flashing option ROM on remote nodes with inbox NDIS drivers will fail. Hence, please ensure that all nodes are updated to the latest version using the Unified Wire Installer.*

Viewing status

You can list the adapters on local host and verify if Option ROM was successfully flashed using the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe chelsio listadapters -B
<group_name>
```

E.g.

```
PS C:\Users\Administrator> chelsio_uwcli.exe chelsio listadapters -B
OptionROM
```

```
PS C:\Users\Administrator>chelsio_uwcli.exe chelsio listadapters -B OptionROM
Executing...
CLI version 2.4.59
**********10.193.184.62***********
Agent Version 2.4.59

Adapter information:

Adapter #                     : 0
Model                         : T440-LP-CR
Serial Number                 : NB15110005
Connector                     : 10G FIBER_XFI
PHY                           : No Phy / No information Available
PCI Vendor ID:Device ID       : 1425:01b8
PCI Location                  : 08:00:04
Factory MAC address           : 00:07:43:04:75:01
OPROM                         : Present
Bios Version                  : 1.0.4.57

**********10.193.184.78***********
Agent Version 2.4.59

Adapter information:

Adapter #                     : 0
Model                         : T404-BT
Serial Number                 : PT12110700
Connector                     : 1G/100M BT_SGMII/RJ-45
PHY                           : VCS8634
PCI Vendor ID:Device ID       : 1425:01b8
PCI Location                  : 02:00:04
Factory MAC address           : 00:07:43:10:36:00
OPROM                         : Present
Bios Version                  : 1.0.4.57
```

Figure 64 - Viewing status

● Erasing Option ROM

To erase Option ROM using UM, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe chelsio eraserom
adapter=BULKADAPTER force=1 –B <group_name>
```

In addition to flashing and erasing Option ROM, you can perform additional bulk operations on remote nodes like setting MTU and VLAN ID, changing adapter and port parameters, etc. To know more about these options available, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe -
```

```
PS C:\Users\Administrator> chelsio_uwcli.exe chelsio eraserom adapter=BULKADAPTER force=1 -B OptionROM
Executing...
CLI version 2.4.59
**********10.193.150.73***********
Agent Version 2.4.59

Command completed successfully.
```

Figure 65 - Erasing option ROM using UM CLI

7.4.3. Help

To view Unified Wire Manager's CLI help, run the following command:

```
PS C:\Users\Administrator> chelsio_uwcli.exe -m
```

8. *cxgbtool/cxgbnano* help

The *cxgbtool* (*cxgbnano* for Nano Server) 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]

In case of Nano Server, run the commands using **cxgbnano** instead of **cxgbtool,** i.e.,

cxgbnano [*vbdIface|nicIface|iSCSIIface|rdma*] [*category*] [*commands*] [*options*]

Definitions

[*nicIface*] is the name of the network device to work on, given in the format *nic[0]*, *nic[1]*.

[*vbdIface*] is the name of the Chelsio Bus Enumerator instance to work on, given in the format *vbd[0]*, *vbd[1]*

[*iscsIface*] is the name of the Chelsio iSCSI interface to work on, given in the format *iSCSI[0]*, *iSCSI[1]*

[*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**
    - o Displaying cxgbtool help: `cxgbtool`
    - o Displaying list of available cxgbtool commands: `cxgbtool -h all`
    - o Displaying category specific help:

    `cxgbtool [vbdIface|nicIface|iSCSIIface] [category] -h`

    - o Displaying list of cudbg entities: `cxgbtool [vbdIface] cudbg dbg -h`
    - o 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 [bootImg]`
`[pf{val}|offset{val}|clear]`

Parameters:

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

offset {val}: Flash Option ROM image to the offset specified by the user.

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

Examples:

  o 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
```

  o 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]`

Parameters:

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

Examples:

o   Loading boot configuration file:

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

o   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:

o   Read SEEPROM data to a bin file: `cxgbtool [vbdIface] boot seeprom read`
o   Display SEEPROM data onscreen: `cxgbtool [vbdIface] boot seeprom dump`
o   Write SEEPROM data: `cxgbtool [vbdIface] boot seeprom write [file]`
o   Verify SEEPROM data: `cxgbtool [vbdIface] boot seeprom verify [file]`

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

Examples:

o   Reading SEEPROM data:

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

o   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
```

o 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
```

o 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:

o Read serialinit data to a bin file: `cxgbtool [vbdIface] boot serialinit read`
o Display serialinit data onscreen: `cxgbtool [vbdIface] boot serialinit dump`
o Write serialinit data: `cxgbtool [vbdIface] boot serialinit write [file]`
o Verify serialinit data: `cxgbtool [vbdIface] boot serialinit verify [file]`

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

Examples:

o Reading serialinit data

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

o   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
```

o   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
```

o   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:

o   Read VPD to a bin file: `cxgbtool [vbdIface] boot vpd read`
o   Display VPD onscreen: `cxgbtool [vbdIface] boot vpd dump`

- o Write VPD: `cxgbtool [vbdIface] boot vpd write [file]`
- o Verify VPD: `cxgbtool [vbdIface] boot vpd verify [file]`

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

Examples:

- o Reading VPD:

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

- o 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
```

- o 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
```

o 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 [entity] [outputFile][options]`

Examples:

o 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
```

o 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 [entity] [outputfile][options]`

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 [entity] [inputFile][options]`

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:

o 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:            0x800000-0x407ffff [1.00 GiB]
RQUDP region:  0xffffffff-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]
```

o 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:

- o 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
```

- o 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
```

o 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
```

o   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
```

o   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
```

o   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]`

> ℹ **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:

- **tidinfo**

Description: Collect/Display TID information

Syntax:

```
cxgbtool [vbdIface] cudbg [collect|view|extract|dbg] tidinfo [file] [dir]
```

Example:

▪ **cctrl**

Description: Collect/Display Congestion control table.

Syntax:

```
cxgbtool [vbdIface] cudbg [collect|view|extract|dbg] cctrl [file] [dir]
```

Example:

```
C:\Users\Administrator>cxgbtool vbd0 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
00000000 00000000
   3375      12225643     0      1  01000010 00560004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
00000000 00000000
   3376      12225643     0      0  01c00010 00000004  0000003b 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
00000000 00000000
   3377      12225643     0      1  01000010 00560004  0000003b 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
00000000 00000000
   3378      12225643     0      0  01c00010 00000004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
00000000 00000000
   3379      12225643     0      1  01000010 00560004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
00000000 00000000
   3380      12225643     0      0  01c00010 00000004  0000003b 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
00000000 00000000
   3381      12225643     0      1  01000010 005a0004  0000003b 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
00000000 00000000
   3382      12225643     0      0  01c00010 00000004  0000003a 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000 00000000  0
00000000 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:   0xffffffff-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:

- **ibqsge0, ibqsge1**

Description: Collect/Display CIM SGE inbound queue.

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

Example:

▪ **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:

▪ **pmindirect**

Description: Collect/Display PM indirect registers information.

Syntax:
```
cxgbtool [vbdIface] cudbg [collect|view|extract|dbg] pmindirect [file] [dir]
```

Example:

▪ **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    -         0    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    -         0    0   Y   0x1  4 65                                                                             0  0  0  0  0
  2 01:00:5e:00:00:01 ffffffffffff    -    -      -    N    -         0    0   Y   0x3  4 65 00000000 00000000 00000000 00000000 00000006 00000000 00000000 00000000   0  0  0  0  0
  3 33:33:00:00:00:01 ffffffffffff    -    -      -    N    -         0    0   Y   0x3  4 65 00000000 00000000 00000000 00000000 00000006 00000000 00000000 00000000   0  0  0  0  0
  4 33:33:ff:f5:69:e5 ffffffffffff    -    -      -    N    -         0    0   Y   0x1  4 65                                                                             0  0  0  0  0
  5 00:07:43:04:b3:98 ffffffffffff    -    -      -    N    -         0    0   Y   0x2  4 66                                                                             0  0  0  0  0
  6 33:33:ff:fd:e6:68 ffffffffffff    -    -      -    N    -         0    0   Y   0x2  4 66                                                                             0  0  0  0  0
  7 33:33:00:00:00:fb ffffffffffff    -    -      -    N    -         0    0   Y   0x3  4 66 00000000 00000000 00000000 00000000 00000006 00000000 00000000 00000000   0  0  0  0  0
  8 33:33:00:01:00:03 ffffffffffff    -    -      -    N    -         0    0   Y   0x3  4 66 00000000 00000000 00000000 00000000 00000006 00000000 00000000 00000000   0  0  0  0  0
  9 01:00:5e:00:00:fb ffffffffffff    -    -      -    N    -         0    0   Y   0x3  4 66 00000000 00000000 00000000 00000000 00000006 00000000 00000000 00000000   0  0  0  0  0
 10 01:00:5e:00:00:fc ffffffffffff    -    -      -    N    -         0    0   Y   0x3  4 66 00000000 00000000 00000000 00000000 00000006 00000000 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
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
```

- **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:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg ibqncsi
cudbg_view() dbg entity : ibqncsi
000000: 00000000 6084f1a4 00000000 02ac5032
0x0010: 00000000 5aee8fdb 00000000 650a9bc8
0x0020: 00000000 78ca245f 00000000 f86aa0e1
0x0030: 00000000 1d3edbfe 00000000 fb6c063b
0x0040: 00000000 89fa5845 00000000 100a5ae2
0x0050: 00000000 644d432a 00000000 349fed29
0x0060: 00000000 ff6c5f83 00000000 0e64d7e3
0x0070: 00000000 6194b426 00000000 73e787d9
0x0080: 00000000 18a9ae41 00000000 5b4ac20f
0x0090: 00000000 d4b1cae5 00000000 16b2f989
0x00a0: 00000000 4a0b2d2a 00000000 f760f8de
0x00b0: 00000000 aa57c5de 00000000 9ec473f7
0x00c0: 00000000 b51ea495 00000000 6859293c
0x00d0: 00000000 ec6d2d7b 00000000 79b6de6d
0x00e0: 00000000 46f36574 00000000 17690232
0x00f0: 00000000 c9a1cad2 00000000 72f65ca9
0x0100: 00000000 11586e13 00000000 89d4b998
```

▪ **obqncsi**

Description: Collect/Display NCSI outbound queue.

Syntax:
cxgbtool [vbdIface] cudbg [collect|view|extract|dbg] obqncsi [file] [dir]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cudbg dbg obqncsi
cudbg_view() dbg entity : obqncsi
000000: 00000001 00000001 00010200 000000ff
0x0010: 00000001 00000001 00010200 000000ff
0x0020: 00000001 00000001 00010200 000000ff
0x0030: 00000001 00000001 00010200 000000ff
0x0040: 00000001 00000001 00010200 000000ff
0x0050: 00000001 00000001 00010200 000000ff
0x0060: 00000001 00000001 00010200 000000ff
0x0070: 00000001 00000001 00010200 000000ff
0x0080: 00000001 00000001 00010200 000000ff
0x0090: 00000001 00000001 00010200 000000ff
0x00a0: 00000001 00000001 00010200 000000ff
0x00b0: 00000001 00000001 00010200 000000ff
0x00c0: 00000001 00000001 00010200 000000ff
0x00d0: 00000001 00000001 00010200 000000ff
0x00e0: 00000001 00000001 00010200 000000ff
```

▪ **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    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
0000000000000000     0    00000000    000000000000000000000000000000000
```

▪ **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: 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

- ## 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)]]`

Parameters:

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 the 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 [rx-usecs-irq {val}] [rx-frames-irq {val}]
[(persistent)]
```

Parameters

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:

- o Setting Rx Coalescing Timer:

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

- o 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:

▪ **rdma_stats**

Description: Print RDMA statistics if NDK/ND is enabled

Syntax: `cxgbtool [nicIface] debug rdma_stats`

Example:

- **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 [filePath]`

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 [filePath]`

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 [nic|vbd] [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]
```

i Note   *Write MDIO register currently not supported.*

  - **sensor**

Description: Display sensor data.

Syntax: `cxgbtool [nicIface] hardware sensor`

Example:

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

- **reg**

Description: Read/Write register.

Syntax:

- o Register Read:

```
cxgbtool [vbdIface|nicIface|iscsiIface] hardware reg [readAddr]
```

- o Register Write:

```
cxgbtool [vbdIface|nicIface|iscsiIface] hardware reg [[writeAddr] = {val}]
```

Example:

- o Register read:

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

- o 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
SGE_DEBUG_DATA_LOW13                             0          0
SGE_DEBUG_DATA_LOW14                             0x1534745   22234949
SGE_DEBUG_DATA_LOW15                             0xfeacb8ba  4272732346
```

- **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[traceQueueId]`

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: 00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000/00000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000@0
```

- **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
```

- **RDMA**

Description: Display RDMA information.

Syntax:

```
cxgbtool rdma -d [iWARPAdapterIndex] [options] | -l
```

Examples:

o   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
```

o   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
```

o   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
```

o 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
```

o Displaying list of iWARP adapters:

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

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

o 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)
```

9. 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.

9.1. 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.

9.1.1. 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**.
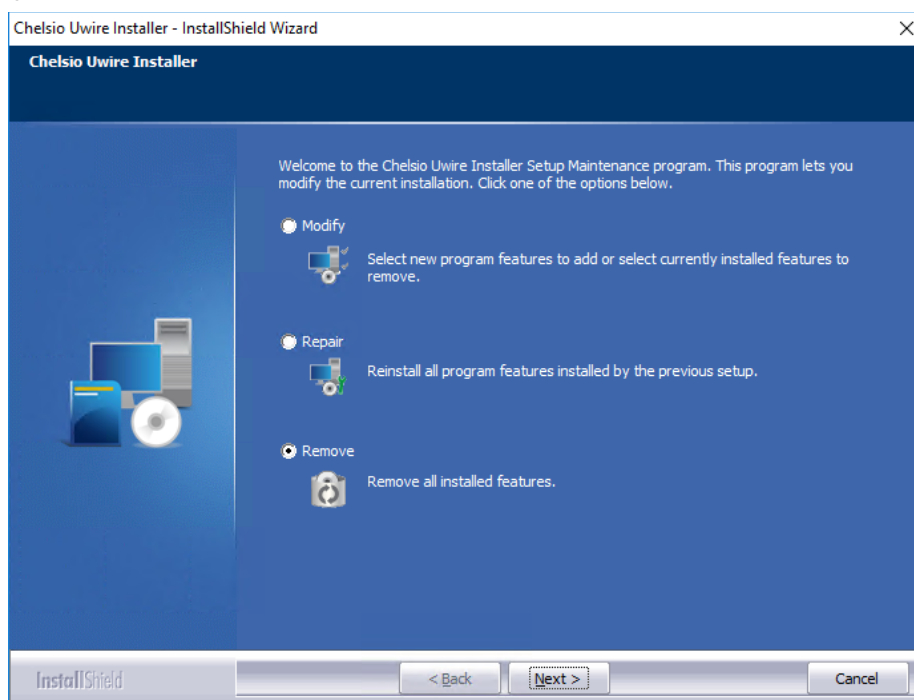
Figure 66 - Selecting maintenance option

iii. Click on the **Finish** button to exit from the installer.
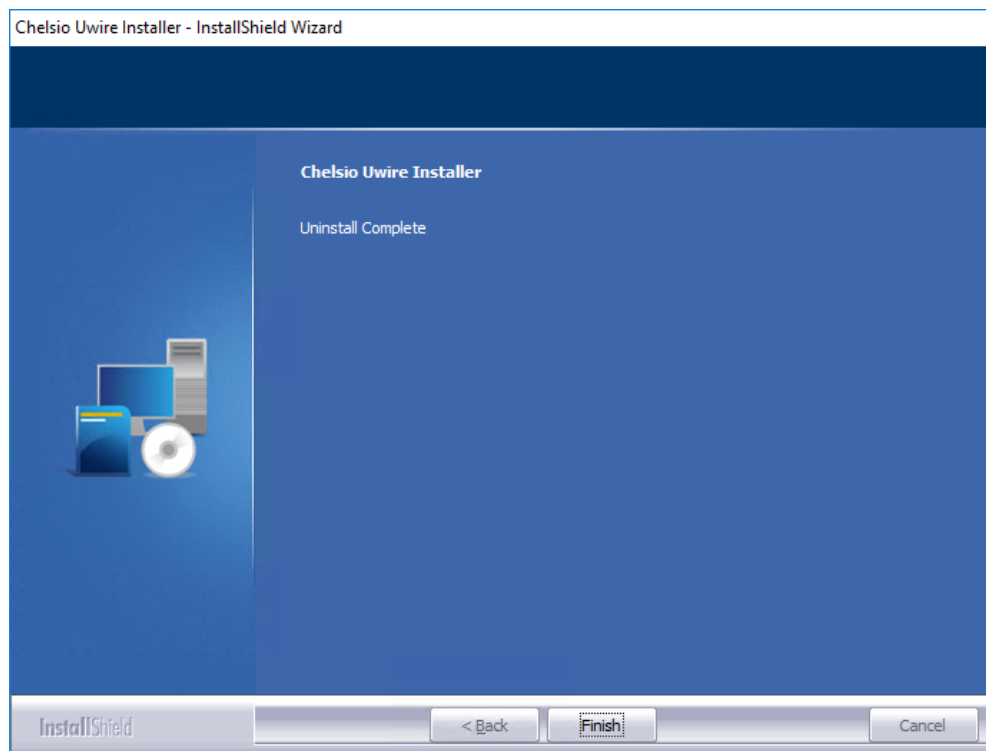
Figure 67 - Finishing uninstallation

🛈 Note
- *This method of uninstallation is possible only if the drivers were installed using Unified Wire Installer.*
- *Uninstalling Unified Wire package using the above method will not uninstall Unified Wire Manager. See* Uninstallation *section of* **Unified Wire Manager** *chapter for more information.*

9.1.2. 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)>
```

E.g.:

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

The above command will uninstall iSCSI Storport Miniport driver.

🛈 Note *Uninstalling Unified Wire package using the above method will not uninstall Unified Wire Manager. You will have to manually uninstall it using "Programs and Features" in the Control Panel.*

9.2. 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.**

iii. 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.**

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

- **Virtual Bus Driver**

iv. 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].***

v. 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.**

iii. 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:

Windows Version	NDIS Components
Server 2016	NIC, NVGRE, VXLAN, PacketDirect, VMMQ/vRSS
Nano Server 2016	NIC, 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*: default

- **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

- o 160
- o 192
- o 224
- o 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*:
    - o Auto Negotiation
    - o Disabled
    - o Rx & Tx Enabled
    - o Rx Enabled
    - o 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*:
    - o Extreme
    - o High
    - o Low
    - o Minimal
    - o Moderate
  - *Default*: Low

- ● **IPv4 Checksum Offload**
  - *Description*: Allow the checksum to be computed by the adapter for IPV4 packets.
  - *Value*:
    - o Disabled
    - o Rx & Tx Enabled
    - o Rx Enabled
    - o Tx Enabled
  - *Default*: Rx & Tx Enabled

- **Jumbo Packet (Maximum Transmission Unit)**
  - *Description*: Specify the Maximum Transmission Unit (MTU) value.
  - *Value*:
    - o 4088 Bytes
    - o 9014 Bytes
    - o 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*: Enabled

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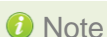
ⓘ Note   *This feature is not fully tested and available for experimental usage only.*

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

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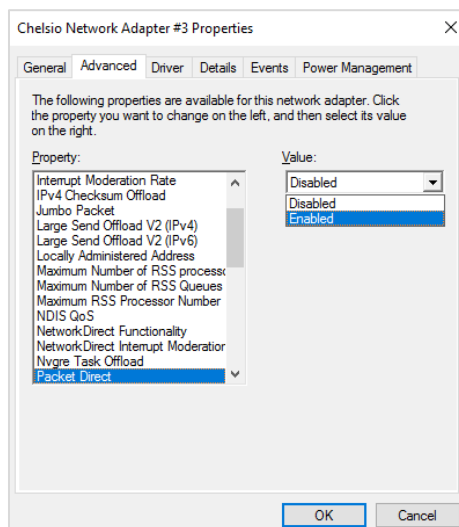
iii. Create a new virtual switch and enable PacketDirect on the host:

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


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>
```

```
PS C:\Users\Administrator> Start-VM VM1
```

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 needs to be enabled for each Virtual Machine using the below command on the host:

```
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 VM -VrssEnabled:$true -VmmqEnabled:$true -VmmqQueuePairs 8
```

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
- 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
Nano Server 2016	SMB Direct, Mode 2
10 AU Client	SMB Direct
Server 2012 R2	SMB Direct

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:

Figure 68 - 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>
```


Figure 69 - 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.

Figure 70 - 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 71 - 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 72 - 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. 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. 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
Nano Server 2016	NDIS SR-IOV
Server 2012 R2	

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

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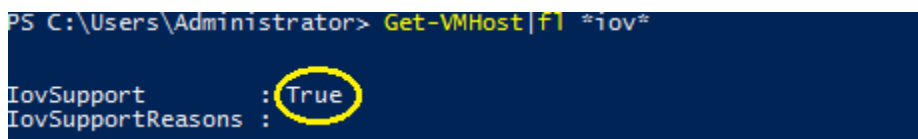
In case of Installer, the script will be copied to *<system_drive>\Windows\System32* during installation.

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 73 - Checking SR-IOV capability

iii. Run the adapter configuration script and select *Windows GUI* (option 1) as the Windows version. Hit [Enter].

   Installer:

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

Figure 74 - Adapter configuration utility (Installer)

Zip Package:

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


Figure 75 - Adapter configuration utility (Zip Package)

iv. Enter the index of the adapter for which the configuration needs to be updated. Hit [Enter].

Figure 76 - Selecting adapter

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

Figure 77 - Setting the configuration type

vi. Reboot system for changes to take effect.

2.1.2. Nano Server (Nano Server Installer & Zip Package)

i.   Connect to Nano Server Machine.
ii. Check if the machine is SR-IOV capable using the following command:

```
[nanomachine-ip]: PS C:\> Get-VMHost | fl *iov*
```

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

Figure 78 - Checking SR-IOV capability

iii. If Unified wire was installed on the Nano Server machine using the Nano Server installer, skip to step (vii). If the driver components were manually installed using the boot image (Zip package), follow all the steps mentioned below.
iv. Copy ChelsioUwire-x.x.x.xx folder to a desired location.
v.   Copy *cxgbnano.exe* from *ChelsioUwire-x.x.x.xx* folder to *<system_drive>\Windows\System32*
vi. Change your working directory to *ChelsioUwire-x.x.x.xx\Adapter Configuration*.
vii. Run the adapter configuration script and select *Windows Nano Server (option 2)* as the Windows version. Hit [Enter].

Nano Server Installer

```
[nanomachine-ip]: PS D:\> chelsio_adapter_config.ps1
```


Zip Package

```
[nanomachine-ip]: PS D:\ChelsioUwire-x.x.x.xx\Adapter Configuration>
.\chelsio_adapter_config.ps1
```

```
[10.193.187.62]: PS D:\ChelsioUwire-5.0.0.57\Adapter Configuration> .\chelsio_adapter_config.ps1
     1.  Windows GUI
     2.  Windows Nano Server
Input: 2

chelsio_adapter_config Version 2.0

Please stop all the traffic on Chelsio adapters before updating !!
```

Figure 79 - Adapter configuration utility

viii. Enter the index of the Chelsio adapter for which the configuration needs to be updated. Hit [Enter].

```
Select the Adapter
       1.T6225-CR        S/N:RE41160011
Input : 1
```

Figure 80 - Selecting adapter

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

```
Choose the configuration type:
        1. NON-SRIOV (Default)
        2. SRIOV
        3. Port settings
Input : 2
Do you want to continue (y/n): y
Successfully updated the selected configuration type.

Verification: Passed

Please reboot the machine for changes to take effect.
```

Figure 81 - Setting the configuration type

x.   Reboot the Nano Server 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.
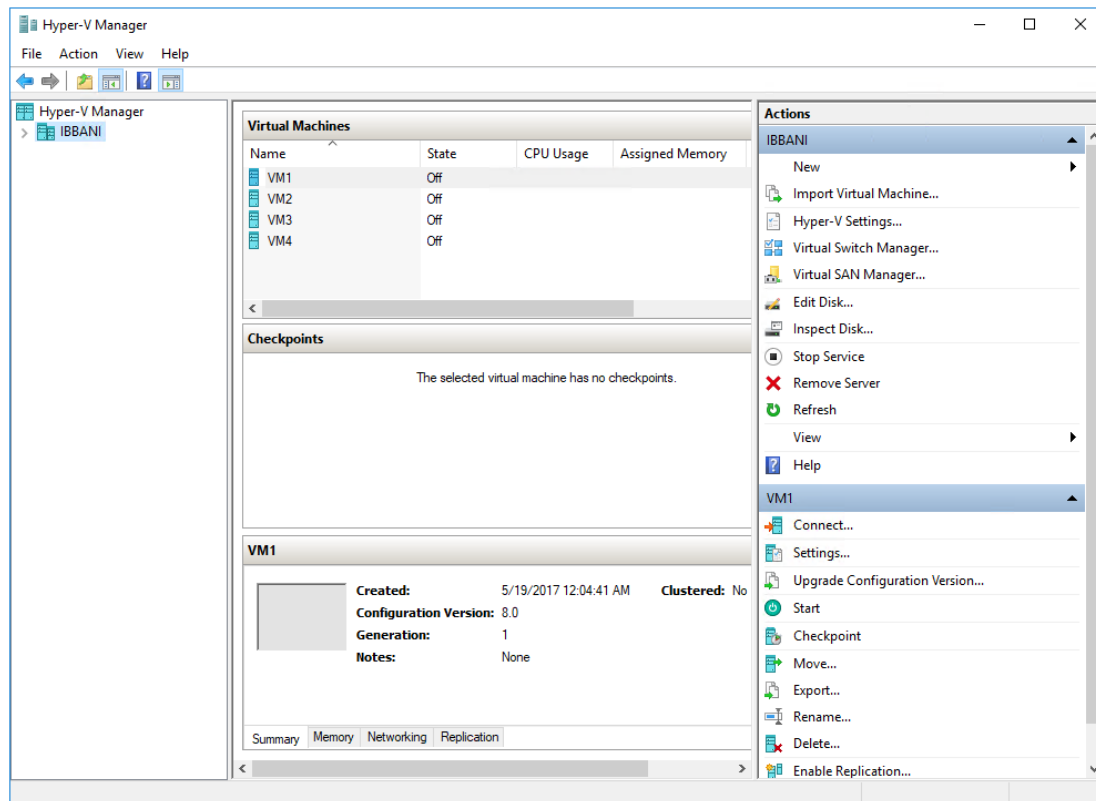
Figure 82 - Hyper-V Manager

b) Click the *Virtual Switch Manager* link under *Actions*.
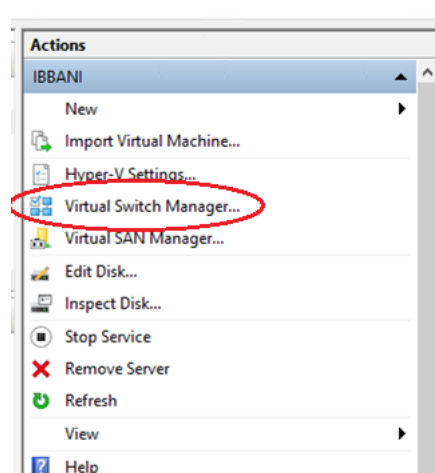
Figure 83 - Opening Virtual Switch Manager

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*.
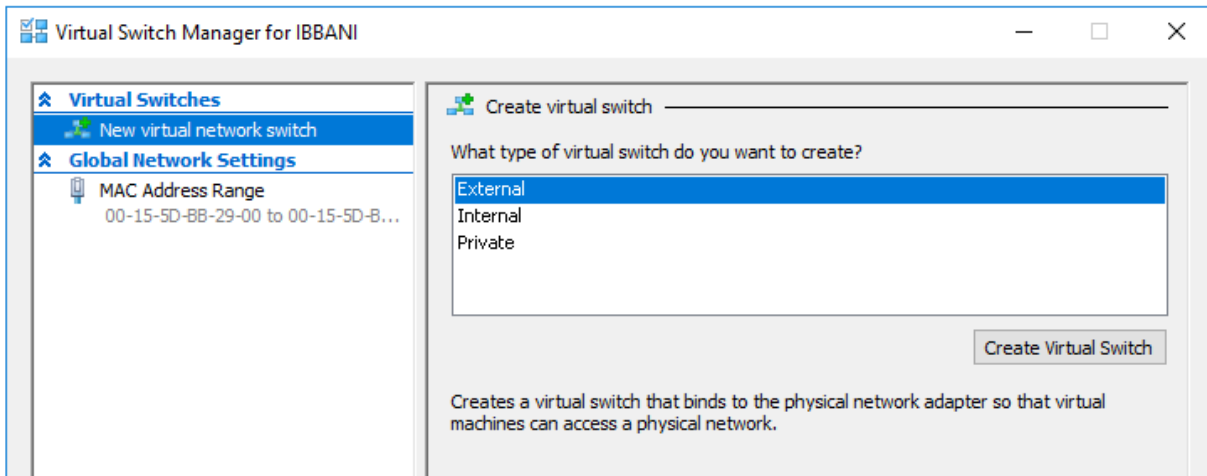
Figure 84 - Creating Virtual Switch

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**.
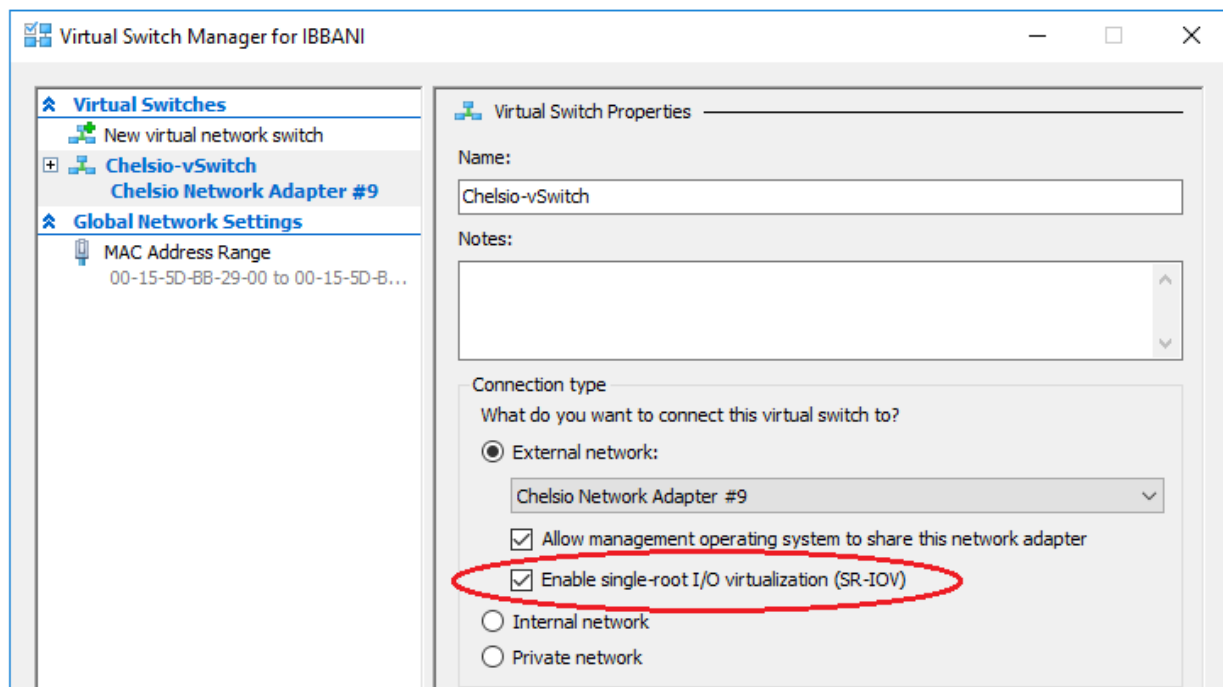
Figure 85 - Enabling SR-IOV on vSwitch

e) Click **Yes** on the prompt that appears next.
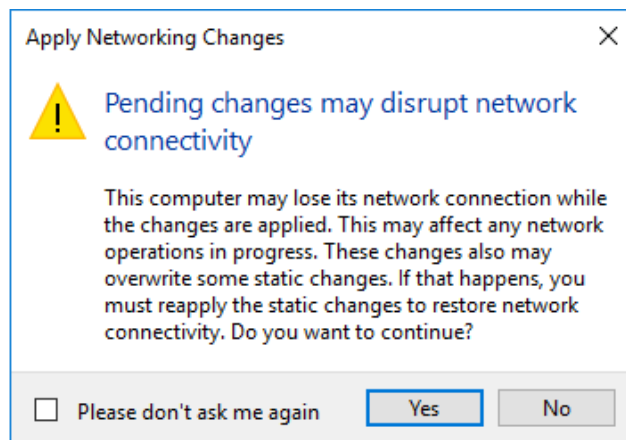
Figure 86 - Confirming Changes

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*.
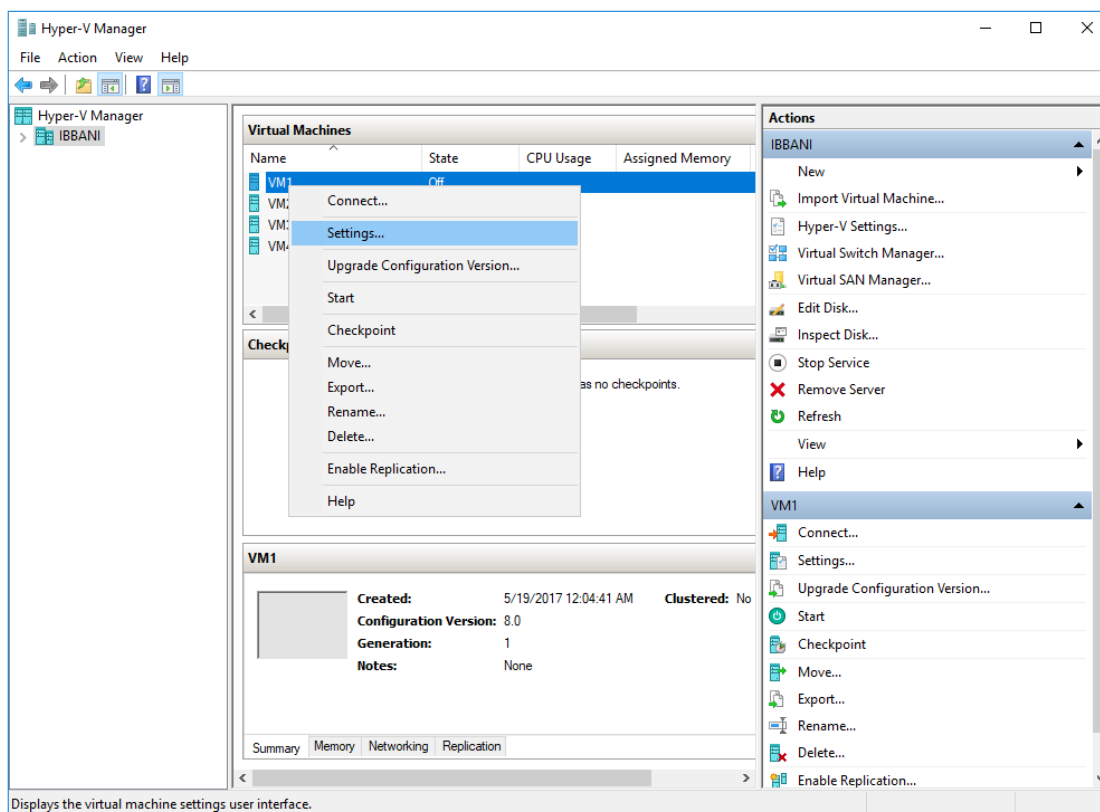
Figure 87 - Hyper-V Manager

b) Expand the *Network Adapter* link, click *Hardware Acceleration* and select *Enable SR-IOV*. Click **OK**.
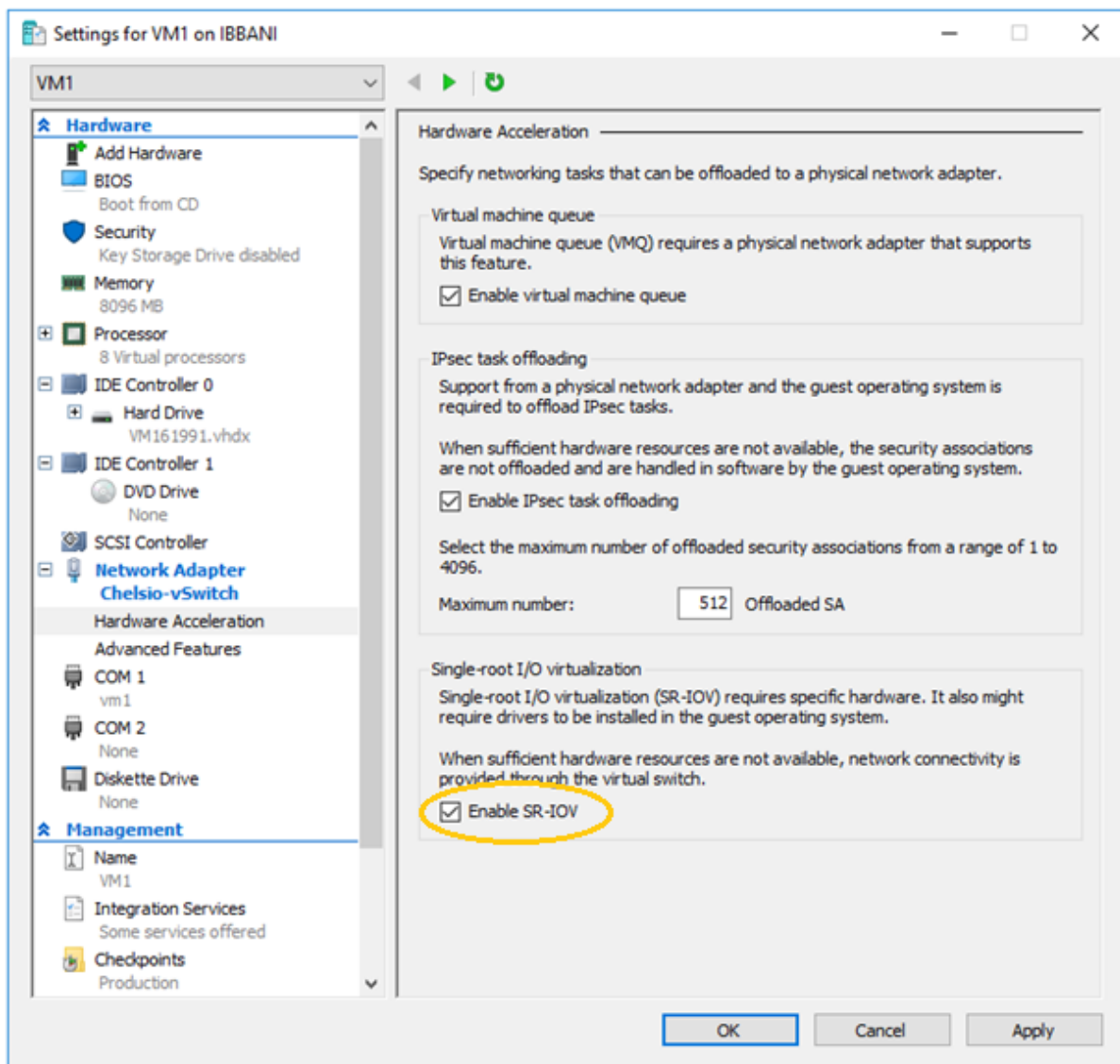
Figure 88 - Enabling SR-IOV on Virtual Network Adapter

c) Right-click on the VM and select *Start* to start the VM.
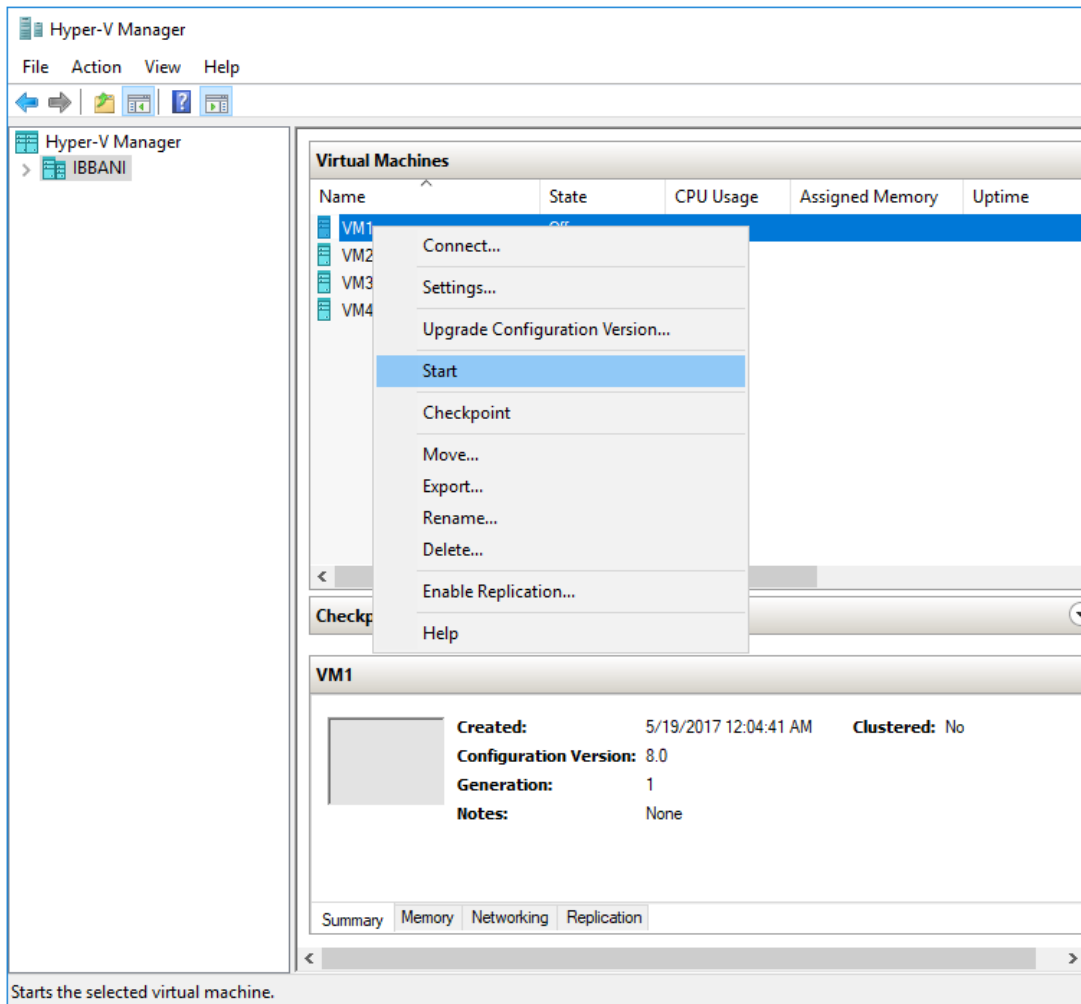
Figure 89 - Starting VM

2.3. Guest (VM) Configuration

i. Install Unified Wire on the guest by referring **Software/Driver Installation** section of the Chelsio 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:

Figure 90 - 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
```
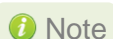
ⓘ Note   Make sure that VM is turned off before running the above command.

Example:

Figure 91 - 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.

Figure 92 - Verifying if Guest RDMA was enabled successfully

vi. In *Hyper-V Manager*, right-click on the VM and select *Start* to start the VM.
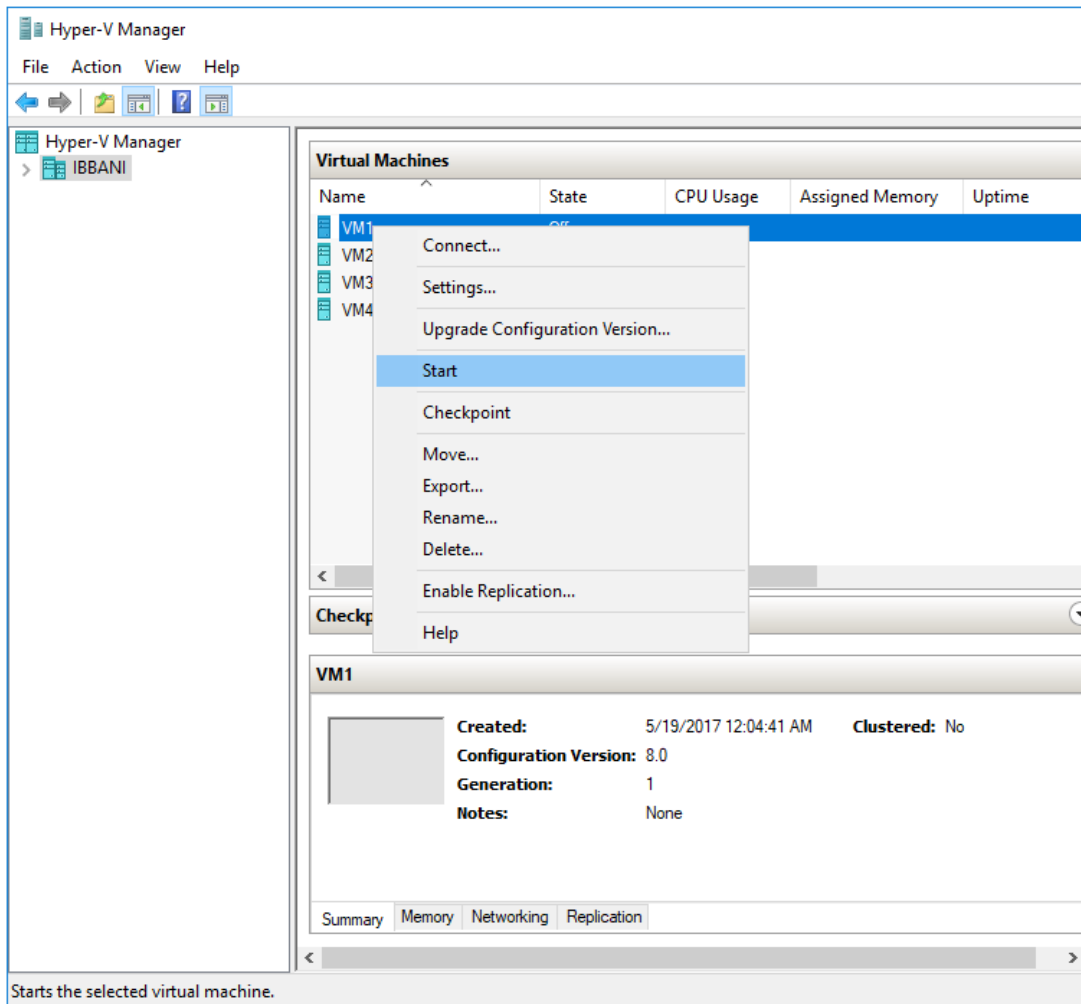
Figure 93 - Starting VM

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.**
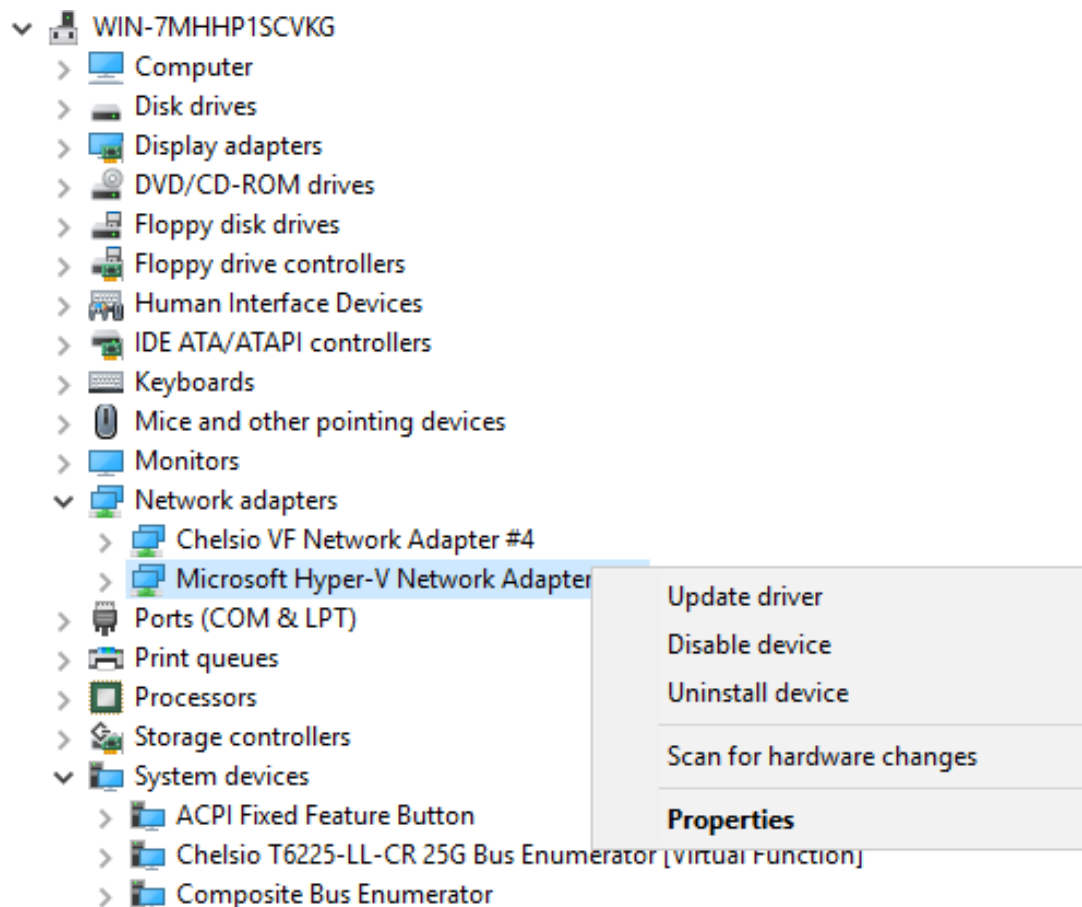
Figure 94 - Device Manager on VM

ix. In the **Advanced** tab, select the *Network Direct (RDMA)* property from the list and enable it. Click **OK**.
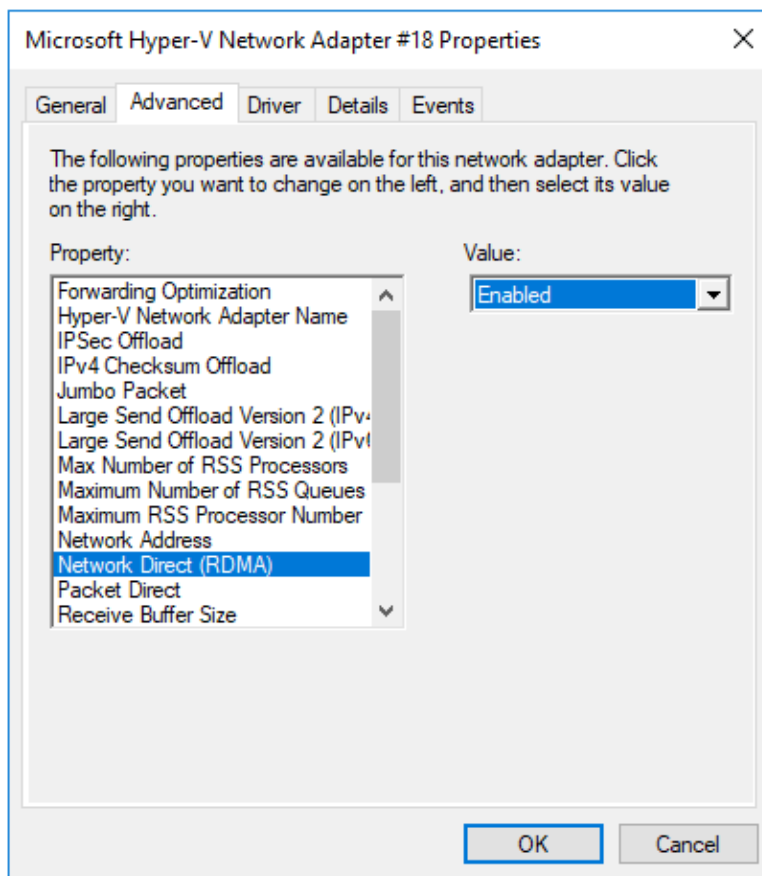
Figure 95 - Enabling Guest RDMA on Microsoft Hyper-V Network Adapter

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:

Figure 96 - 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.
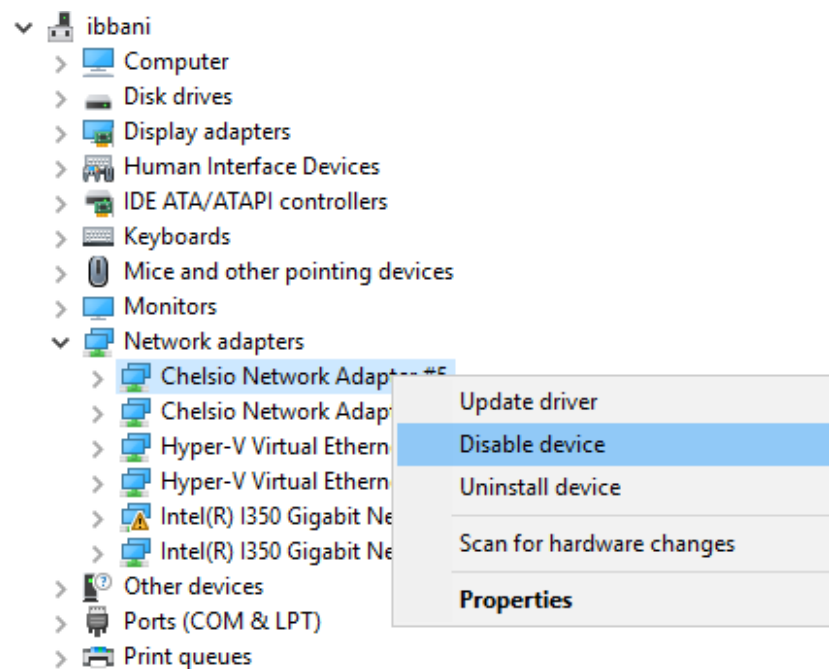
Figure 97 - Troubleshooting

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.

VI. 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

Follow the steps mentioned below to assign IP, Subnet Mask, Default Gateway and VLAN IDs on port 0.

1. Open **Device Manager**, right click on **Chelsio iSCSI Initiator** under **Storage controllers** and click on **Properties**.
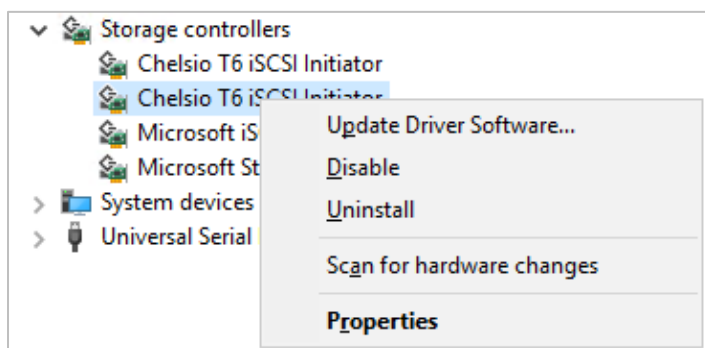
Figure 98 - Device Manager
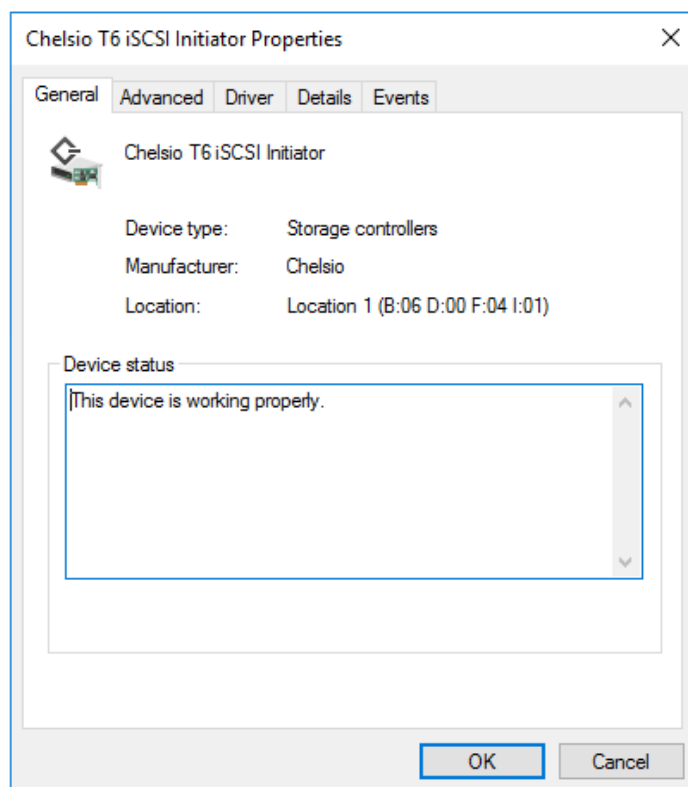
Figure 99 - Chelsio adapter physical port properties

2. Click on **Advanced** tab and select IP option and add IP. Similarly add subnet mask and default gateway.
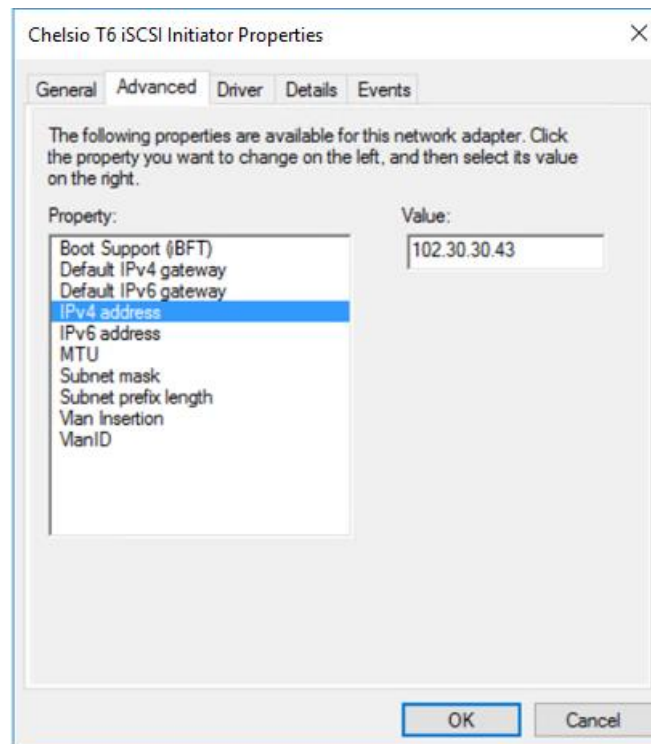
Figure 100 - Assigning IPv4 address
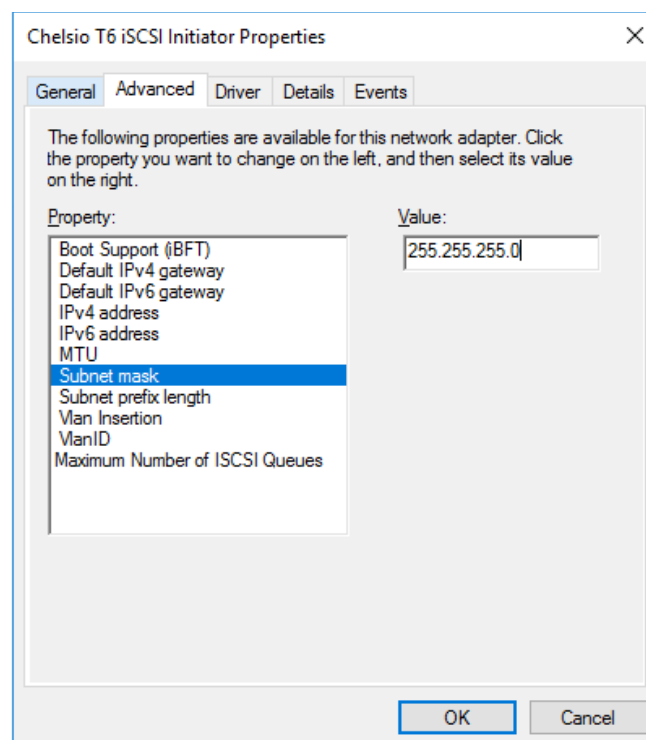
Figure 101 - Adding Subnet mask
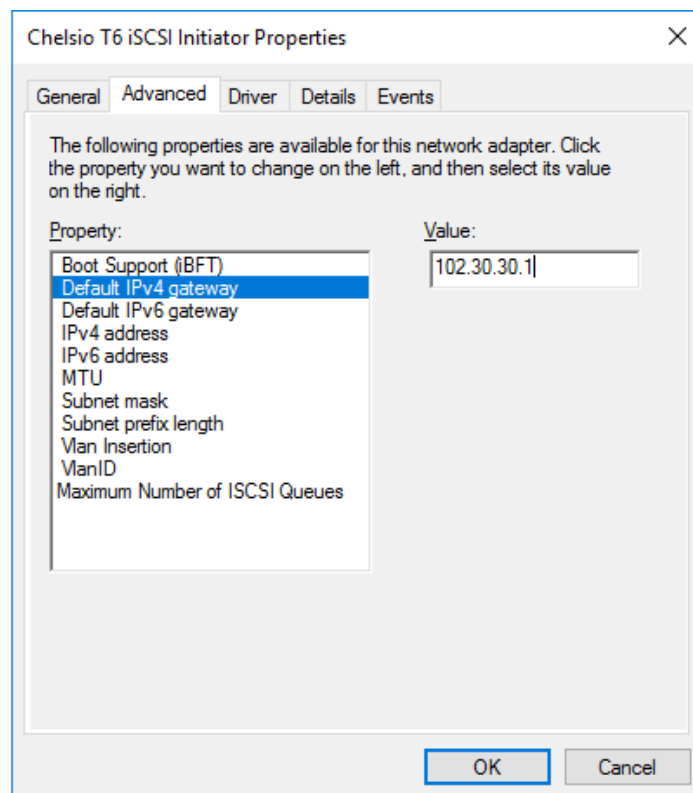
Figure 102 - Adding Default IPv4 gateway
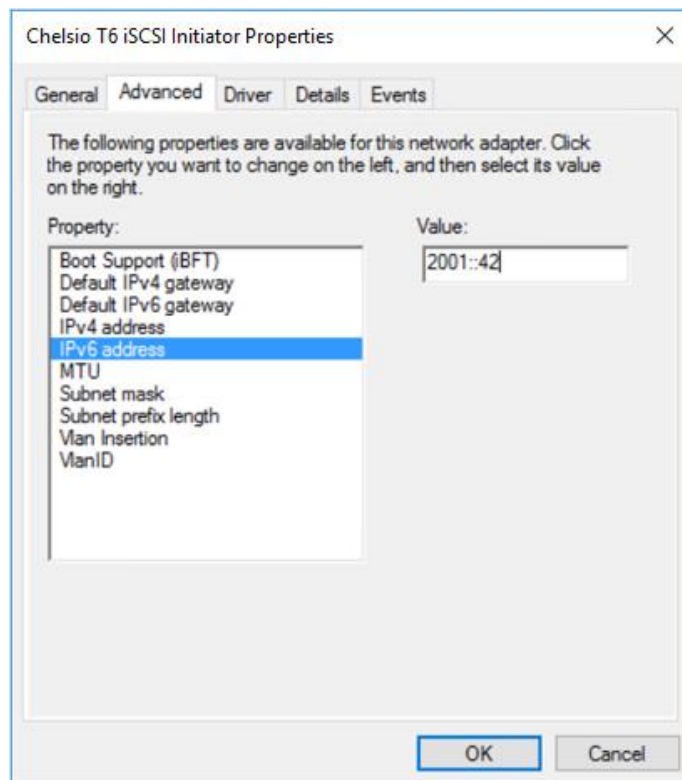
Figure 103 - Assigning IPv6 address
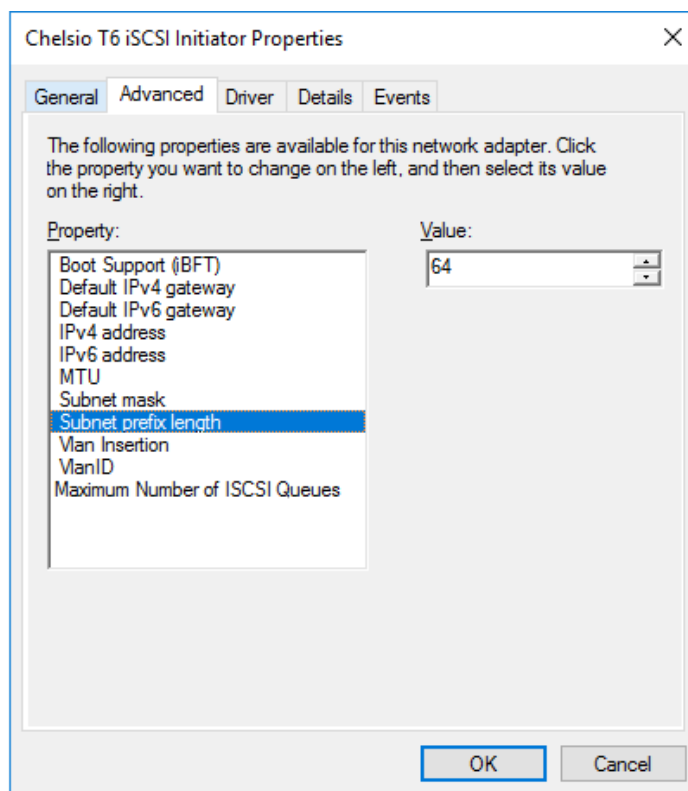
Figure 104 - Adding Subnet prefix length
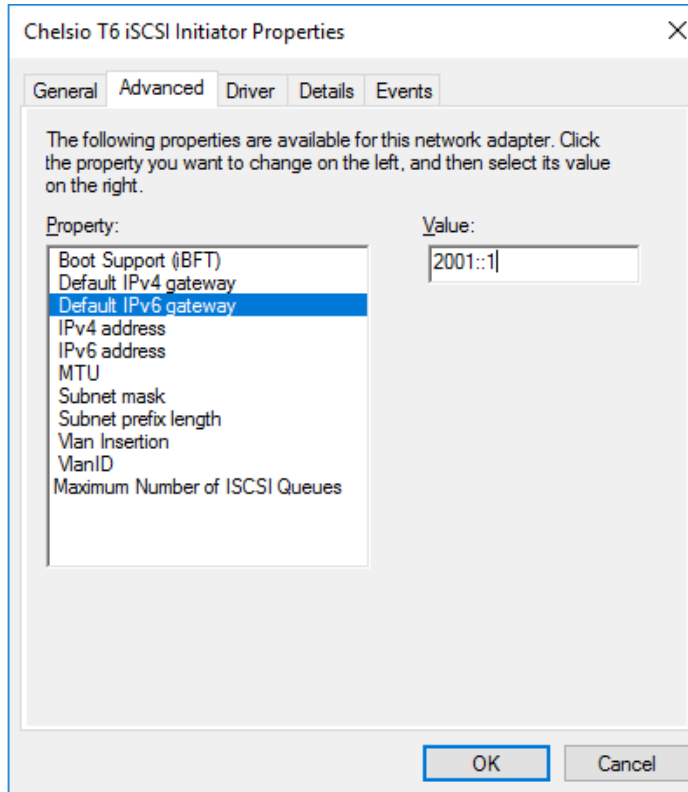
Figure 105 - Adding Default IPv6 gateway

3. Select MTU and assign value as per requirement.
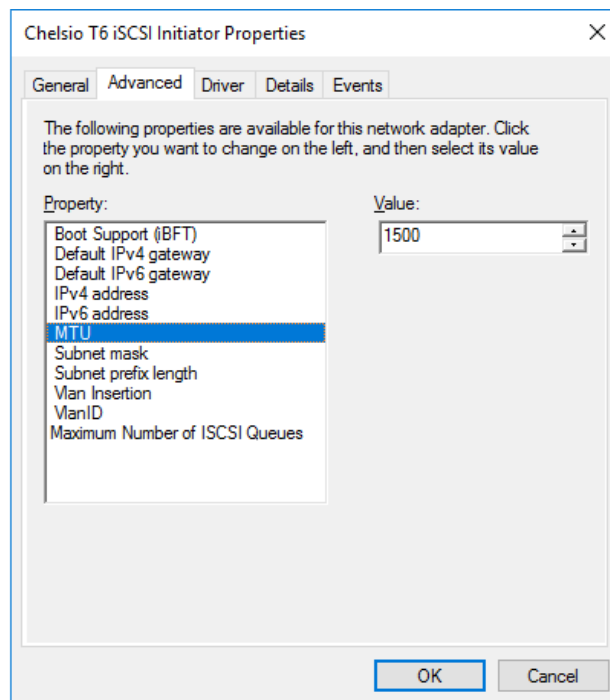
Figure 106 - Setting MTU

4. VLAN is disabled by default, so to enable, click on "Vlan Insertion" and select value as "Enabled".
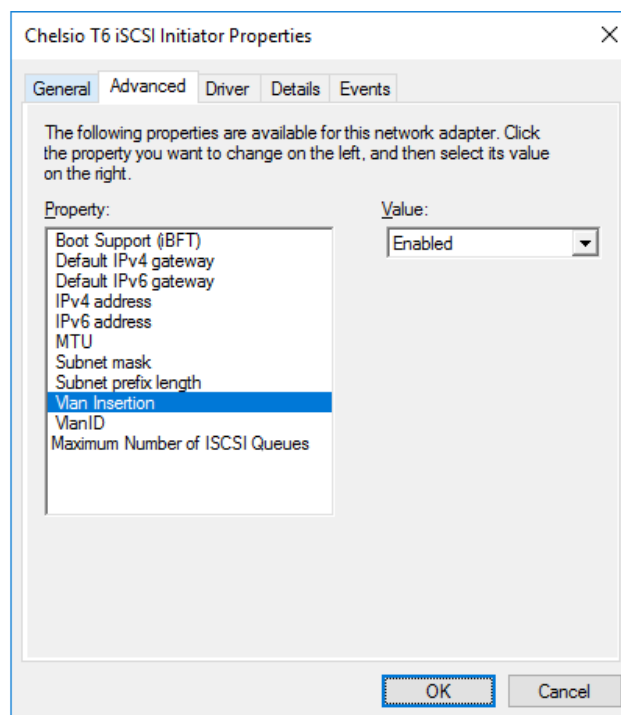
Figure 107 - Enabling VLAN

5. Assign the required VLAN ID and click **OK.**
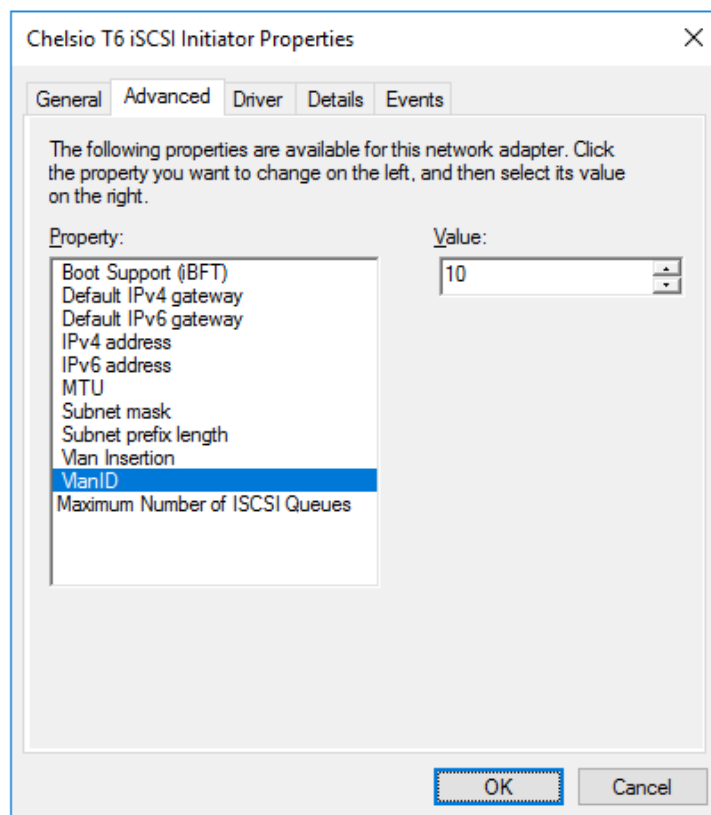
Figure 108 - Assigning VLAN Id

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 a static IP address is assigned in the Device Manager to the respective Chelsio iSCSI node.

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**.**
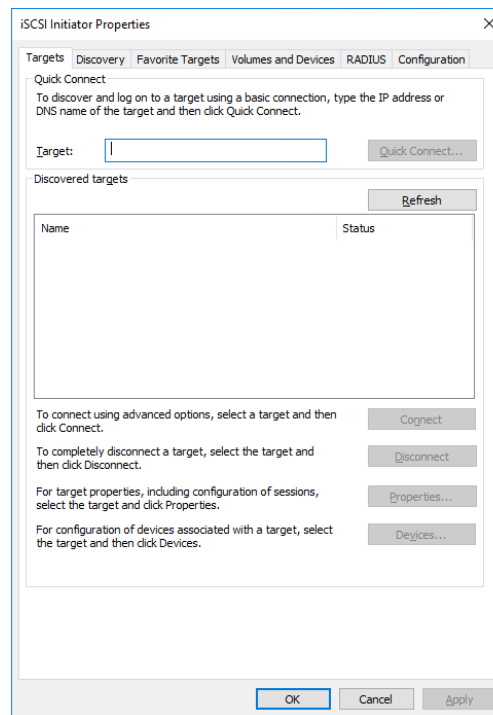
Figure 109 - iSCSI Initiator Properties: Targets tab

2. Choose the **Discovery** tab and click on **Discover Portal**.
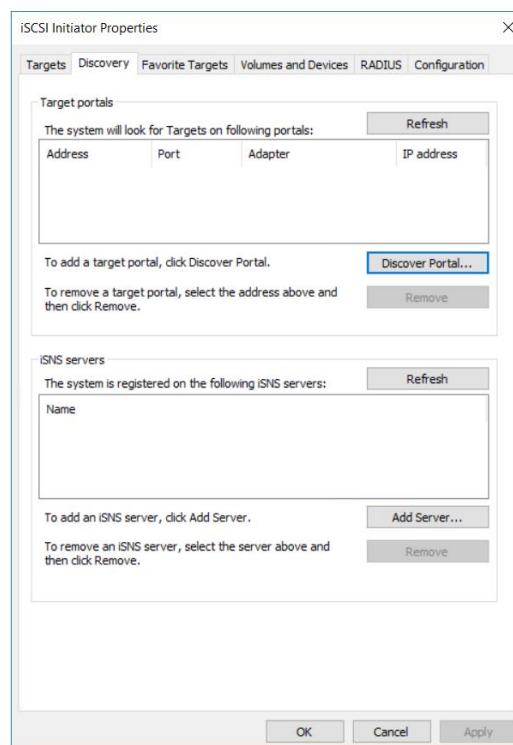
Figure 110 - iSCSI Initiator Properties: Discovery tab

3. Click on **Advanced** button.
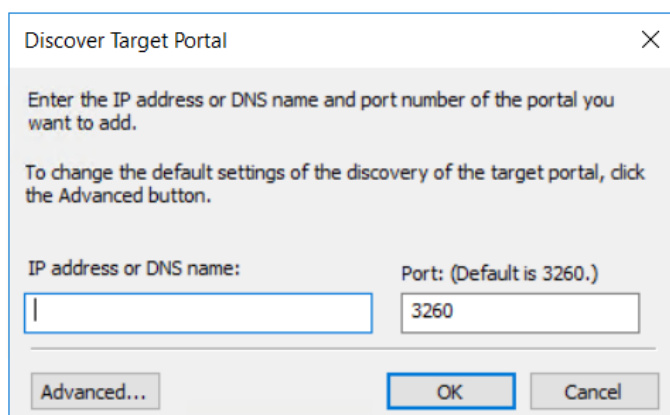
Figure 111 - Discovery Target Portal window

4. In the **Advanced Settings** window, select **Chelsio iSCSI interface** as the local adapter.
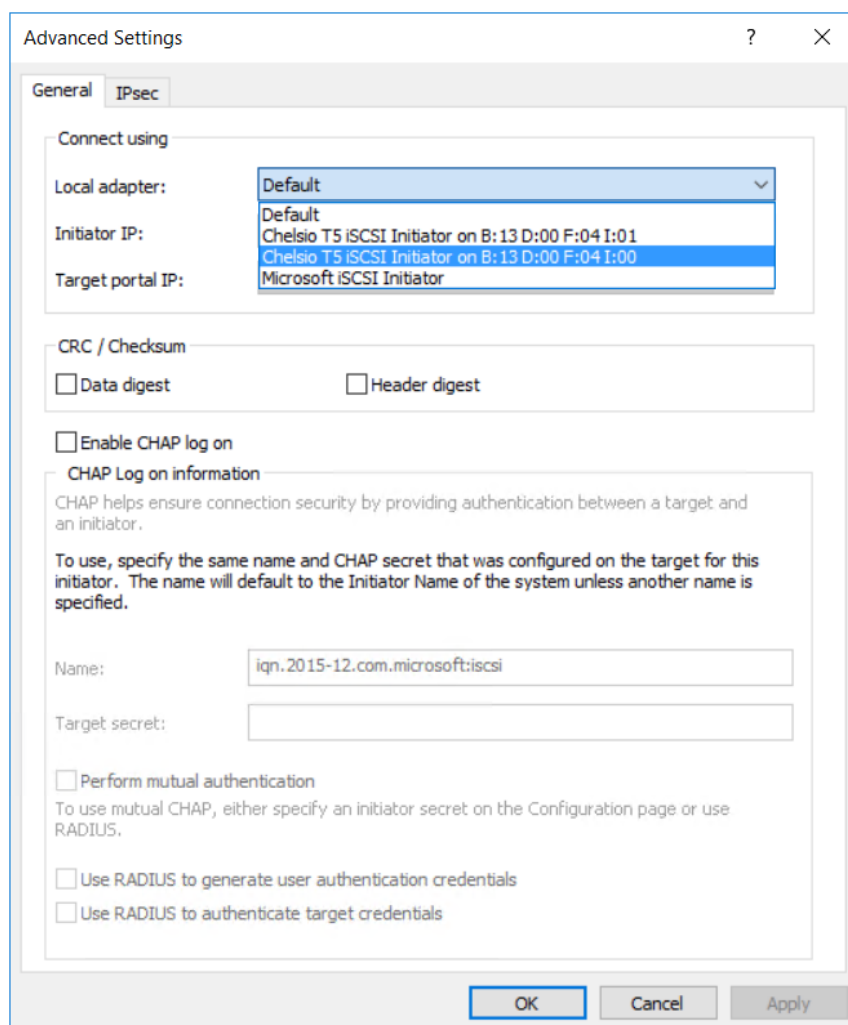
Figure 112 - Discovery Target Portal Advanced Settings: Selecting Local adapter

5. Select the corresponding Chelsio iSCSI Interface IP as the Initiator IP and click **OK**.
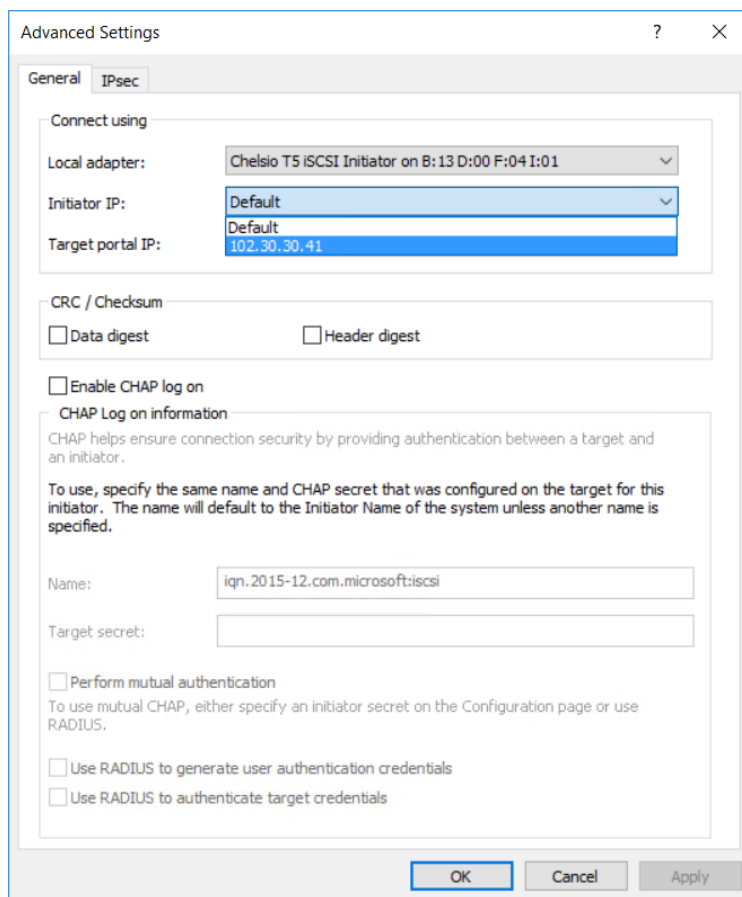
Figure 113 - Discovery Target Portal Advanced Settings: Specifying Initiator IP

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.
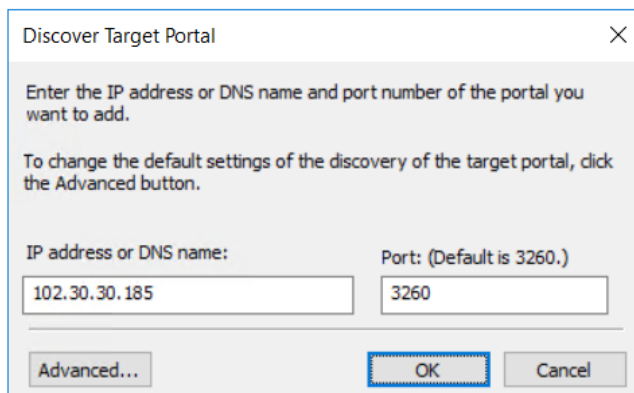
Figure 114 - Adding Target portal

7. Once target portal is added, details like target and initiator machine IP, Port number and Chelsio iSCSI interface IP will be displayed.
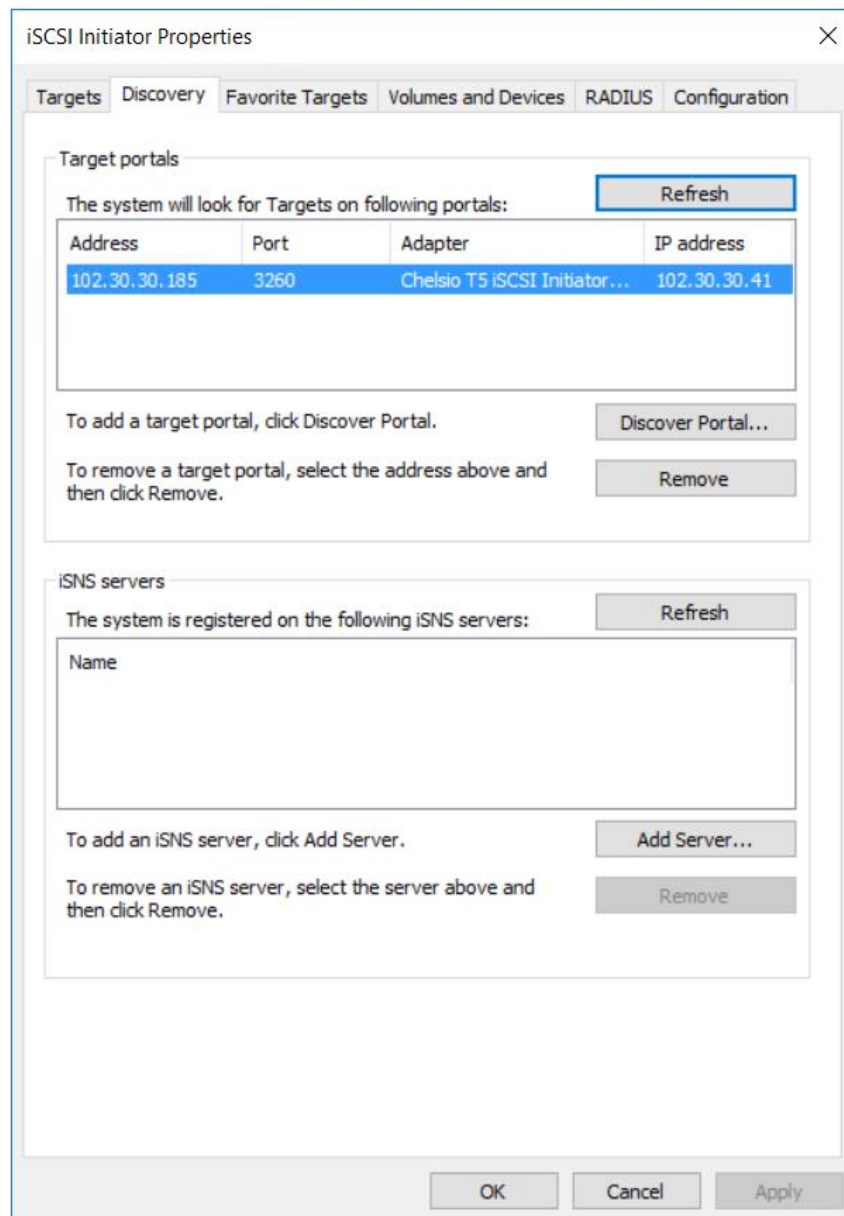
Figure 115 - Target portal added

8. Click on the **Targets** tab to see the list of targets available, choose a particular target and click on **Connect**.
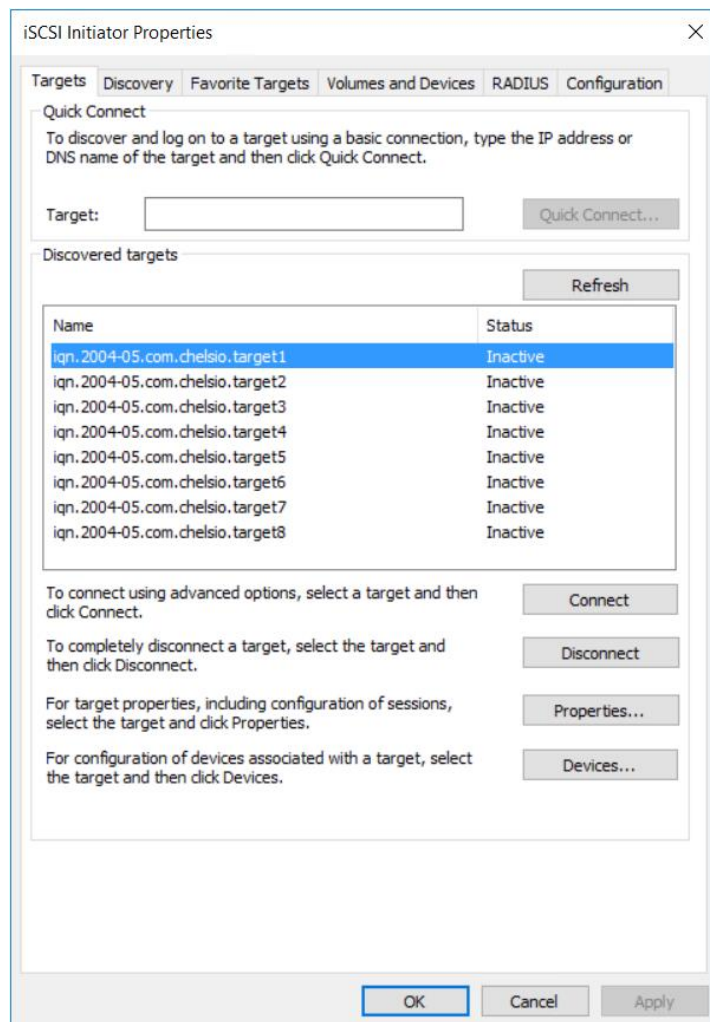
Figure 116 - Targets tab displaying list of available targets

9. A window pops up showing the Target Name, Click **OK**.
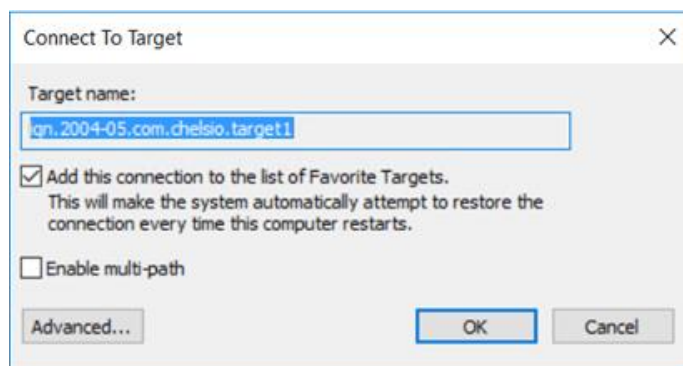
Figure 117 - Connecting to Target

10. After logging in to the target, the state of the target will change from *Inactive* to *Connected*.
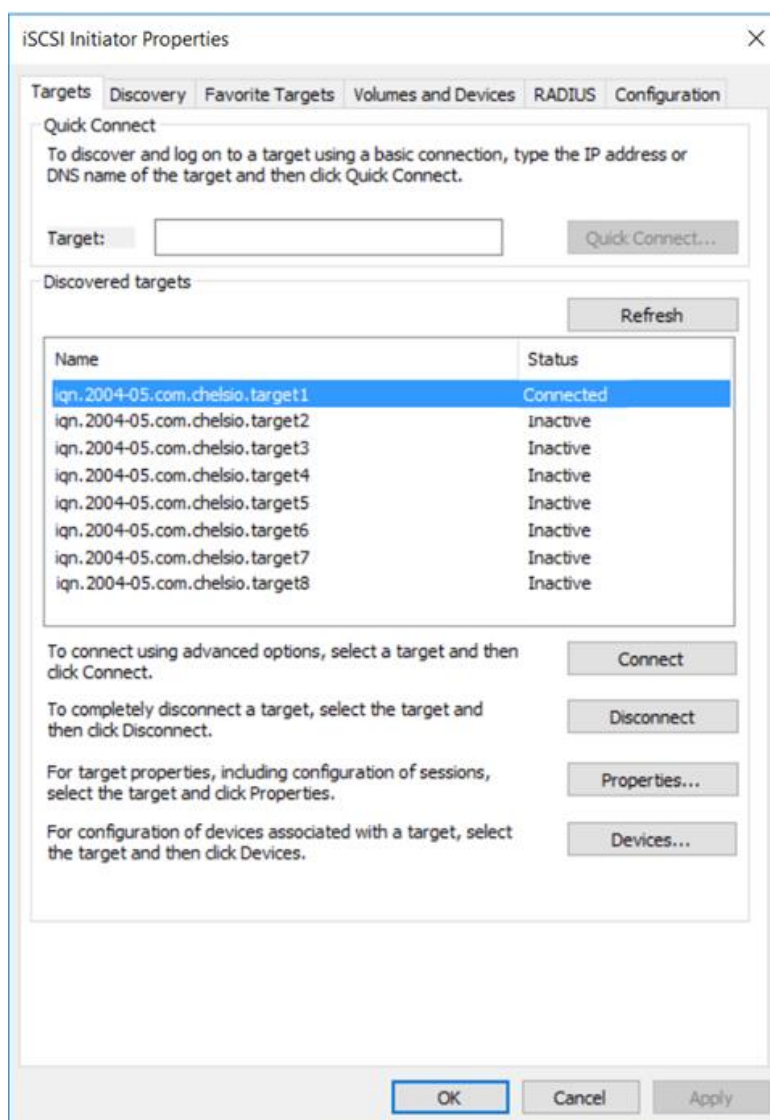
Figure 118 - iSCSI target connected

11. After successful login, go to **Disk Management** and make the respective volume online.
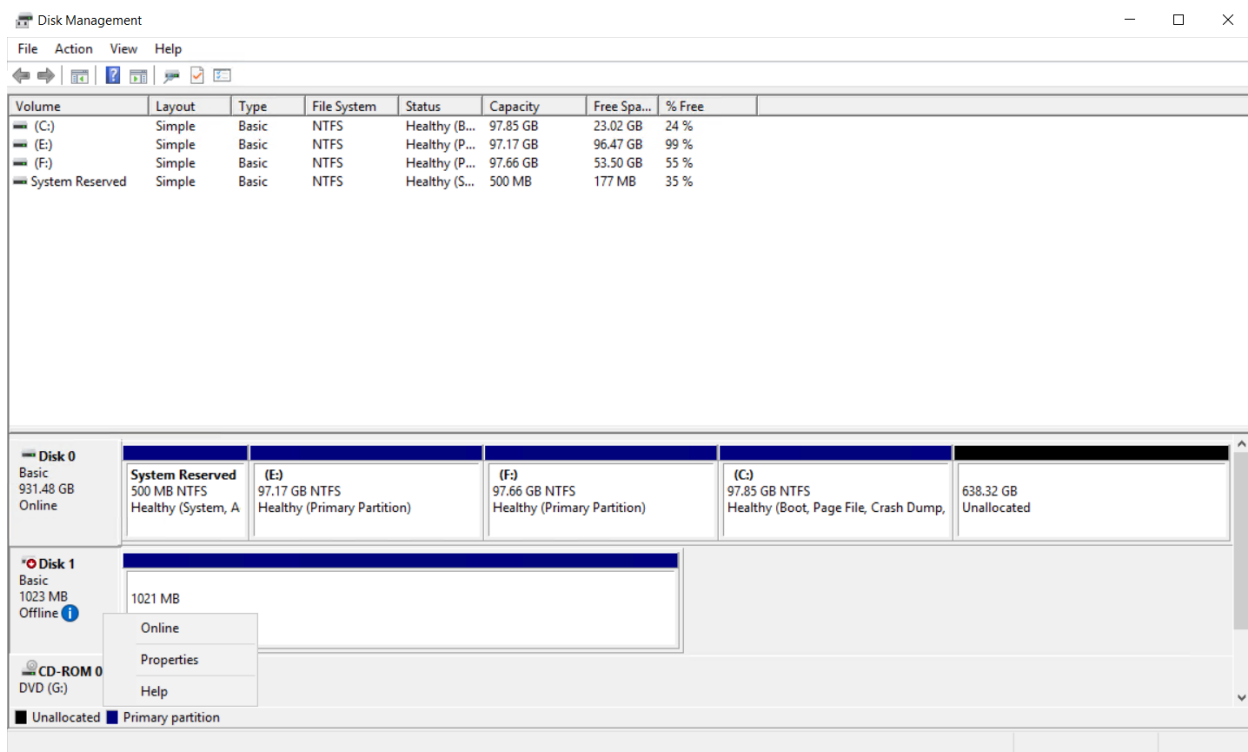
Figure 119 - Disk Management: making volume online

VII. 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.

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
- Nano 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 enabled by default.

> ⓘ 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)#enable dcb
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
```

VIII. Unified Wire Manager (UM)

1. Introduction

Chelsio's Unified Wire Manager is a powerful management software tool, allowing you to view and configure different aspects of the system, including Chelsio hardware installed in the system. The software includes a command line interface (CLI) tool and a web management interface (Web GUI) to help you manage all Chelsio network adapter cards on the network across multiple operating systems.

Unified Wire Manager enables the management of all aspects of the client side of the iSCSI SAN in two main areas. The ability to configure Chelsio adapter's boot option ROM without entering each individual adapter's configuration screen and manage group of iSCSI initiators remotely from a common user interface saves administrator's time considerably. Unified Wire Manager fully supports Microsoft iSCSI initiator. All supported Chelsio FCoE initiators available on Linux can be managed.

Users can manage Option ROM (PXE and FCoE) capability for Chelsio cards using various tools available in the software.

Additionally, Unified Wire Manager allows for Chelsio adapter NIC and TOE parameters to be centrally managed through the same easy-to-use user interface. It can bring interfaces up or down, tune parameters for optimal performance, and any number of administrative tasks normally done at each individual machine.

Chelsio's Unified Wire Manager is an indispensable tool for saving administrator's time for managing the network and SAN. Chelsio's high performance network adapters with its Unified Wire approach to networking can now be managed centrally in a simple and fast way!

1.1. Features

Chelsio's Unified Wire Manager is designed to provide the following features to the end-user:

- Remotely manage Chelsio adapters and various related tasks like driver installation from a single application.
- Single tool with CLI and Web interface that works across Storage, Networking and Hardware.
- Manage all Chelsio adapters installed on the system.
- Tool for FAE to debug issues on the Customer front.
- Freedom to choose various modes of management i.e. CLI or Web GUI.

1.2. Reference Architecture

Chelsio's Web GUI is a web-based management interface that lets you remotely manage several Chelsio CNAs from anywhere, at any time on the network using a web browser. The Web GUI provides a great amount of flexibility, efficiency and accessibility to system administrators in managing the Network and SAN. The users have the freedom to access the interface using any of the major browsers available, based on individual preferences and corporate policy.

System performance degradation issues will not be observed when using the Web GUI, since it's lightweight and utilizes very less system resources.
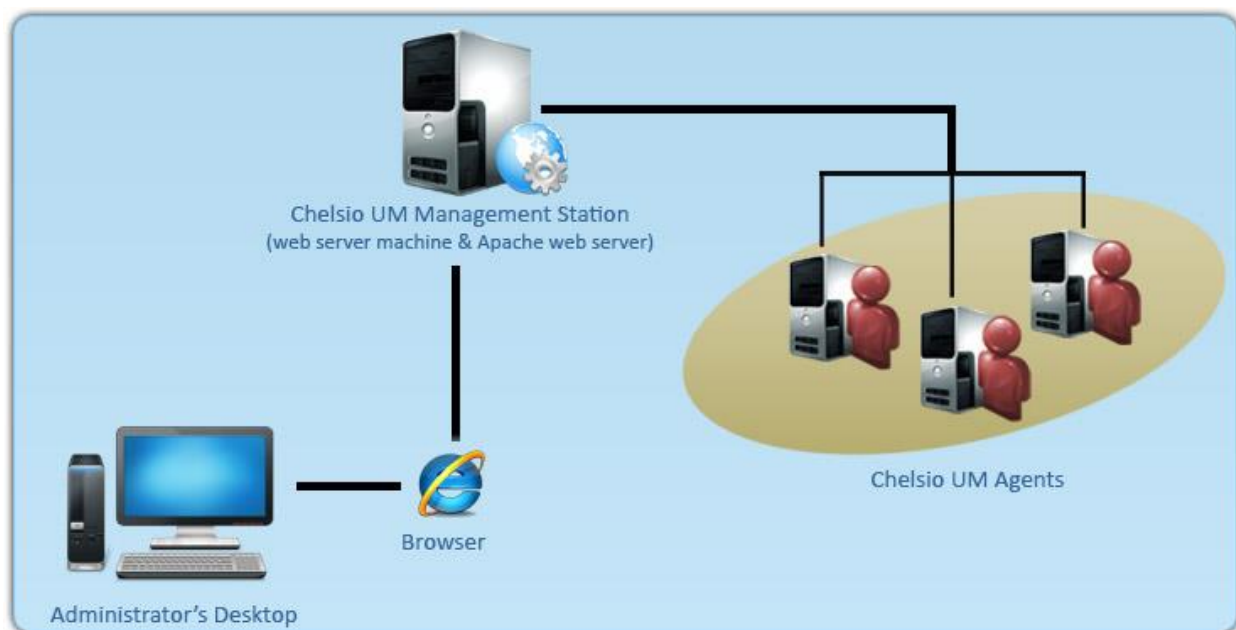
Figure 120 - Chelsio Unified Wire Manager with web interface (Web GUI)

1.3. Unified Wire Manager Components

1.3.1. Management Agent

The Management Agent is a binary executable, which runs as a service on the system that has at least one Chelsio card installed. It is installed along with libraries that can manage various components of the system and enabled during system startup.

1.3.2. Management Client

The Management Client can be used to connect to agents and manage them. Once connected you can view and configure Chelsio CNAs and related networking, storage and hardware properties. You can use either the CLI or Web GUI client to manage agents based on your preference. It makes service requests based on the command issued by the user and returns the appropriate information.

• CLI Client

The **CLI Client** (*chelsio_uwcli*) is an executable binary which allows you to manage and configure agents using the command-line interface. It is not a command shell with a prompt; it accepts all command parameters as arguments when launching it, making it script-friendly.

• Web GUI Client

The **Web Management Interface** (Web GUI) client is a web-based management interface which allows you to securely manage agents from anywhere using a web browser. The management interface uses a secure 256-bit encrypted HTTP connection, ensuring that authentication and configuration data are protected during transmission from the web browser to the system and vice versa. Many agents can be accessed on single interface making it very efficient & user-friendly.

Currently supported browsers are **Internet Explorer 9+**, **Mozilla Firefox 3.6.9+**, **Google Chrome 5+** and **Apple Safari 5+**.

1.4. Authentication and encryption

The Unified Wire Manager requires user authentication to manage a system. A user must have administrative privileges to manage a system. The authentication credentials, as well as all data exchanged between the CLI client or the Web GUI and the agent, are encrypted using SSL. This ensures that the data cannot be accessed when it is being transmitted over the network.

2. Hardware and Software

2.1. Supported Adapters

Following are the currently shipping Chelsio adapters that are compatible with Chelsio Unified Wire Manager:

- T580-CR
- T580-LP-CR
- T580-SO-CR
- T540-CR
- T520-CR
- T520-LL-CR
- T520-SO-CR

2.2. Platform/Component Matrix

The table below lists the Windows versions and the supported UM components.

Table 2 - Supported UM components

Version/Distribution	Supported UM Components
Windows Server 2012 R2	Management Agent, Management Client, Management Station

2.3. Platform/Driver Matrix

The table below lists the Chelsio driver(s) and their supported versions:

Table 3 - Supported NIC version

Chelsio driver	Version
NIC	6.6.4.0

3. Installation (Zip Package)

This section describes the procedure to install UM. This is applicable only for zip package since the Unified Wire installer will install UM by default. If an older version of the software exists, the installer will upgrade it to the version provided in the package.

i.   Run the **UnifiedWireManager-x.x.xx-x64** installer application located in *ChelsioUwire-x.x.x.xx/UM* folder.

ii. Click the **Next** button for the Chelsio End User License Agreement Window.
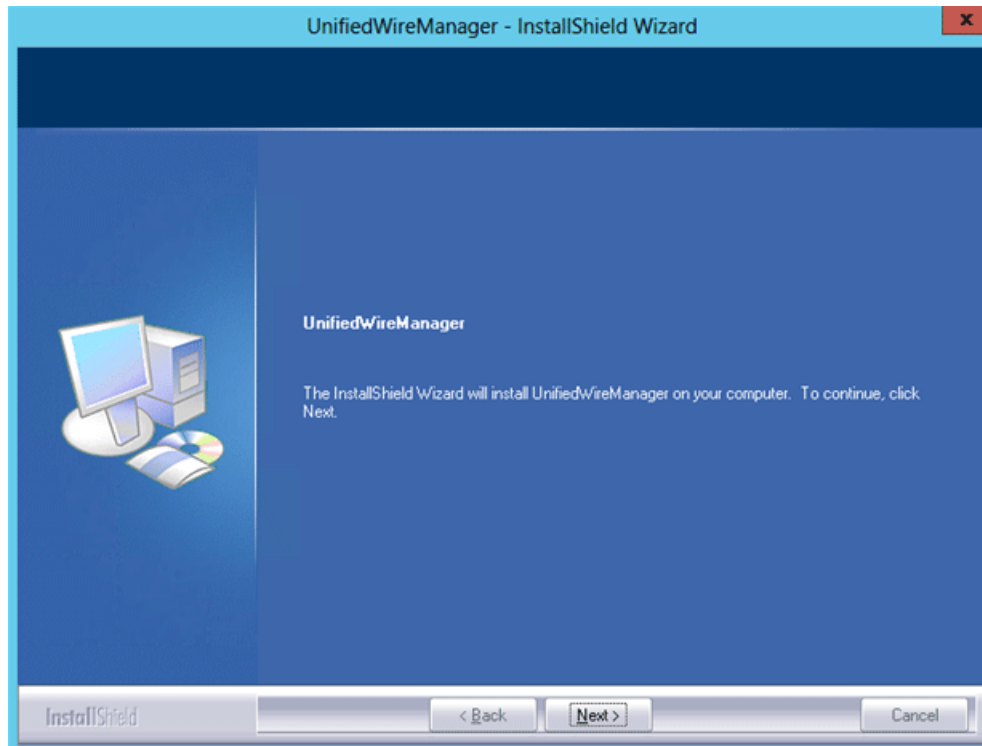
Figure 121 - UM installer welcome window

iii. Select the radio button **I accept the terms of the license agreement** and click **Next**.
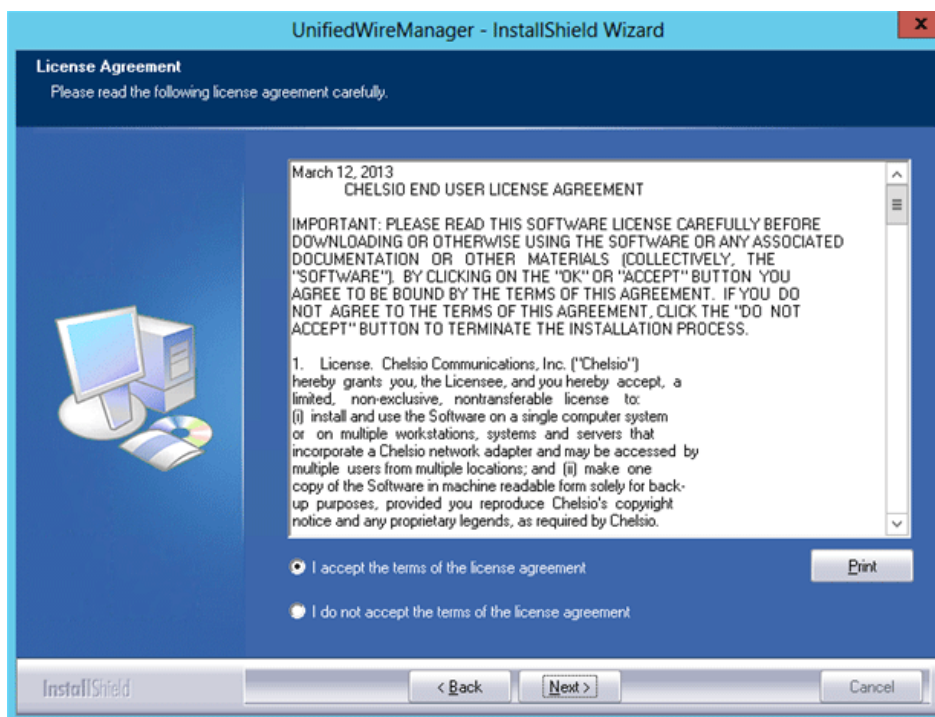
Figure 122 - Chelsio EULA window

iv. The next window will display the pre-requisites for various UM components. Ensure that they are met before proceeding. Click **Next**.
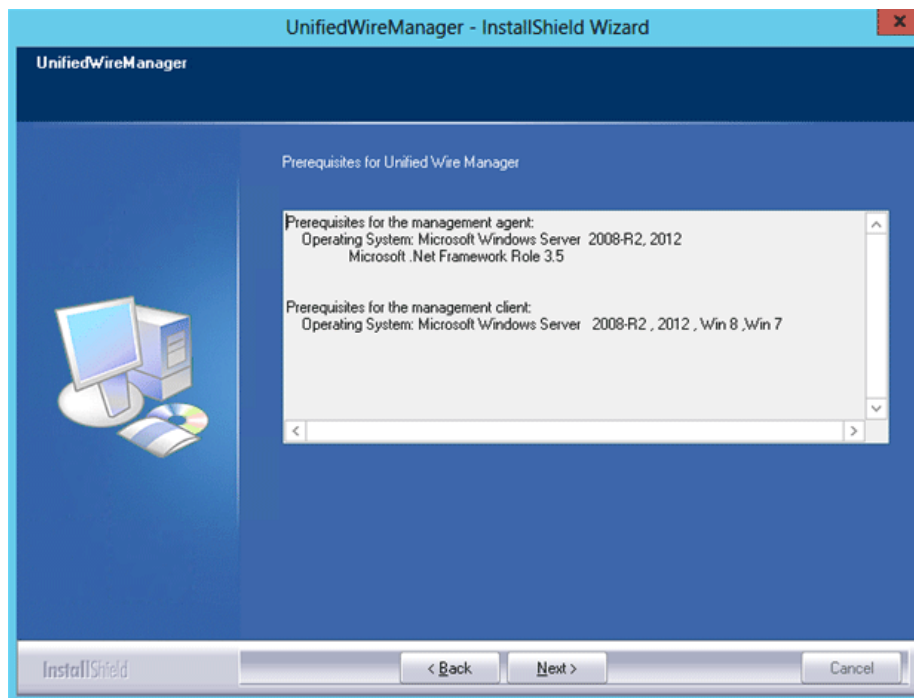
Figure 123 - UM prerequisites window

v. Now, either select **Complete** for complete package installation or else select **Custom** radio button to customize the installation. Click **Next.**
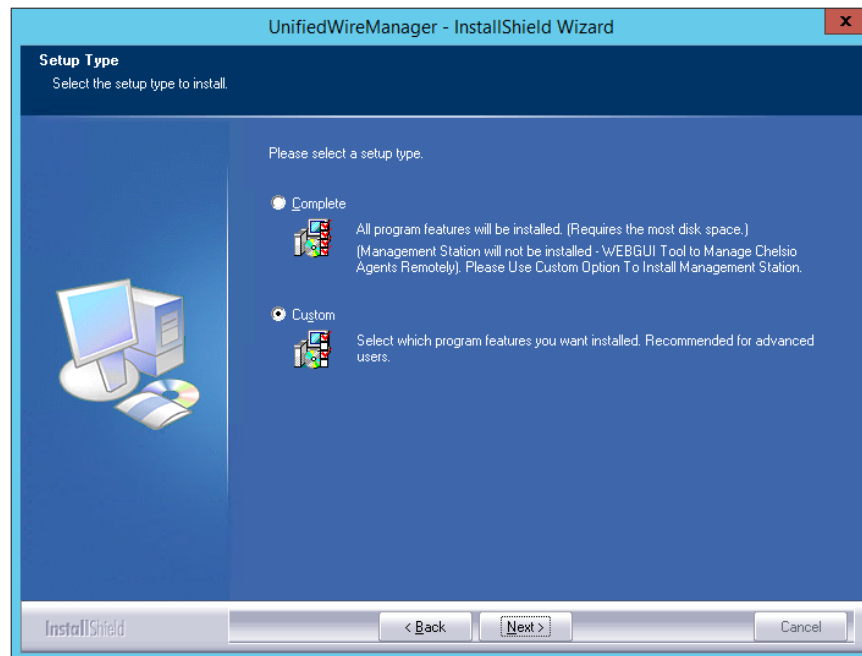
Figure 124 - Select setup (installation) type

a. If you selected **Custom**, the next window will display the location where UM will be installed by default. You can change the location by using the **Change** button or click **Next** to continue with the default path.
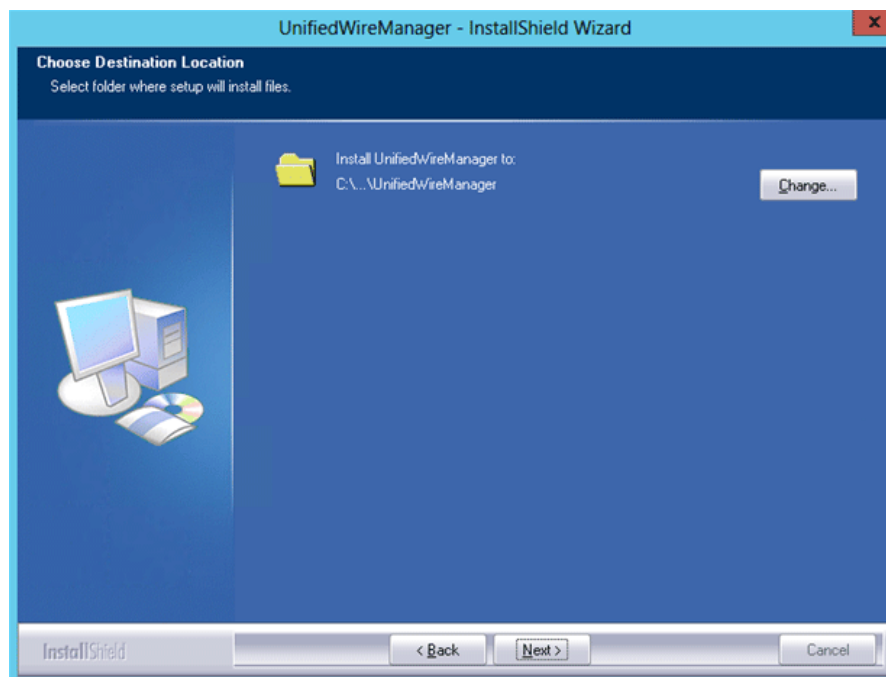
Figure 125 - Changing UM installation path

b. Next, you can choose the UM components you wish to install. Deselect the components you don't wish to install and click **Next.**
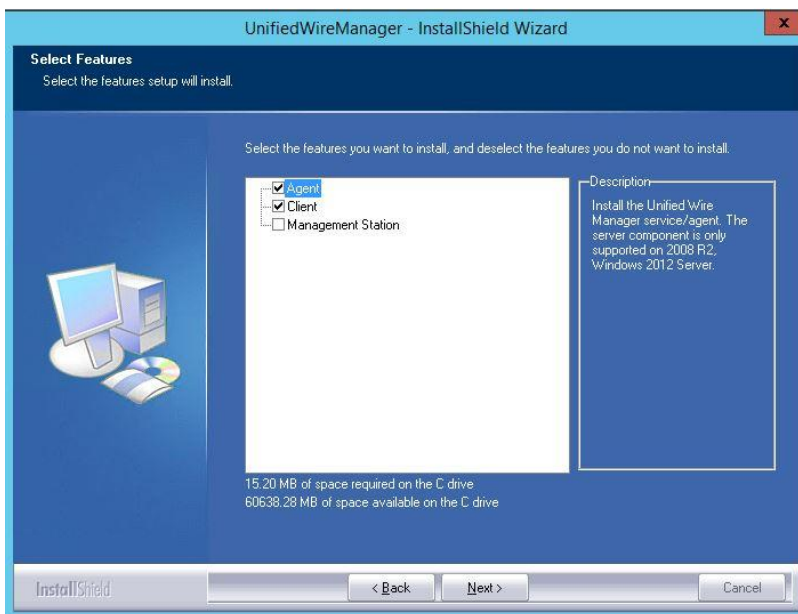
Figure 126 - Selecting UM components to install

> 🛈 Note
>
> *If Management Station is selected, please make sure that all related prerequisites are met before proceeding (See* Pre-requisites*) or else the component will be skipped during installation.*

vi. Click **Install** to start the installation. Unified Wire Manager will now be installed with the selected options.
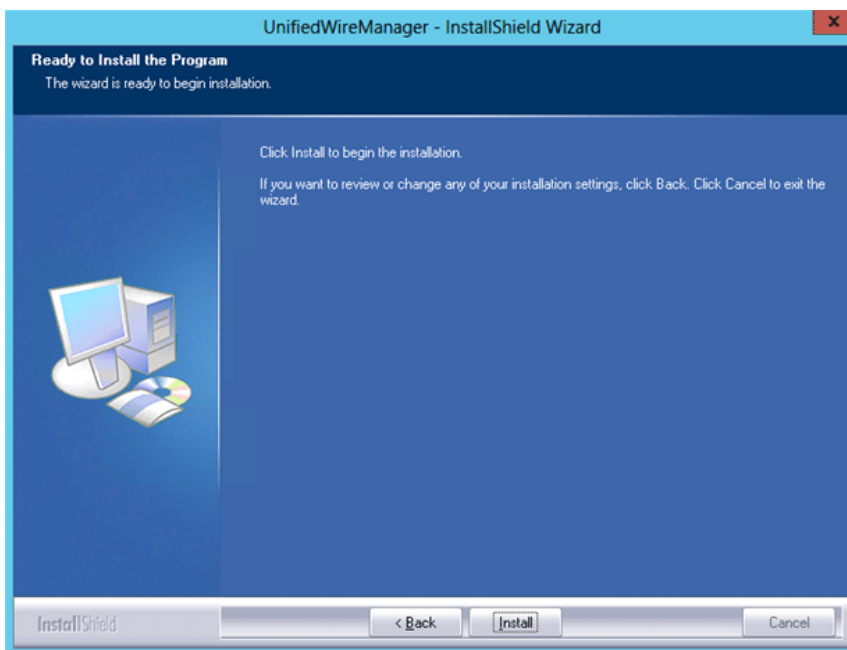
Figure 127 - Starting UM installation

vii. Select **Finish** to exit the UM Installer.
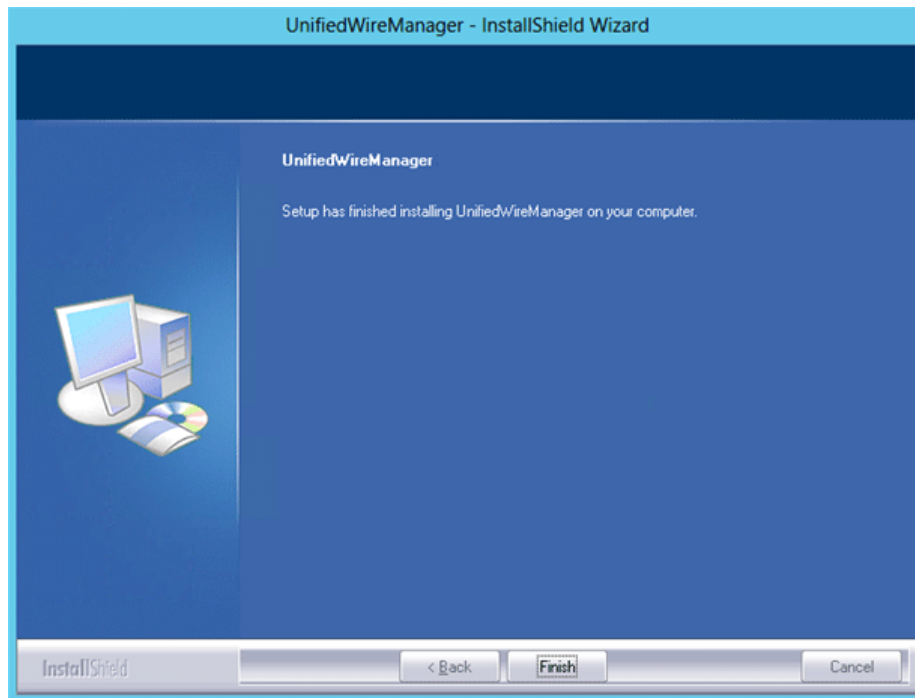
Figure 128 - Finishing UM installation

4. Configuration

4.1. Configuring Management Station

Follow the steps mentioned below to configure Apache Server and SSL certificate.

4.1.1. Apache HTTP Server

i. Assuming that Apache HTTP Server 2.2 is installed at the default location, append the following lines in *C:\Program Files (x86)\Apache Software Foundation\Apache2.2\Conf\httpd.conf*

```
ScriptAlias /python/ "C:/Program Files (x86)/Apache Software
Foundation/Apache2.2/htdocs/chelsio/python/"


<Directory "C:/Program Files (x86)/Apache Software
Foundation/Apache2.2/htdocs/chelsio/python/">

    PythonPath "['C:/Program Files (x86)/Apache
Software           Foundation/Apache2.2/htdocs/chelsio/python']+sys.path"


    SetHandler mod_python
    PythonHandler mod_python.publisher
    PythonDebug on

</Directory>
LoadModule python_module modules/mod_python.so
ThreadStackSize 8388608
```

ii. Perform the following tasks in *C:\Program Files (x86)\Apache Software Foundation\Apache2.2\Conf\httpd.conf* file

   a. Change "DocumentRoot" to "C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs\chelsio"
   b. Uncomment "LoadModule rewrite_module modules/mod_rewrite.so"
   c. Uncomment "Include conf\extra\httpd-ssl.conf"
   d. Uncomment "LoadModule ssl_module modules/mod_ssl.so"
   e. Add the following lines:
       RewriteEngine On
       RewriteCond %{HTTPS} !=on
       RewriteRule ^/?(.) https://%{SERVER_NAME}/$1 [R,L]*

4.1.2. SSL Certificate

   i. Execute the following lines in command prompt and provide info to generate OpenSSL certificate and keys:

```
 cd "C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf"
..\bin\openssl req -config openssl.cnf -new -out blarg.csr -keyout blarg.pem
..\bin\openssl rsa -in blarg.pem -out blarg.key
..\bin\openssl x509 -in blarg.csr -out blarg.crt -req -signkey blarg.key -
days 365
```

ii. Browse to the following location:
C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\extra

iii. Open the file *httpd-ssl.conf* and make the following modifications:
iv. Modify "SSLCertificateFile" to "C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\blarg.crt"
v. Modify "SSLCertificateKeyFile" to "C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\blarg.key"

Note   *On some installation, Apache is seen to report problems regarding SSL cache. In such a case, do the following in C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\extra\httpd-ssl.conf*

a. *Uncomment the following line*
   SSLSessionCache   "dbm:C:/Program Files (x86)/Apache Software Foundation/Apache2.2/logs/ssl_scache"
b. *Comment the following line*
   SSLSessionCache   "shmcb:C:/Program Files (x86)/Apache Software Foundation/Apache2.2/logs/ssl_scache(512000)"

Note   *Use the Windows Event Log (Event Viewer) for troubleshooting any installation related issues.*

5. Verifying UM components status

The following section explains how to verify status of various UM components.

5.1. Verifying Management Agent

i. On the system running Management Agent, launch **Services** from the Control Panel. You can also type s*ervices.msc* in the **Run** command**.** This will open the **Services** window.
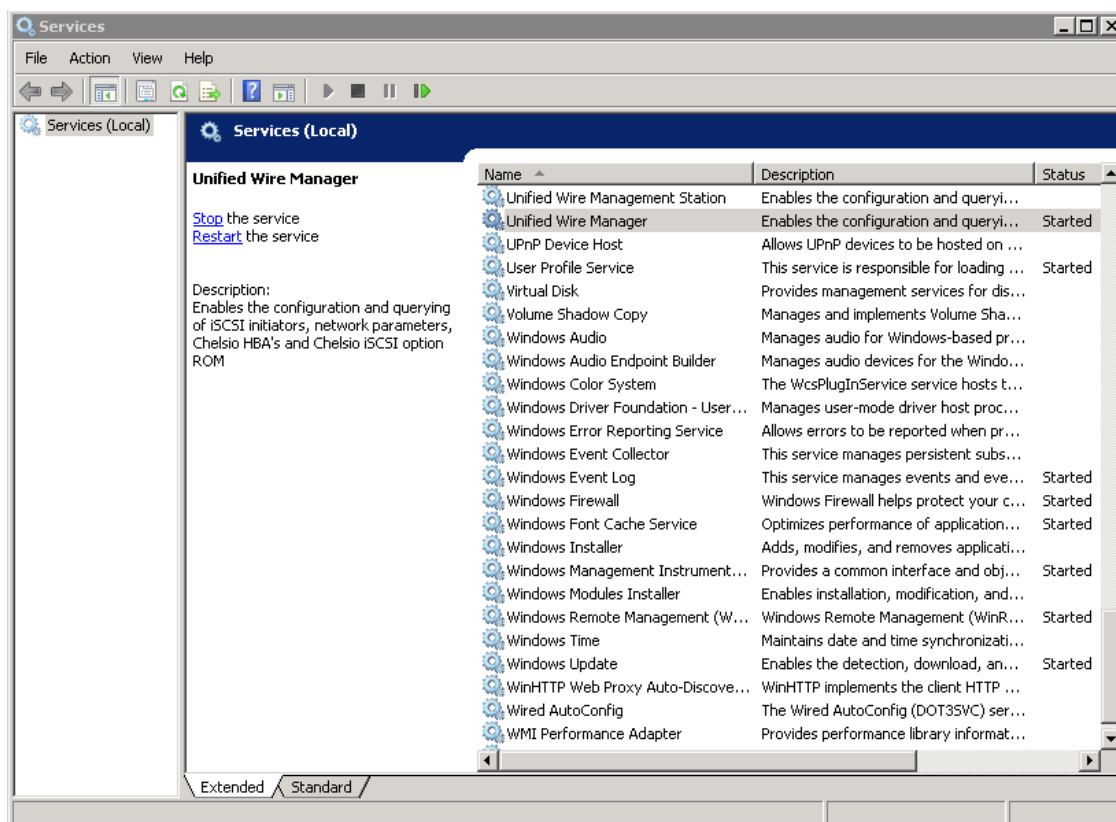
Figure 129 - Windows Services

ii. Search for *Unified Wire Manager* in the list of services. You should get the options to stop and restart in the *Actions* menu on the left side, indicating that Management Agent is already running.
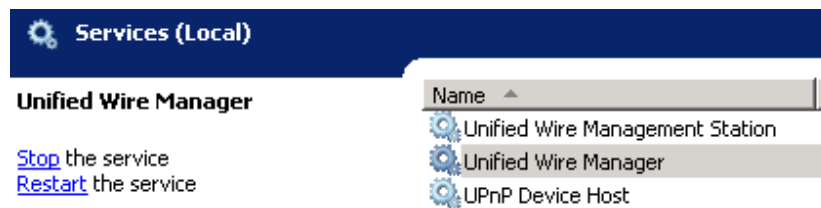
Figure 130 - Actions available for Unified Wire Manager service

5.2. Verifying Management Client

Execute the following query command to determine if Management Client is installed:

```
C:\Users Administrator>chelsio_uwcli.exe -V
```

The above query should confirm that Management Client is installed by displaying a similar result:

```
Unified Manager client CLI version : 2.x.yy
```

5.3. Verifying Management Station

i.   Launch **Services** from the Control Panel. You can also type s*ervices.msc* in the **Run** command**.** This will open the **Services** window.

ii. Search for *Unified Wire Management Station* in the list of services. You should get the options to stop,start and restart the service in the *Actions* menu on the left side.
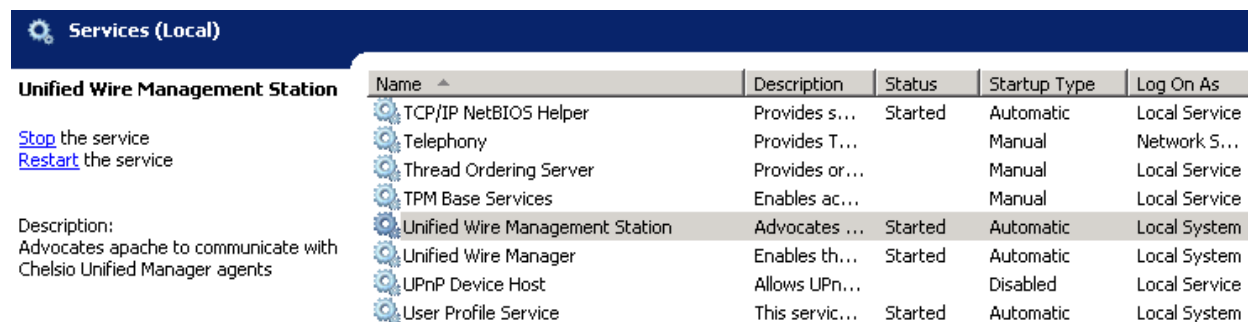
Figure 131 - Verifying Management Station service status

ⓘ Note   *While accessing the Web Management Interface, if an error "Management Station is not running" is displayed at the log-in page, follow the above steps to restart Management Station and try again.*

6. Management Agent

6.1. Communication

The agent uses a TCP connection over IP to communicate with the client. After the connection is established, SSL (Secure Sockets Layer) encryption is enabled using the Open SSL libraries. The agent listens on a TCP port for new incoming connections from clients. This port is set to 35001 by default.

6.2. Service configuration

The agent is installed as a service on the system, and enabled to start on boot. The following sections will describe the procedure to configure service startup manually:

6.2.1. Service startup configuration

The service name on Windows is *Unified Wire Manager* and the service startup configuration can be changed by following these steps:

i.   Launch **Services** from the Control Panel. You can also type s*ervices.msc* in the **Run** command. This will open the **Services** window.
ii. Scroll and highlight the "Unified Wire Manager" service, and double-click on it.
iii. Select the startup type, either "Automatic (Delayed Start)", "Automatic", "Manual", or "Disabled".

6.2.2. Service start/stop/restart

i.   Launch **Services** from the Control Panel. You can also type s*ervices.msc* in the **Run** command**.** This will open the **Services** window.
ii. Scroll and highlight the "Unified Wire Manager" service.
iii. Click on **Start**/**Stop**/**Restart** to carry out the respective tasks in the Description section on the left.

6.3. Firewall

If the system has a firewall configured, it should be configured to allow traffic to the management agent's default TCP port configured i.e. 35001. Review the firewall documentation and configure it appropriately. If there is a firewall appliance / software protecting the network that the system is on, and you wish to connect to the system from a different network, using the client, the firewall appliance also needs to be configured appropriately.

7. CLI client

7.1. CLI Help system

A detailed help and usage documentation is built into the CLI, and is accessible through its help system. The help can be invoked by the usual argument of /? or --help.

7.1.1. Viewing help

i.   Open command prompt by typing *cmd* in the **Run** command.
ii. Type *chelsio_uwcli.exe --help.* Press enter to view the built-in help.

7.2. Client conflict resolution

The CLI and Web GUI cannot manage the same system at the same time by default. This is to ensure that configuration changes being applied by one client are not interrupted by another client. Also, two different Web GUI or CLI clients cannot connect to a management agent at the same time. There is no mechanism to allow this scenario.

8. Web GUI client

8.1. Management Station

In order to access the Web Management Interface, start the Apache HTTP server, if it's not running already. Also, Cookies and Javascript must be enabled in the browser.

8.1.1. Running Management Station

i.   Launch **Services** from the Control Panel. You can also type s*ervices.msc* in the **Run** command. This will open the **Services** window.
ii. Search for "*Apachex.x*" in the list of services and click *Start* in the **Action** menu on the left*.*
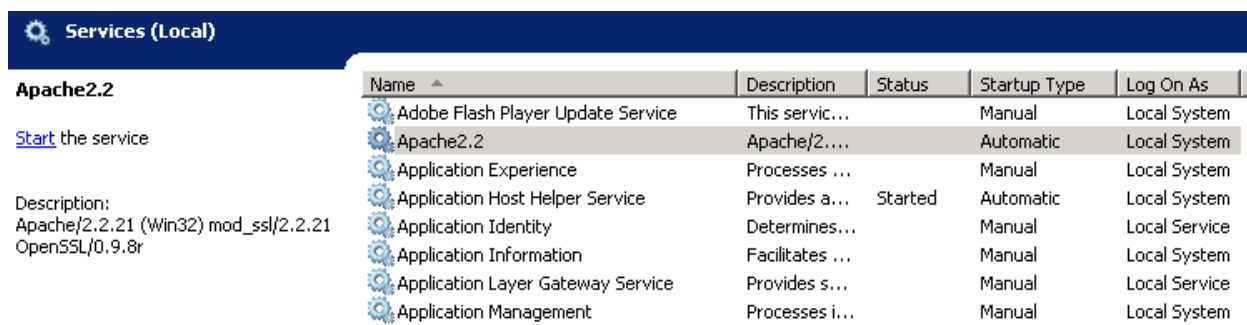
Figure 132 - Starting Apache HTTP server

iii. Search for "*Unified Wire Management Station*" and click *Start* in the **Action** menu on the left*.*
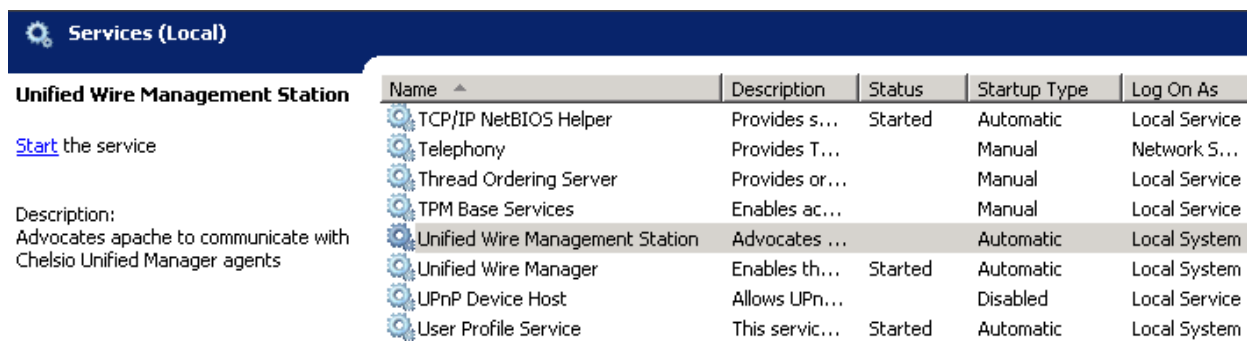
Figure 133 - Starting Management Station service

8.2. Accessing Web Management Interface

i. To access the Web GUI, type in the URL https://<management station IP address> in a web browser.

ii. The security certificate used by the web server is a generic one. It may cause the following types of prompts in different browsers. You will need to select the correct option to continue.
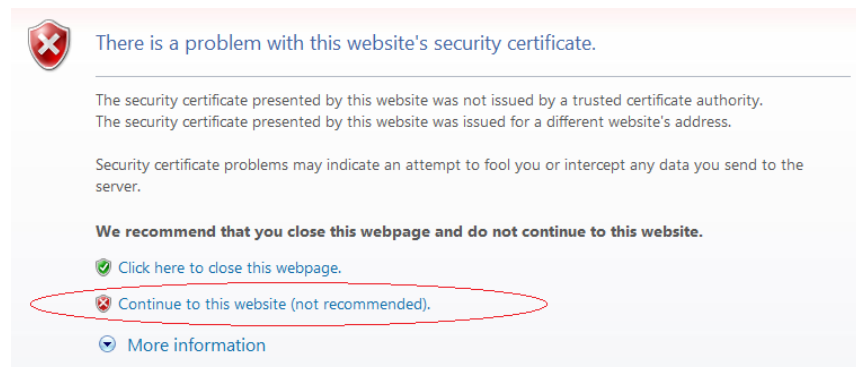
Figure 134 - Security Certificate prompt in Internet Explorer
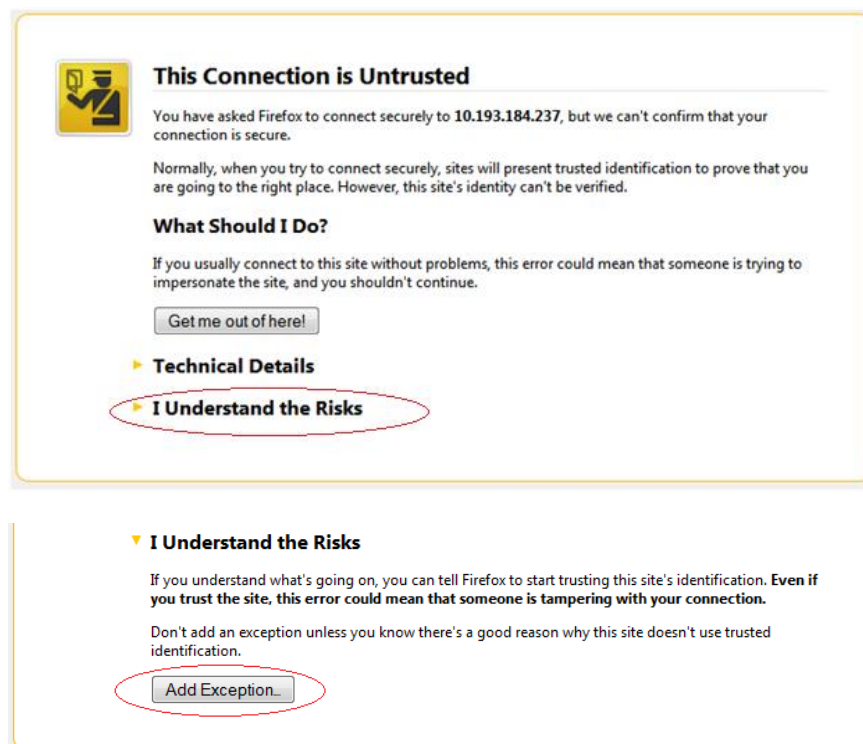
Figure 135 - Security Certificate prompt in Mozilla Firefox
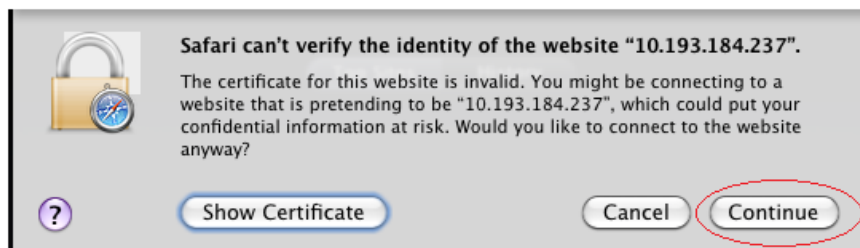
Figure 136 - Security Certificate prompt in Apple Safari

Figure 137 - Security Certificate prompt in Google Chrome

iii. The web interface requires password authorization to be accessed. Enter the administrator/root credentials that were set up on the management station system and click on the *Login* button.

Figure 138 - Web GUI Login page

📥 Note   *Not performing any operation/action for 5 minutes will result in session timeout. You will have to re-login and connect to the Agents again.*

8.3. Layout and Navigation

The Web Management Interface consists of the following:

- **Title bar** displaying the username on the left, Unified Wire Manager logo and name in the centre; and a Logout button on the right.
- **Menu Bar** consisting of the **Home, Add System, Remove System, Refresh, Subscribe** and **Bulk Configuration** buttons**.**
- The **Navigation Pane** with a cascading tree of links to various configuration modules for a UM Agent. You can navigate between connected agents and various sections of the managed agent's interface. You can view and hide the configuration modules for each Agent by clicking on the "**+**"and "**-**" links respectively
- The **Details Pane** on the right displaying panels associated with the tree menu item selected in the **Navigation Pane.** The panels can be expanded and collapsed by clicking on the panel heading.
- The **Bottom bar** has the **About** link on the right and copyright details on the left.

Welcome, root
Chelsio Unified Wire Manager 2.4
Logout
Menu
torpedo
torpedo
Network
Storage
Hardware Features
HOME
Add System
Remove System
Refresh
Subscribe
Bulk Configuration
Bookmarks and History
Service Discovery
Bulk Driver Installation

Figure 139 - Web Management Interface

8.4. Home page

The home page is displayed by default on launching the Web GUI. It displays **Bookmarks and History**, **Service Discovery** and **Bulk Driver Installation** modules. Options to go back to home page, add/remove system, refresh and configure email alerts are also available.

8.4.1. Home

This option will display the home page.

◦ Bookmarks and History

A history of the last 128 systems that were managed from this system, by the current user, will be shown here in a list. Each system's management IP address, TCP port, and Login details are also stored. This may be edited and saved. Any systems that are not required in the list may be deleted.

> ⚠ Important
>
> ***Storing login passwords for the managed systems is inherently insecure***. *The passwords are encrypted, but it is still advisable to store passwords only if the system you are running the GUI client on, is secure.*
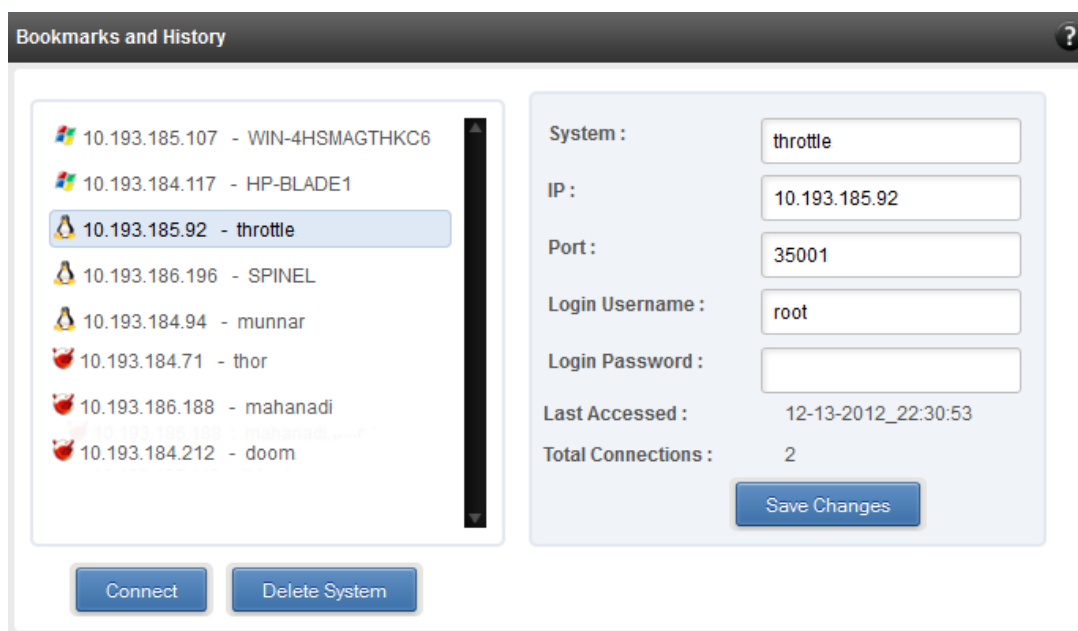
Figure 140 - Bookmarks and history module

- **Connecting to a system**

Select the system from the Bookmark list and click *Connect.* Once successfully connected, the system will appear on the left pane with different related modules on the right to view and manage.

- **Deleting a system**

Select the system from the Bookmark list and click *Delete system* to remove it.

> ⓘ Note   *Once removed, the system will no longer appear in the Bookmarks and History module. If you wish to manage that system again, you will have to use the "Add system" option.*

Service Discovery

Using this module, all the Unified Wire Manager agents connected in the same or different subnet can be discovered. One can choose to discover agents based on OS type or search for a particular agent if the agent's IP or hostname is known. Select the appropriate discovery method and provide the relevant information. For example, to search using hostname, select *Hostname* as the **Input Type** and provide the agent's hostname in the **Search for Hostname/IP** field. Finally click **Discover Agents**.

The **Add Agents** button adds the selected system to the list of discovered agents in the **Bookmarks and History** module. The **Clear Agents** button resets the list of discovered agents.
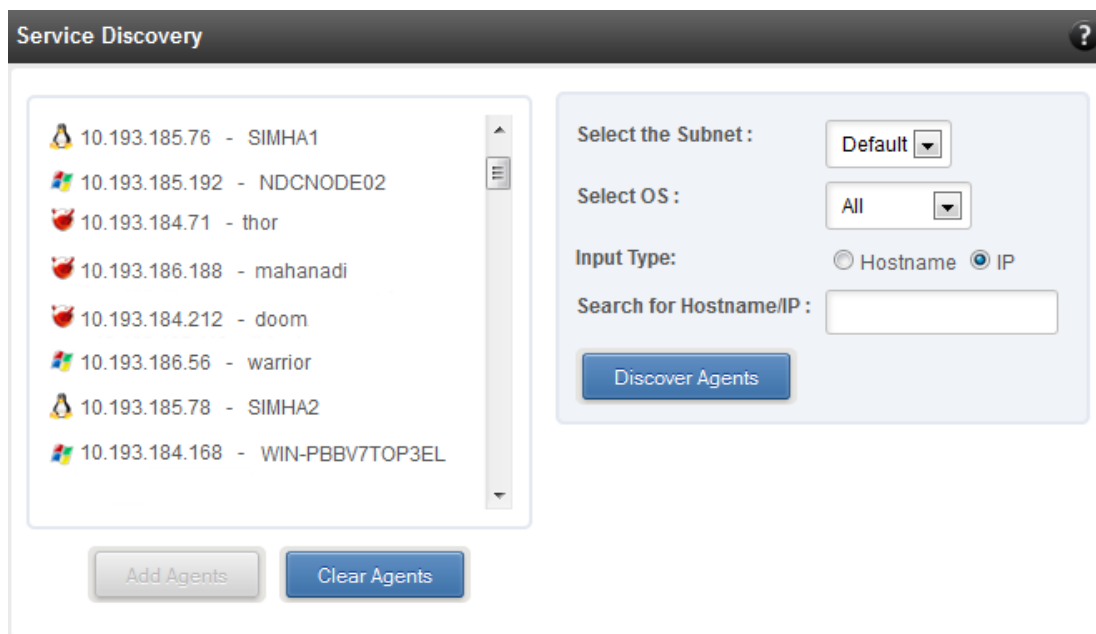
Figure 141 - Services Discovery module

Bulk Driver Installation

This module allows you to install drivers for multiple systems simultaneously. Drivers available for installation for a particular system may differ depending on the network adapter (T5, T4 or T3) and operating system selected.

- **Installing Driver**
- i. In the **Choose the card** fields, select T3 or T4/T5 depending on the chip revision of the network card.
- ii. Select the operating system for which drivers are to be installed in the **Choose the OS Type** field. All the systems with the selected operating system will be displayed in the list below.
- iii. Select a system or systems from the list and choose the driver to be installed in the **Driver Installation** section.
- iv. Download the appropriate driver from Chelsio's Download Center, service.chelsio.com.
- v. Locate the driver package.
- vi. Click **Install** button to install the driver.

Figure 142 - Bulk Driver Installation module

ⓘ Note   *Agents that report errors or whose passwords are not stored, will be automatically skipped during the driver installation.*

8.4.2. Add System

Use this option to connect to new Agents using their IP or Hostname. The TCP port for connection is by default 35001. You will have to provide correct user credentials for the agent in order to connect successfully.

After connecting to the Agent, the menu bar on the left will display the connected system and its related modules.

If you deselect the 'Remember Password' option, you will be asked to enter the password every time you try to connect to the system.
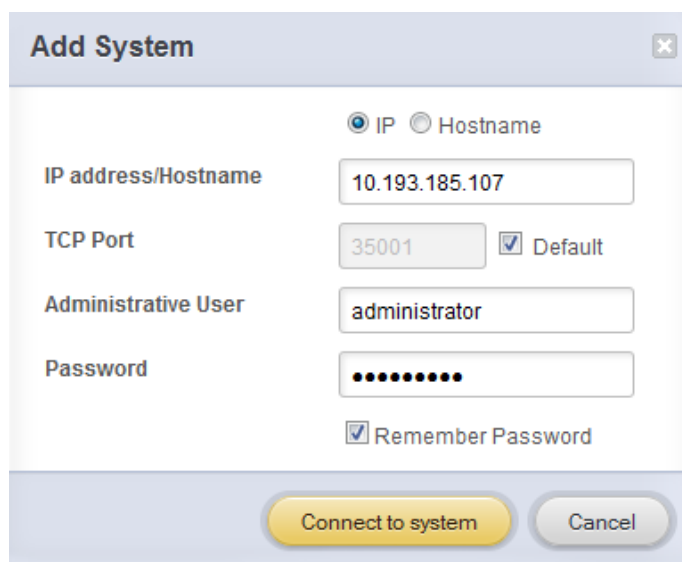
Figure 143 - Adding a UM Agent

8.4.3. Remove System

Use this option to disconnect an Agent. To remove an agent, click on the name of the system in the tree menu in the left and click *Remove System*. Then click *Yes* to confirm.
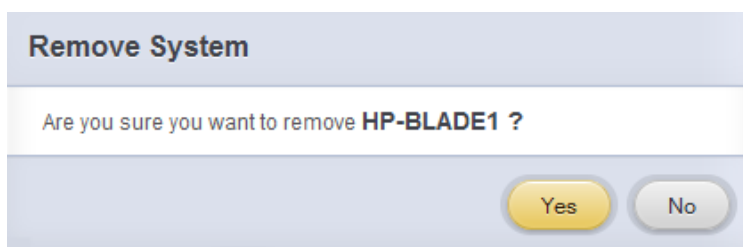
Figure 144 - Removing a UM Agent

8.4.4. Refresh

This option can be used to reload the Web GUI or UM Agent.

To reload the Web GUI, navigate to the Home page (by clicking on the "Home" button and click Refresh. You can use this option to refresh Home page panes (Bookmarks and History, Service Discovery and Bulk Driver Installation).

To reload an Agent, click on the name of the system in the tree menu in the left and click "Refresh". You can use this option to update any changes made to system settings like load/unload drivers.

8.4.5. Subscribe (Email Alerts)

This feature is available only on the Web Management Interface.

Using this option, you can receive email alerts regarding the link status of a Chelsio Network Interface Card. This feature sends email notifications regarding the port and the card, on which the link up/down event has occurred. Not only can you configure multiple email addresses to receive notifications, but also customize the email id of the sender for troubleshooting purposes.

To subscribe to **Email Alerts,** enter the sender's email address in the *Email address* field. It should be in the format of <name>@<domain>.<extension>. You can enter multiple email addresses for the *Recipients* field separated a comma. Enter Mail server details and ensure that the "Enable email Alerts" field is enabled. Select the Agent(s), for which you want to receive alerts and Click on **Save.**

Figure 145 - Subscribing to Email Alerts

8.4.6. Bulk Configuration

The **Bulk Configuration** page allows you to execute common configuration changes to multiple agents and their network adapters simultaneously. You can conveniently perform bulk operations like installing option ROM, setting MTU and VLAN ID, changing adapter and port parameters on various devices, without having to access multiple modules and thus saving considerable amount of administration time.

Various configurable parameters have been categorized into several modules like **Boot Configuration** module to install and erase option ROM, **Network Configuration** module to set MTU and VLAN ID, **Card Configuration** module to change driver parameters, etc.

Before accessing these modules, you will have to create *groups* and then add *members* to that group. Once done, you can select the group in the modules and the new setting will be applied to all members of that particular group.

◉ Manage Groups

This is where you can add, delete and manage groups. Use the **Create a Group** section to create a group by specifying agent's platform and group type. There are various types of groups to choose from depending on the type of configuration setting you want to change. For example, to change the MTU size of a network interface (in the Network Configuration module), create a group with group type *Network*. To install or erase option ROM on a Chelsio T4 adapter (in the Boot Configuration module), create a group with group type *t4adapter*.

Here is a list of available configuration modules and corresponding group type:

- **Boot Configuration:t3adapter, t4adapter, t5adapter**
- **Network Configuration: Network**
- **Card Configuration: t3adapter, t4adapter, t5adapter**
- **Port Configuration:t3port, t4port, t5port**
- **Bypass Configuration: Network**

After the group has been created, add members to that group using the **Add a member row** button. Depending on the group type selected, you may be asked to provide additional details for the new member. Use the **Fetch Details** button to do so and finally click **Save a member** button to add the member to the group.

To delete a group, select it from the **Select a Group** drop-down list, and click **Delete Group.** To delete a member from a group, select the group to which the member belongs, select the radio button corresponding to the member to be deleted in the **SNO** field and finally click **Delete Member(s).**

Figure 146 - Creating a group

Figure 147 - Managing a group

- **Boot Configuration**

Using this module, you can install option ROM or erase option ROM on Chelsio network devices. The **Set Default Boot Settings** button will reset the adapter to factory boot settings.

Figure 148 - Boot Configuration module

Network Configuration

In the **Network Configuration** module, you can set Maximum Transfer Unit (MTU),Virtual LAN (VLAN) ID and change the IP address type for the members (network interfaces) of the *Network* group. MTU can be set between 1500-9000 bytes. VLAN id can be set for an adapter within the range 0-4094 (enter 0 to disable it). The IP type can be static or DHCP.

Figure 149 - Network Configuration module

Card Configuration

The **Card Configuration** module allows you to set various adapter settings including TCP Offload. Offload settings are only available when using the TOE capable drivers (*t3_tom* and *toecore* for T3 cards; *t4_tom* and *toecore* for T4 cards).

Figure 150 - Card Configuration module

- **Port Configuration**

In the **Port Configuration** module, you can set various port settings like enabling Tx checksum and TCP segmentation offload, setting Link speed and link duplex mode, etc. The settings depend on the device driver installed.

Figure 151 - Port Configuration module

Bypass Configuration

Use the **Bypass Configuration** module to configure Chelsio's bypass adapters like B420-SR and B404-BT.

Figure 152 - Bypass Configuration module

8.5. System page

The system page is displayed, when the system hostname / IP address is selected in the tree menu on the left. On adding a system, this item is automatically selected, and this page is displayed. The system page contains generic system and support modules which are discussed below:

8.5.1. System summary

This module lists the system Hostname, Operating System, platform and also gives the count of the Chelsio cards found.
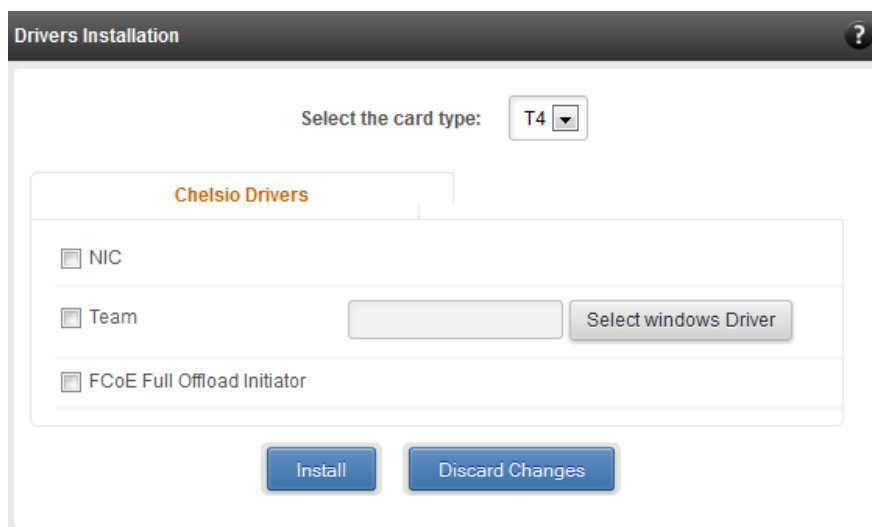
PROPERTY	VALUE
Hostname	shambu-dc
Connected IP:Port	10.193.184.168:35001
Chelsio Cards	1
Operating system	Windows Server 2012
Platform	x86_64 (amd64 / x64)

Figure 153 - System Summary module

8.5.2. Drivers Installation

Using this module, one can install various Chelsio drivers for different operating systems.

Figure 154 - Drivers Installation module connected to Windows Agent

8.5.3. Driver Details

A list of Chelsio device drivers with related information like driver description, version, current load status and installation date is shown in this module. To load or unload a particular driver, select the appropriate option (Yes to load, No to unload) in the corresponding cell of the *Loaded* column and click **Load/Unload Driver** button. Click **Refresh** if changes are not reflected immediately. To reject the load/unload option selected, click **Discard Changes**.

Figure 155 - Driver Details module

8.5.4. System Diagnostics

Using this module, you can run various diagnostic tests on Chelsio adapters to troubleshoot adapter related issues. Select the adapter(s) from the list for which you want to run the test, select the operation (type of test; you can run more than one test at a time) and click **Run Test**. After the tests are completed, the results will be displayed in a tabular format.

Figure 156 - System Diagnostics module for a T4 CNA

8.5.5. Unified Wire Manager Component Versions

A list of the Unified Wire Manager agent components installed on the managed system is shown in this module. The versions of the components are useful in case of reporting an issue to support.

COMPONENT	VERSIONS
Server	2.4.29
msiscsi	1.1.6
winchiscsi	2.0.0
win_net	2.0.0
winteam	2.0.0
winhwlib	2.0.3
winosapi	2.0.0

Figure 157 - Unified Wire Manager Component Versions module

8.5.6. VM Configurations

The VM Configurations module allows you to view UUID and Power and Health state of Virtual Machines. You can perform various system power options like enable (start VM), disable (stop VM), reboot (restart VM), pause and suspend (save VM state).

You can perform similar actions on multiple virtual machines. To do so, click on the machine names in the list. The properties box will display the domain state of the machines selected. Now, click on any of the system power actions provided at the bottom.

Virtual Machine Properties

RHEL6.1

UUID : 77C5D36F-F482-4EAE-A7F9-420D956C82C0

Power State : Enabled

Health State : OK

Enable Disable Reboot Pause Suspend
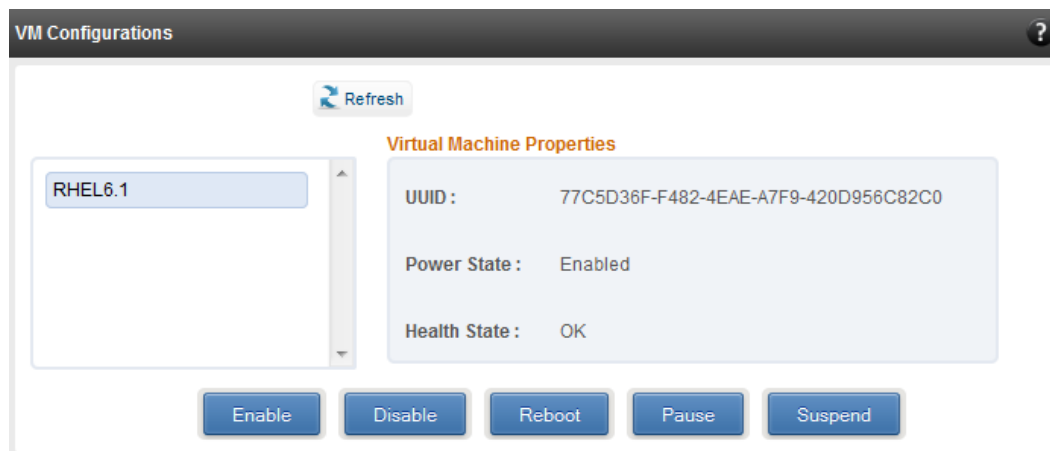
Figure 158 - VM Configurations module

8.5.7. Managed system application logs

The management agent logs its activities and any errors that occur, in */var/log/chelsio* in *Linux* and FreeBSD and in the Event log, in Windows. This log can be obtained in this module. Only 20 entries can be obtained and viewed at a time. Logs can be viewed by either choosing from a list of fixed range or by specifying a custom starting point.

Use the **Get Logs** button to retrieve, and **Hide Logs** button to clear the log entries. The **Delete Logs** button will remove the logs permanently from the agent.
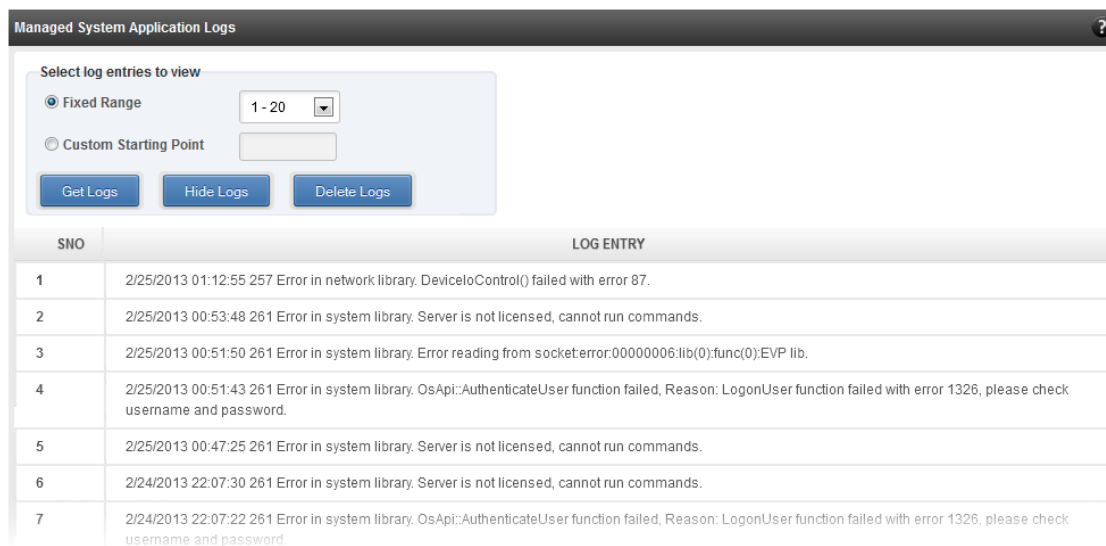
Figure 159 - Managed System Application Logs module for Windows Agent

8.6. Network page

8.6.1. Network summary

The **Network Summary** module provides the total number of Chelsio adapters present, including the number of T5, T4 and T3 adapters. It also provides the total number of Network interfaces including corporate and Chelsio interfaces and VLANs.

Figure 160 - Network Summary module

8.6.2. Chelsio card page

When a Chelsio card is selected in the tree menu on the left, this page is displayed. It provides details of the card and associated settings. It also displays any card specific statistics that the hardware provides. The modules available on this page are as below:

Card summary

This module provides PCI, firmware and other details of the card. The card's serial number and factory MAC address are also provided for inventory purposes.

Card Summary	
PROPERTY	**VALUE**
PCI ID (Vendor : Device)	1425 : 4403
PCI bus location (Bus : Device : Function)	01:00:04
Card Serial Number	PT18111222
Factory MAC address	00:07:43:10:71:80
Firmware Version	1.8.3.0
Ethernet Ports	4
Offload Support	Chelsio T440-CR 10G Ethernet Function Enumerator
In maintenance mode	No
Hardware is TCP Chimney offload enabled	No
Connector	10G SFP+

Figure 161 - Card Summary module for a Windows Agent

TCP Offload settings (Linux & FreeBSD)

The TCP offload settings applicable to the card are shown here. These settings are only available when using the TOE capable drivers (*t3_tom* and *toecore* for T3 cards; *t4_tom* and *toecore* for T4 cards). On changing the settings, the changed settings may not reflect immediately on refreshing the data. Highlight the system item in the tree menu on the left, and click "Refresh", to refresh data from the system, in case the updated settings are not being shown.

Figure 162 - TCP Offload Settings module for a FreeBSD Agent

Device Driver settings (Windows)

The device driver settings applicable to the card are shown here. For Chelsio T5 and T4 adapters, only the *MaxVMQueues* field will be displayed. On changing the settings, the changed settings may not reflect immediately on refreshing the data. Highlight the system item in the tree menu on the left, and click **Refresh**, to refresh data from the system, in case the updated settings are not being shown.

Figure 163 - Device Driver Settings module for a Windows Agent

Card statistics

Certain statistics are maintained on a per card basis (instead of a per port basis), since the card has a TCP/IP offload capability. The statistics are for TCP and IP protocol processing done in the card's hardware. These statistics may only be applicable if the card is TOE enabled.

Figure 164 - Card Statistics module

8.6.2.1. Chelsio card's port

The port page is displayed on selecting a port of a Chelsio card listed in the tree menu on the left. It provides details of the port and port settings. It also displays any port specific statistics that the hardware provides. The modules available on this page are as below:

○ Port summary

The port details such as the Ethernet adapter name and link details are shown in this module.

Figure 165 - Port Summary of T4 CNA

○ Port settings

Port settings such as MTU, Link speed and others can be set in this module. The settings depend on the device driver installed.

Figure 166 - Port Settings of T4 CNA

- **Port statistics**

Ethernet statistics and additional hardware statistics for the port are displayed in this module.

Port Statistics	
STATISTIC	**VALUE**
tx_octets	0
tx_frames	0
tx_bcast_frames	0
tx_mcast_frames	0
tx_ucast_frames	0
tx_error_frames	0
tx_frames_64	0
tx_frames_65_127	0
tx_frames_128_255	0
tx_frames_256_511	0
tx_frames_512_1023	0
tx_frames_1024_1518	0
tx_frames_1519_max	0
tx_drop	0

Figure 167 - Port Statistics of T4 CNA on Windows Agent

8.6.3. Networking Management page

The system networking configurations are shown on this page. IP addresses, MTU, VLAN Ids, DNS and default gateway settings can be viewed and modified here. Network adapters can also be enabled or disabled as required. The modules available on this page are as below:

- **System Network configuration**

The list of network adapters on the system is displayed in a list on the left. The icon for the adapter indicates whether it is administratively enabled and if it is connected to the network. The primary IP address (IPv4) can be set for the adapter, when it is selected. There is an option to add/modify/delete additional IP addresses or aliases for the specified adapter. Use the option to add additional IP addresses with caution, since multiple IP addresses configured on the same adapter, for the same network, may result in unpredictable behavior of the system's networking

stack. Maximum Transfer Unit (MTU) can be set between 1500-9000 bytes. VLAN id can also be set for an adapter within the range 0-4094 (enter 0 to disable it).

You can use the View/Set IP addresses option to add, modify or delete IP aliases.
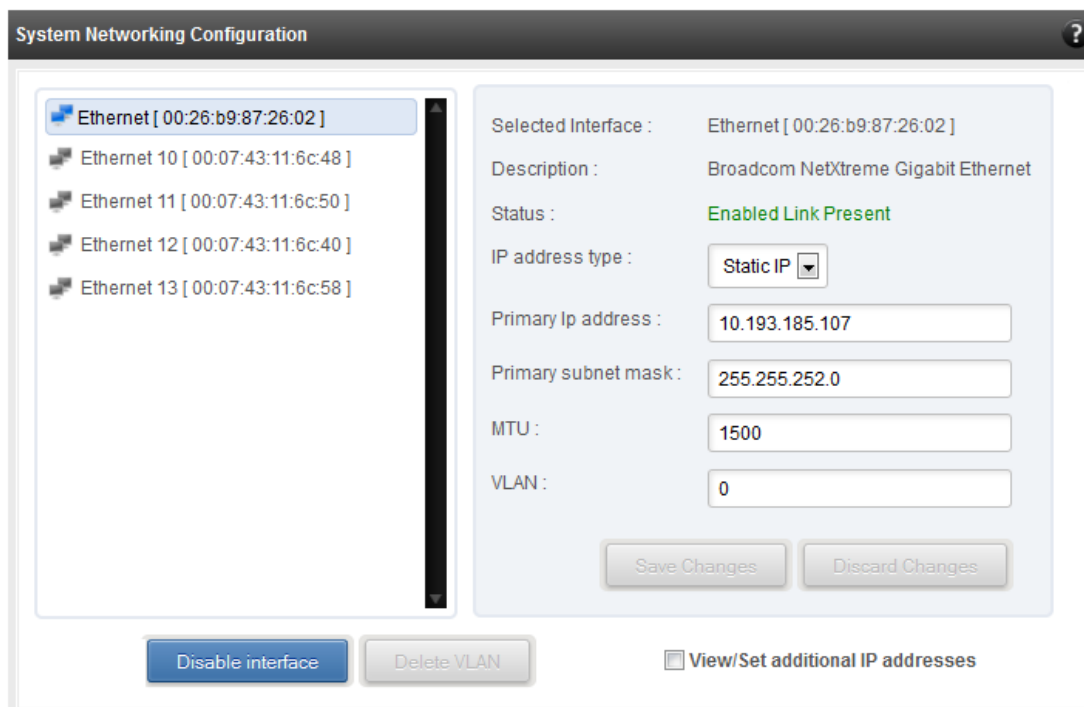
Figure 168 - System network configuration module
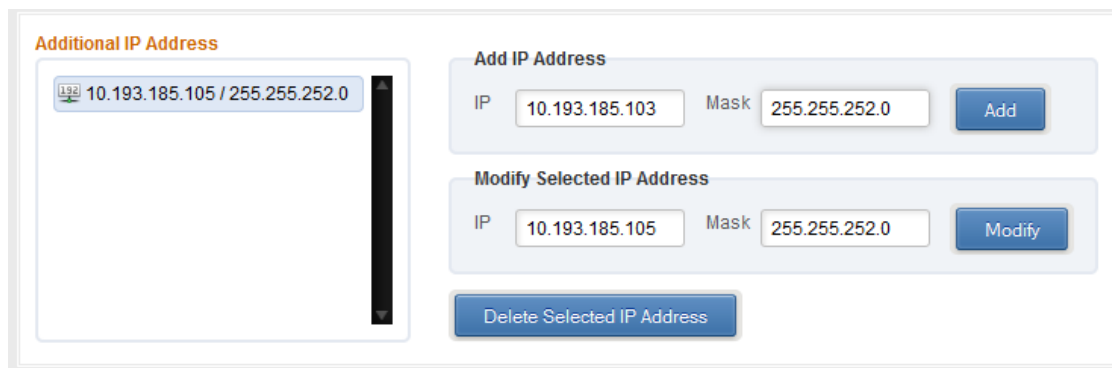
Figure 169 - Managing IP aliases

System network statistics

Using this module, one can generate reports based on Throughput pkts/sec and Throughput Mbs (Receive, Transmit, Bi-direction) in Table and Graph format for a network adapter. A report for hardware statistics can be generated based on different parameters, only in the Table view in the **Advanced NIC characteristics**. The **polling time** field sets the average time (in seconds) based on which the table/graph updates the report.
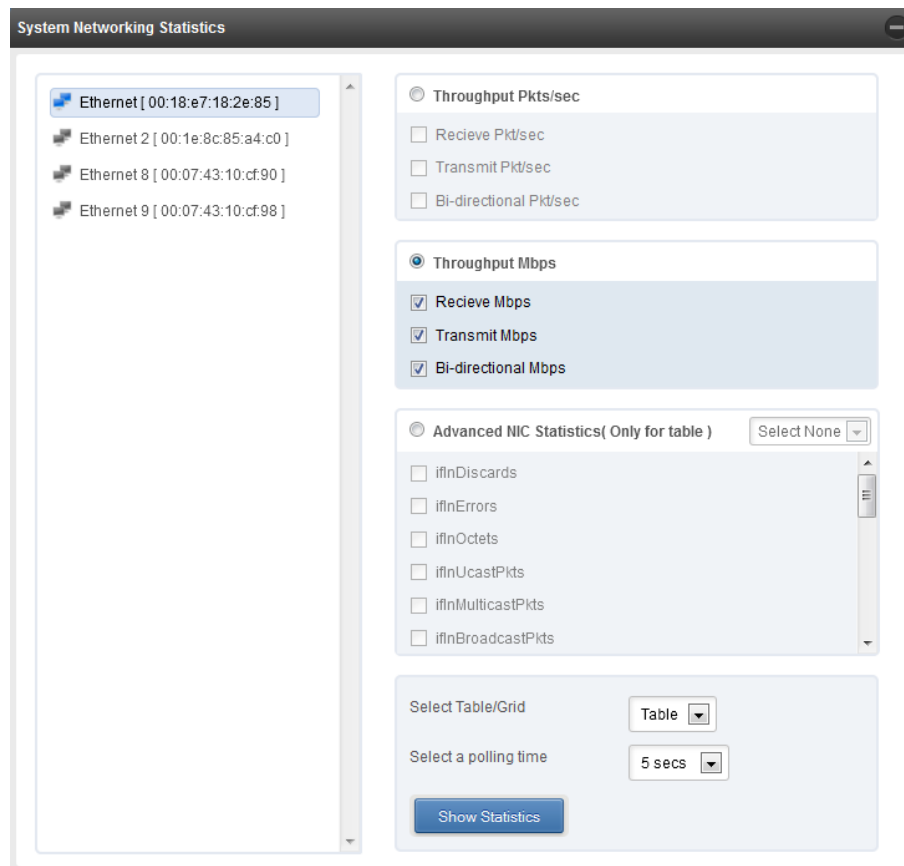
Figure 170 - System network statistics module
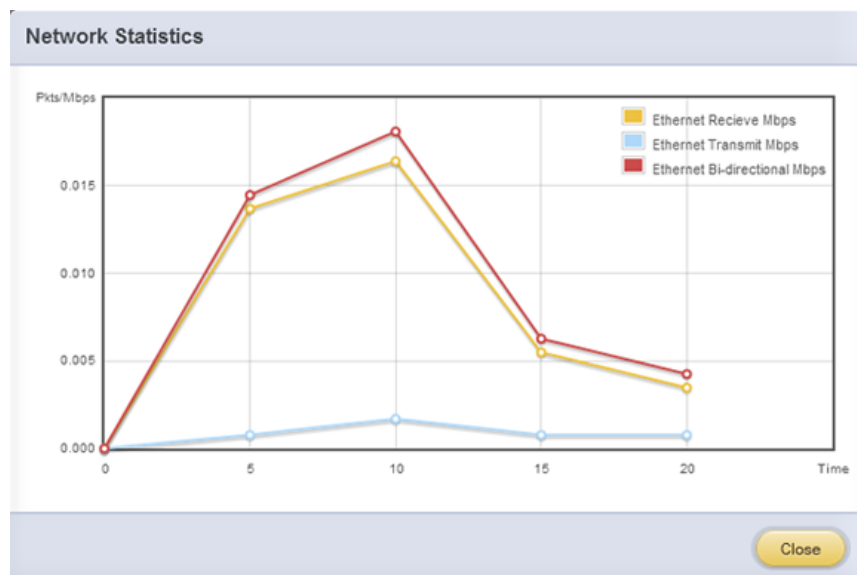
Figure 171 - Network Throughput Vs Time instant Graph

Figure 172 - Network Throughput Vs Time instant Table

● Default Gateway and DNS configuration

The DNS servers list can be set here. The default gateway for remote networks and the Internet can also be set here. On Linux and FreeBSD, only one default gateway is allowed. On Windows, you may set multiple default gateways. Use the option to set multiple default gateways with caution, since it may cause the system to stop communicating with external networks.

Figure 173 - Default gateway and DNS configuration module for a Windows Agent

● Create a network team/bond device (Linux and FreeBSD)

A list of regular network adapters is provided here, to create a Network Team / Bond device. The available modes for the team depend on the OS teaming / bonding driver in use. On Linux the

team may be created with a DHCP or Static IP address. Please check with the driver documentation for the supported modes for creating a team / bond, with offload enabled Chelsio cards. All modes may not be available with all configurations / combinations. Also, the team members can only be 2 ports of a single offload-enabled card, and not across Chelsio cards. Do not mix third party cards and offload-enabled Chelsio cards in a single team.

Figure 174 - Create a network team/bond device module for Linux Agent

Network troubleshooting

This module allows detecting and troubleshooting various network connectivity issues. The Ping utility helps to contact a system by specifying IP address, Number of ICMP packets to send and packet timeout. The result of the ping can be viewed by clicking on the **Ping Result** button.

Using **TraceRoute** one can determine the route taken by packets across an IP network.

Use the **GetConnections** utility to view currently active TCP/UDP connections. Offload status for each connection is also displayed if protocol offload hardware is available. This is useful for troubleshooting any connectivity issues for clients to various services.

Ping

Destination :	10.193.190.140
No. Of ICMP Packets to send :	4
ICMP Packet timeout in seconds :	5

Pkts Send: 4 , Pkts Recvd: 4 , AVgRtt: 5 ms Clear

Figure 175 - Ping Utility

TraceRoute

Destination : www.chelsio.com

Hop Count	Round Trip Time	Ipv4 Address
1	2 ms	10.193.184.1
2	0 ms	10.193.177.3
3	5 ms	111.93.129.157
4	8 ms	121.241.196.101
5	3 ms	121.240.1.242
6	23 ms	172.29.250.33
7	24 ms	180.87.38.5
8	131 ms	80.231.217.17
9	130 ms	80.231.217.6
10	131 ms	80.231.154.17
11	132 ms	208.178.58.109
12	274 ms	208.178.63.114
13	272 ms	72.13.84.18

Hide

Figure 176 - TraceRoute Utility

GetConnections

GetConnections

PROTOCOL	LOCAL ADDRESS	REMOTE ADDRESS	STATE	OFFLOAD
TCP	0.0.0.0:135	0.0.0.0:0	Listening	In host
TCP	0.0.0.0:445	0.0.0.0:0	Listening	In host
TCP	0.0.0.0:3389	0.0.0.0:0	Listening	In host
TCP	0.0.0.0:35001	0.0.0.0:0	Listening	In host
TCP	0.0.0.0:47001	0.0.0.0:0	Listening	In host
TCP	0.0.0.0:49152	0.0.0.0:0	Listening	In host
TCP	0.0.0.0:49153	0.0.0.0:0	Listening	In host
TCP	0.0.0.0:49154	0.0.0.0:0	Listening	In host
TCP	0.0.0.0:49155	0.0.0.0:0	Listening	In host

Figure 177 - GetConnections Utility

8.6.3.1. Hypervisor

● Virtual Network Manager

This module allows you to view and manage virtual networks. The left pane displays a list of different virtual networks created. Clicking on a virtual network name will display related properties on the right.

If a virtual network is added to a virtual machine, a "+" link appears. Expanding the link will display the virtual machines to which the network is attached. Click on the virtual machine names to view their properties on the right.

To delete a virtual network, click on the network name and then click "Delete Switch". If it is attached to a virtual machine, you will have to detach the virtual machine first. To do so, click on the virtual machine and click "Detach". Similarly, detach all the virtual machines and then use the "Delete Switch" to delete the virtual network.
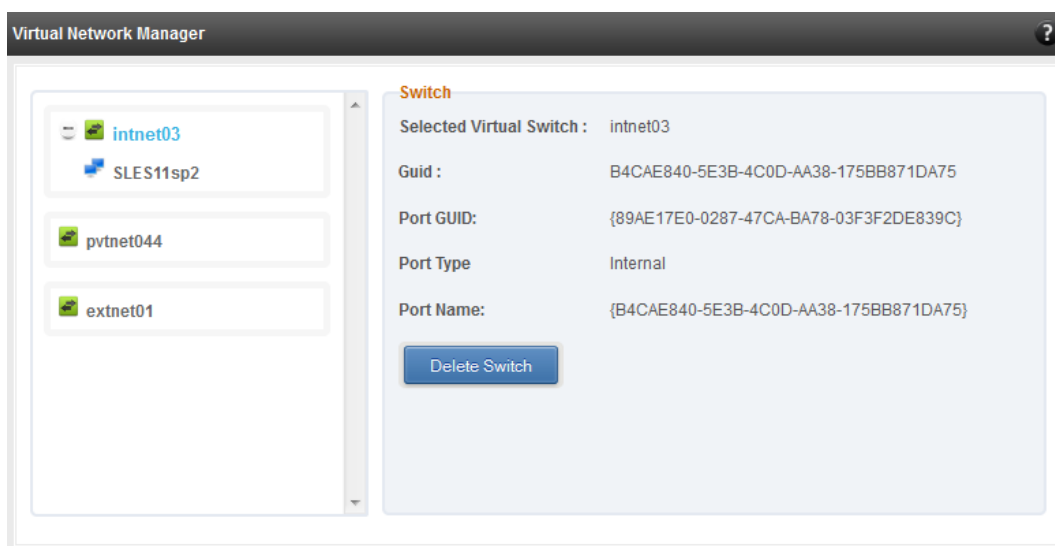
Figure 178 - Virtual Network Manager module

● Add Virtual Network Configuration

There are three kinds of virtual networks you can create using this module:

- External network: Using this type, you can provide virtual machines access to external networks and vice versa via a physical network adapter in the host system. The virtual machines can also communicate with each other on the same virtual network.
- Internal Network: This type allows communication between virtual machines in the same virtual network and also between the virtual machines and the host. This type of virtual network is not bound to any physical network adapter and no access to external networks is provided.

- Private Network: A Private Network is similar to Internal Network in that physical adapter is not required for setup and access to external networks is not provided. However, unlike Internal Network, guest operating systems can only communicate with guest operating systems in the same private network and not with the host. The host operating system cannot access the virtual machines on private network.

Once created, you can manage the virtual networks in the **Virtual Network Manager** module.
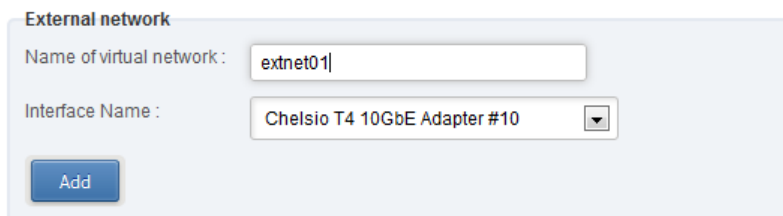
Figure 179 - Creating external virtual network
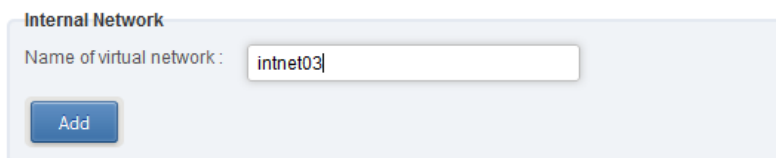
Figure 180 - Creating internal virtual network
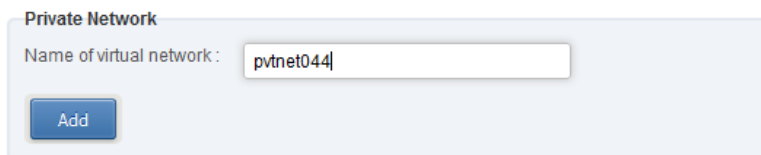
Figure 181 - Creating private virtual network

Virtual Network Settings

To attach a virtual network to a virtual machine, select the virtual network from the **Virtual Network** list and the virtual machine from the **VM** list. Finally click *Attach*.
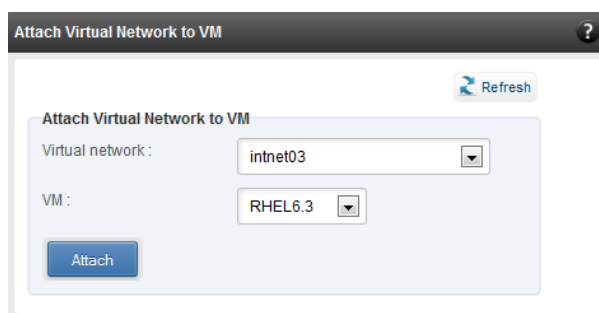
Figure 182 - Attaching Virtual Network to VM

8.6.4. iWARP

iWARP Settings

On Linux Agents, iWARP parameter settings for Chelsio's RDMA capable NICs can be set using this module. These settings can be set only when iWARP driver (*iw_cxgb4* for T4 and T5; *iw_cxgb3* for T3) is loaded. If you set any parameter for a T5 adapter, it applies for all the T5 adapters present. Same applies for T4 and T3 adapters.

On Windows Agents, only T3 HBAs are supported currently. Parameters can be set per port.

On FreeBSD Agents, only T4 CNAs are supported. iWARP parameter settings can be set only when *iw_cxgbe* driver is loaded.

iWarp Settings	
Select a Driver:	iw_cxgbe ▾

Save Changes   **Discard Changes**

DESCRIPTION	VALUE
peer2peer	No ▾
ep_timeout_secs	60 ⬍
mpa_rev	1 ⬍
markers_enabled	No ▾
crc_enabled	Yes ▾
rcv_win	262144 ⬍
snd_win	131072 ⬍
db_delay_usecs	1 ⬍
ocqp_support	Yes ▾
db_fc_threshold	2000 ⬍
fastreg_support	No ▾
dack_mode	1 ⬍
c4iw_max_read_depth	8 ⬍
enable_tcp_timestamps	No ▾
enable_tcp_sack	No ▾
enable_tcp_window_scaling	Yes ▾
c4iw_debug	Yes ▾
p2p_type	1 ⬍

Figure 183 - iWARP settings for T4 CNA for FreeBSD Agent

Figure 184 - iWARP settings for T4 CNA for Linux Agent

8.7. Storage

Storage Summary

The **Storage** module lists the status of configuration modules under Storage section, running on the agent.

Figure 185 - Storage Summary Module

8.7.1. FCoE Initiator (Linux, Windows, XenServer)

All supported Chelsio FCoE initiators available on the operating system can be managed from this page. FCoE support is extended on Linux, Windows and XenServer platforms. Please refer Platform/Driver Matrix section on the list of operating systems that are supported.

● FCoE Initiator Summary

This module provides details about the driver installed; such as driver name and its version. The module also gives information about the number of FCoE enabled cards that are present on the machine.
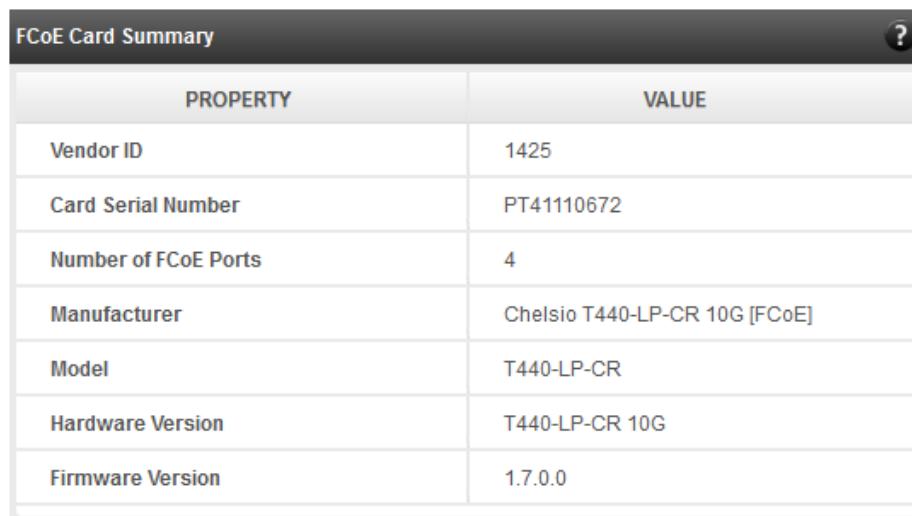
FCoE Initiator Summary	
PROPERTY	VALUE
FCoE Driver	csiostor (1.1.0.9)
No. of FCoE enabled cards	2
No. of FCoE Ports	6

Figure 186 - FCoE Initiator Summary module for Linux Agent

8.7.1.1. FCoE Initiator Card

● FCoE Card Summary

Details pertaining to the card used such as model, firmware/hardware version etc, are provided in this module.

FCoE Card Summary	
PROPERTY	VALUE
Vendor ID	1425
Card Serial Number	PT41110672
Number of FCoE Ports	4
Manufacturer	Chelsio T440-LP-CR 10G [FCoE]
Model	T440-LP-CR
Hardware Version	T440-LP-CR 10G
Firmware Version	1.7.0.0

Figure 187 - FCoE Card Summary module

○ **FCoE Attributes**

Information such as Interrupt modes (MSI/MSI-X/INTx), SCSI mode and the card state are provided in this module.

Figure 188 - FCoE Attributes module

8.7.1.2. FCoE Port

This is an actual N_Port which communicates with the fabric and performs FIP and FCoE device discovery. This page lets the user to retrieve all the FCoE specific port information and also extend NPIV management support. It contains the following sections:

○ **FCoE Port Summary**

The SCSI adapter name and the underlying ENODE MAC address of the physical port can be found here.

Figure 189 - FCoE Port Summary module for Linux Agent

○ **FCoE Port Attributes**

This module provides details about link status and port identifiers such as WWPN, WWNN, FC ID and NPort MAC Address. The module also contains fabric information such as fabric name, VLAN on which the FCoE service is currently running and the number of SCSI targets that are being discovered by this port. Port speed being mentioned in this section varies on the card type (10G/1G) being used. Note that only class 3 service is supported by the initiator for now and the frame size is fixed to 2128 bytes as per spec.

Figure 190 - FCoE Port Attributes module for Linux Agent

○ **FCoE NPIV management**

NPIV is a fibre channel facility allowing multiple N_Port IDs to share a single physical N_Port. This module allows the user to manage virtual ports on the corresponding FCoE Port.

To create a virtual port, select the option **Create** and the GUI allows two ways of creating a virtual port.

i. Manual: Where the user can manually create a virtual port by providing a value to the WWPN and WWNN fields.
ii. Auto-generate: Where the FCoE function auto-generates a WWPN and WWNN for the virtual port.

To delete a virtual port, select the option **Delete** and select the virtual port WWPN which you want to delete and click on **delete**.

Figure 191 - FCoE NPIV management module

8.7.1.3. FCoE Remote Port

Remote ports are the SCSI targets that are discovered by their respective N_port/virtual ports. The GUI conveys the same via a tree structure so that the end user knows the initiator-target mapping.

● FCoE Remote Port Attributes

This module provides details about the discovered target such as target's FC ID, WWPN and WWNN so that the user can identify the discovered target accordingly.

Figure 192 - FCoE Remort Port Attributes module

FCoE Remote Port Lun Details

This module provides the LUN information such as size of the LUN, SCSI address, and LUN address. For Linux, the SCSI address is displayed in H:C:T:L (Host:Channel:Target:Lun) format and for Windows, it is displayed in P:B:T:L(SCSI Port:Bus:Target:Lun) format.

Figure 193 - FCoE Remote Port Lun Details module

8.7.1.4.   FCoE Virtual Port

A virtual port allows multiple Fibre Channel initiators to occupy a single physical port, easing hardware requirements in SAN design, especially where virtual SANs are called for. The virtual ports appear under their respective N_Ports after creation and the GUI conveys it via a tree structure so that the end user knows the N_port-VN_Port mapping. It contains the following modules:

FCoE Virtual Port Summary

The SCSI adapter name and the underlying ENODE MAC address of the physical port can be found here.

Figure 194 - FCoE Virtual Port Summary module for Linux Agent

- **FCoE Virtual Port Attributes**

The module provides details about link status and port identifiers such as WWPN, WWNN, FC ID and Virtual NPort MAC Address. The module also contains fabric information such as fabric name, VLAN on which the FCoE service is currently running and the number of SCSI targets that are being discovered by this virtual port. Port speed being mentioned in this section varies on the card type (10G/1G) being used. Note that only class 3 service is supported by the initiator for now and the frame size is fixed to 2128 bytes as per spec.

PROPERTY	VALUE
State	Operational
NodeWWN	50:00:74:30:46:3A:71:09
PortWWN	50:00:74:30:46:3A:71:89
NPort MAC Address	0E:FC:03:77:00:1D
Vlan Id	5
Fabric Name	20:05:00:05:73:D5:7A:C1
Nport ID	77:00:1D
Type	VN_Port
Supported Class Of Service	3
OS Device Name	/sys/class/fc_host/host127
Speed	10 GBPS
Maximum Frame Size	2128
No. of SCSI Targets	1

Figure 195 - FCoE Virtual Port Attributes module

- **FCoE Remote Port Attributes**

This module provides details about the discovered target for remote port associated with virtual port. Details such as target's FC ID, WWPN and WWNN are provided so that the user can identify the discovered target accordingly.

PROPERTY	VALUE
FC ID	54:00:53
State	Operational
NodeWWN	20:01:00:11:0D:56:29:00
PortWWN	20:01:00:11:0D:56:29:00

Figure 196 - FCoE Remort Port Attributes module

FCoE Remote Port Lun Details

This module provides LUN information for remote port associate with virtual port. Details such as size of the LUN, SCSI address, and LUN address are provided. For Linux, the SCSI address is displayed in H:C:T:L (Host:Channel:Target:Lun) format and for Windows, it is displayed in P:B:T:L(SCSI Port:Bus:Target:Lun) format.
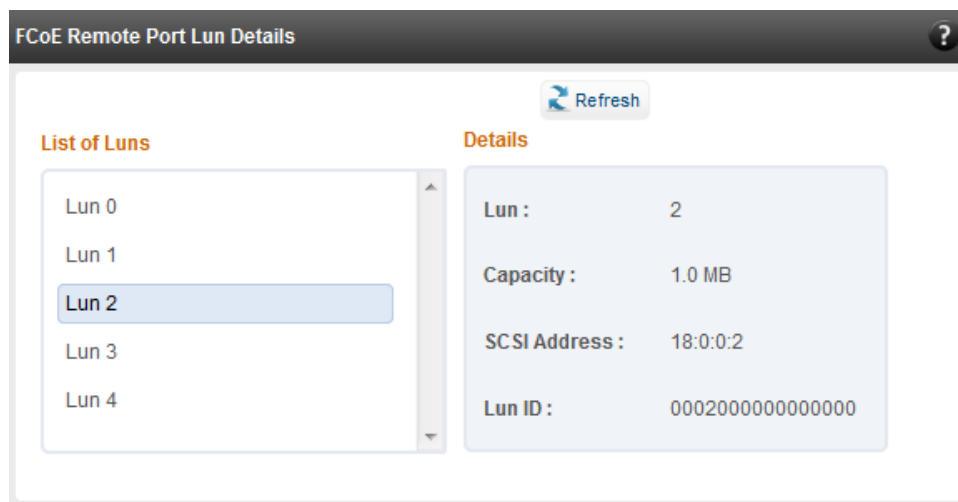
Figure 197 - FCoE Remote Port Lun Details module

8.7.2. iSCSI initiator (Linux, Windows)

All supported iSCSI initiators can be managed from this page. The supported initiators on Windows are Microsoft and Chelsio iSCSI initiator (T4 adapters). On Linux, Open iSCSI initiator is supported. The modules available on this page are:

Initiator nodes

This module lists the initiator nodes / virtual adapters configured in the initiator stack. The node can be enabled or disabled (Chelsio node cannot be disabled in Windows), and its properties can be viewed and edited in this module. In the Chelsio Linux stack, new initiator nodes can be created too. Disabling the initiator causes it to log out of any iSCSI targets that it is connected to, thus removing any disks provided by the iSCSI targets that were connected. Use the **Disable** option with caution. The CHAP authentication secret should be between 12 and 16 characters in length, and the initiator's IQN name should start with "iqn.".
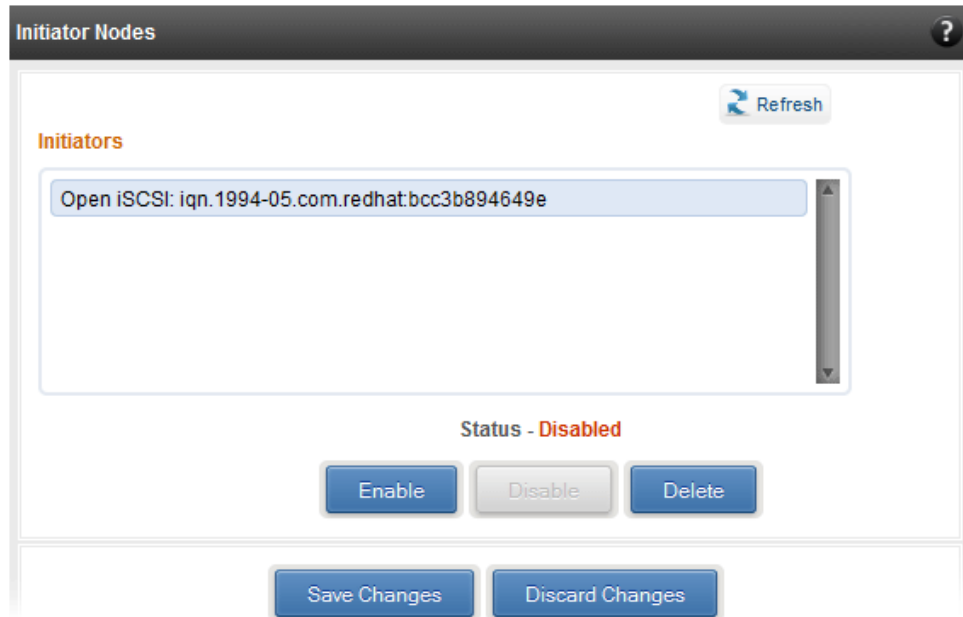
Figure 198 - Open iSCSI initiator

Figure 199 - Microsoft iSCSI initiator

Figure 200 - Chelsio iSCSI initiator

- **Discover targets**

iSCSI targets can be discovered by providing the IP address and TCP port (usually 3260) of the target. The discovery operation fetches the targets found at that Portal (combination of IP address and TCP port). The discovery operation also fetches all the other Portals that the target(s) are listening on. The discovered target can be deleted if required. Please note that all the Portals that the target sent are listed. The delete operation will not work on all the portals, only on the original discovery portal (the IP address and TCP Port specified when discovering the target).

ⓘ Note   *If there are any pre-existing iSCSI sessions established to the target, deletion of the target Portal from the discovered targets list will fail.*
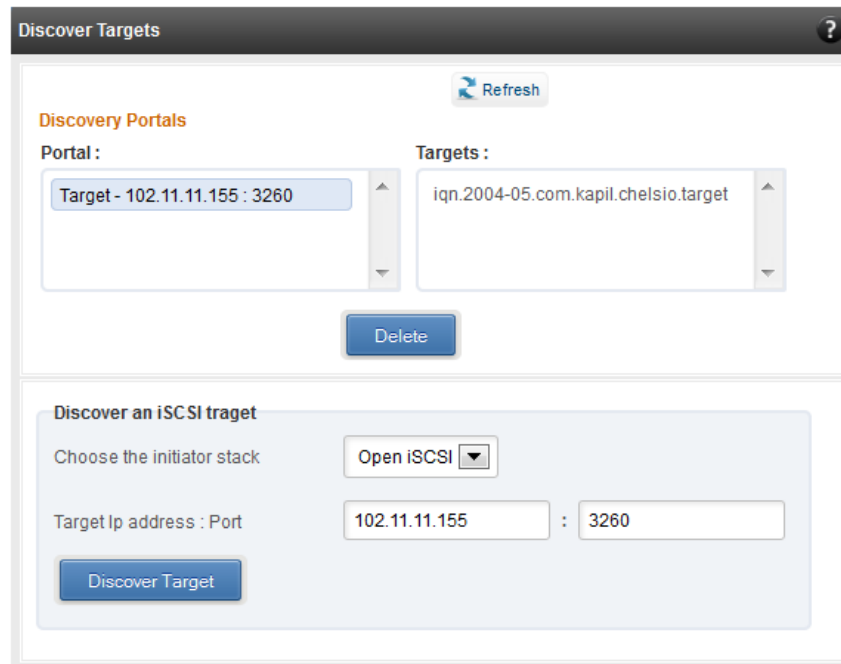
Figure 201 - Discover targets module

- **Targets**

The iSCSI targets that have been discovered, or are currently connected, are listed here. You may login, logout and delete the target from the initiator's configuration. If a target is connected, the sessions and connections to the target, and the disks provided by the target will be listed.
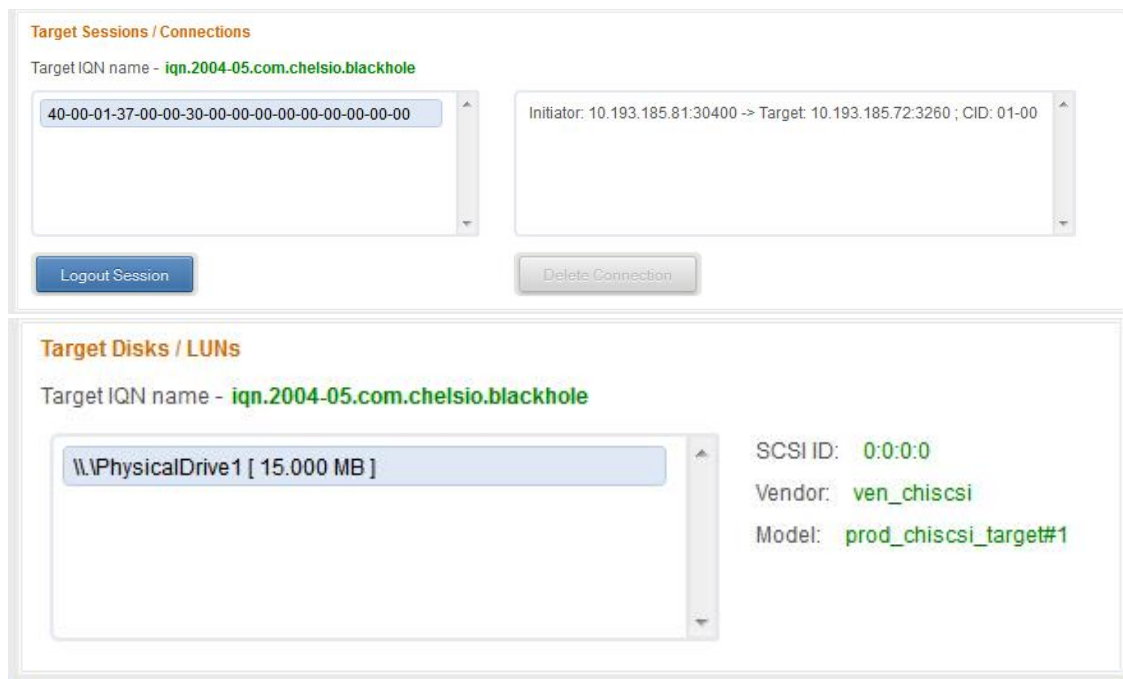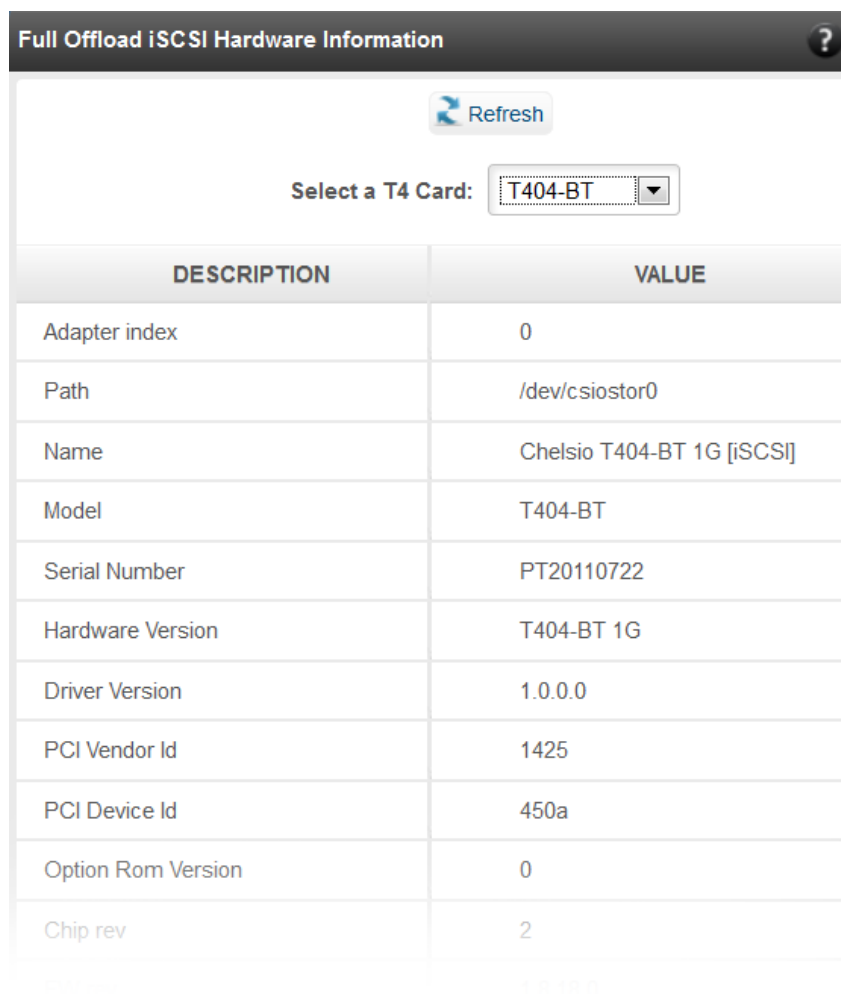
Figure 202 - Targets module after logging in

8.7.3. FO iSCSI Initiator (Linux)

● Full Offload iSCSI Hardware Information

PCI, firmware and other adapter related details are provided in this module. Select the Chelsio adapter for which you want to view properties from the **Select a T4 Card** drop-down list and the module will expand to display related properties. You can also view details like link id, status, enode mac, etc of all the ports of the selected adapter.
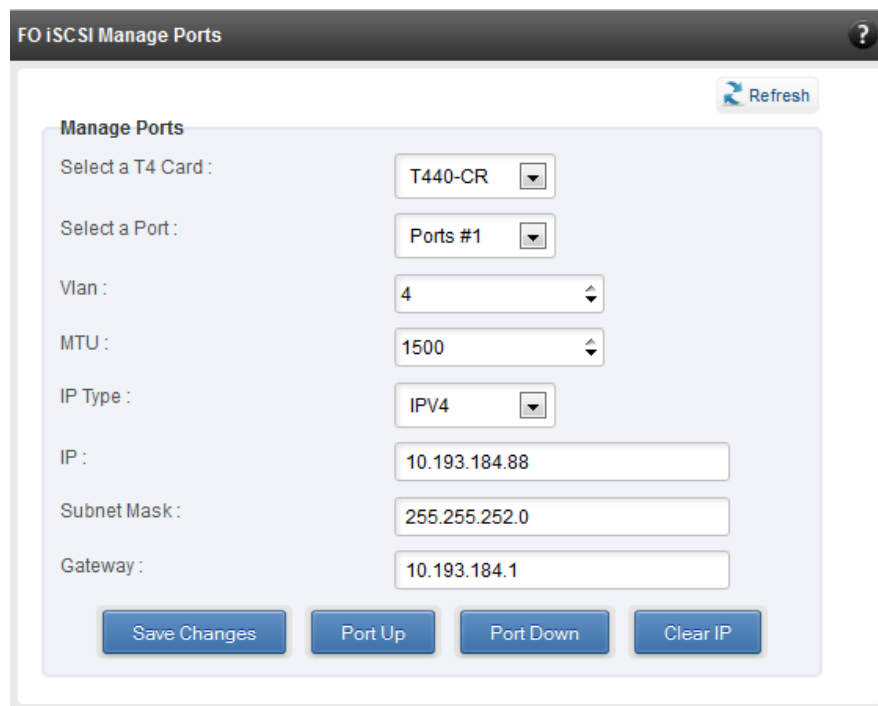
Figure 203 - Full Offload iSCSI Hardware Information module

● FO iSCSI Manage Ports

Here you can configure various port settings like VLAN id, Maximum Transmission Unit (MTU) and IP. Select a Chelsio adapter from **Select a T4 Card** drop-down list and then select the port for which you want set any of the aforementioned properties. MTU can be set between 1500-9000 bytes. VLAN id can be set within the range 0-4094 (enter 0 to disable it). The IP type can be *IPV4* (static) or *DHCP*.

The **Port Up** and **Port Down** buttons will enable and disable the selected port respectively. The **Clear IP** button deletes values set for the IP Type, IP, Subnet Mask and Gateway properties and resets them.

Figure 204 - FO iSCSI Manage Ports module

- ## FO iSCSI Initiator Properties
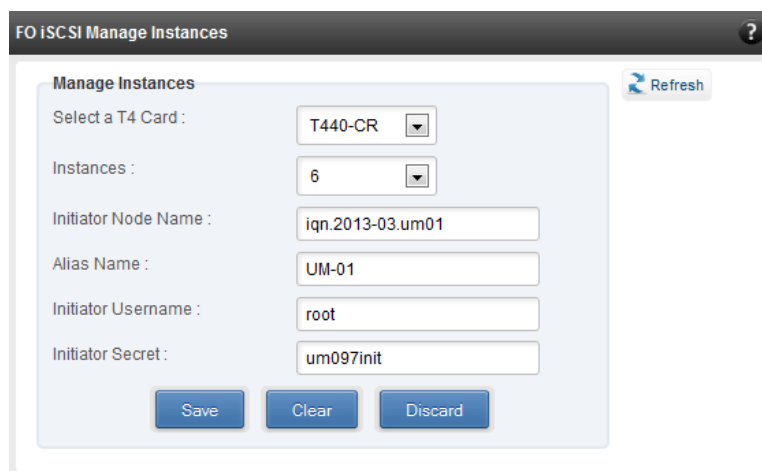
In the **FO iSCSI Initiator Properties** module, you can configure FO iSCSI Initiator by setting different properties like enabling/disabling CHAP authentication, setting Header and Data digest, etc.

Figure 205 - FO iSCSI Initiator Properties

- ## FO iSCSI Manage Instances

The FO iSCSI Initiator service maintains multiple instances of a target depending on the discovery method. In this module, you can set upto 8 instances. Configurable parameters include initiator node name (IQN), alias (friendly) name, Initiator (CHAP) Username and password.
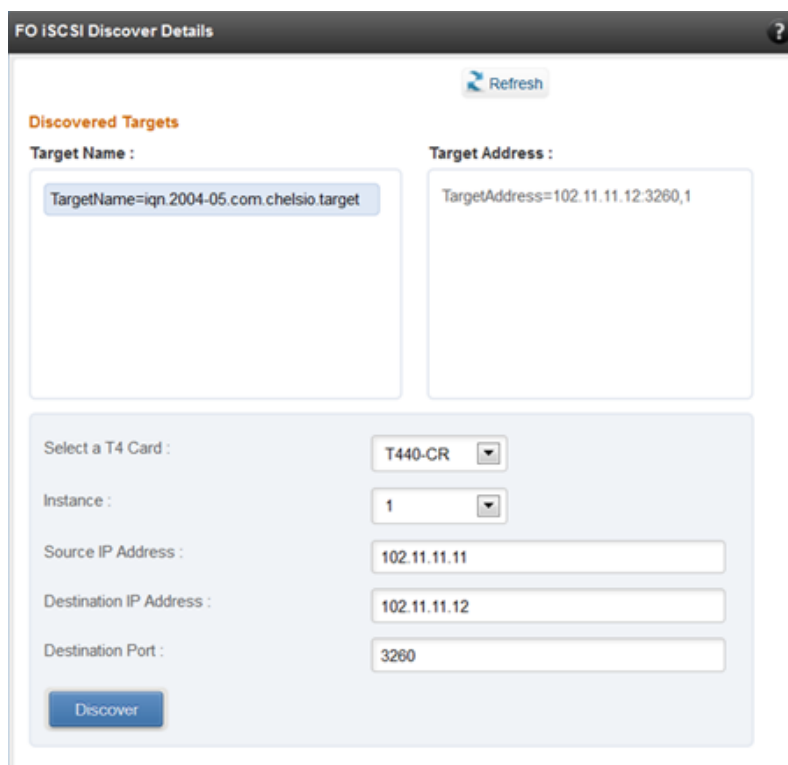
Figure 206 - FO iSCSI Manage Instances module

● FO iSCSI Discover Details

iSCSI Targets can be discovered using this module. Select a Chelsio adapter and initiator instance using which you want to discover targets. Next, provide the source (initiator) and destination (target) IP. Finally, click **Discover**. After successful discovery, all the discovered targets will appear in the **Discovered Targets** section. To view more details, click on the Target name.

Figure 207 - FO iSCSI Discover Details module

○ **FO iSCSI Session Details**

The FO iSCSI Session Details module can be used to log onto targets and view details of established iSCSI sessions. You can also logout from a target

Use the **Login** section to connect to a target. *Adapter*, *(initiator) instance*, *Target Name*, *Source (Initiator) IP*, *Destination (Target) IP* and *Destination Port* are mandatory. After providing values for these fields, click **Login.**

By default, no authentication mechanism is used while connecting to a target. You can however configure CHAP for a secure iSCSI connection. **One-way** (target authenticates the initiator) and **Mutual** (target and initiator authenticate each other) authentication methods are supported.

Figure 208 - FO iSCSI Session Details module: Login

After successful login, details of the established iSCSI session will be displayed under the **Established sessions** section. Select the adapter and session id. Details of the selected session will be displayed. To end the session, click **Logout**.
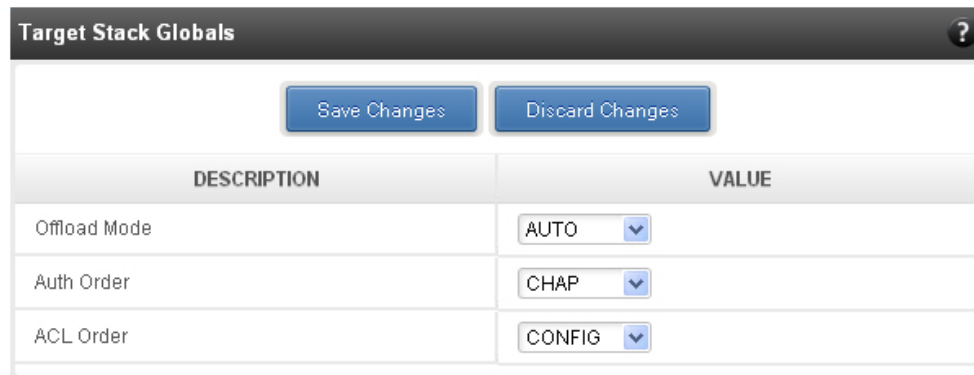
Figure 209 - FO iSCSI Session Details module: Established Sessions

8.7.4. iSCSI Target page (Linux)

This page allows to create new Targets and manage them (add/delete portals, add/delete LUNs, add/delete ACLs). It also provides information on Session details. Viewing and modifying Target properties is also available. The modules available on this page are as below:

- **Target Stack Globals**

This module displays various global properties of a currently connected iSCSI target. Authentication priority between CHAP and ACL can be set here.

Figure 210 - Target Stack Globals module

Target properties

Properties such as Target name and Alias, Max Data Receive Length, Authentication mode related to a specific iSCSI target can be viewed and modified here. iSCSI targets can be started/stopped or deleted.

Figure 211 - Target properties module

○ **Session details**

Details including Session ID, Initiator IQN and Connections List of all discovered and currently connected iSCSI targets are listed here.
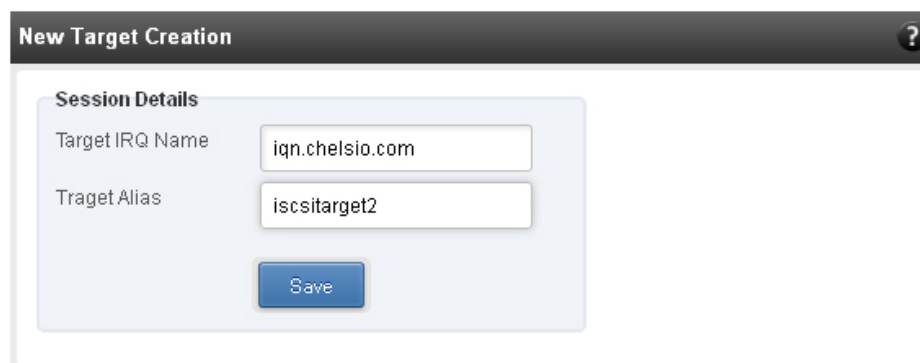
Figure 212 - Session Details module

○ **New Target Creation**

New iSCSI target can be created here by specifying the Target IQN and Target Alias name.

Figure 213 - New Target Creation module

8.7.5. LUNs

Various Logical Units created in an iSCSI Target can be managed here. The modules available on this page are as below:

- **View/Edit iSCSI Target LUNs**
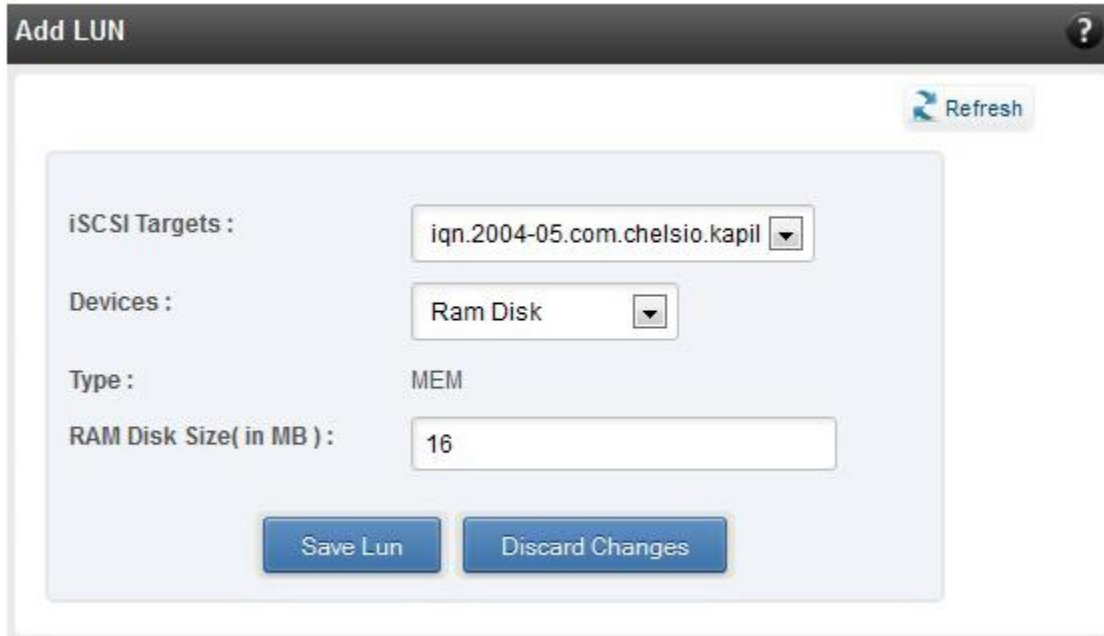
This module displays various Logical Units created in an iSCSI Target. Selected LUNs can be deleted.

Figure 214 - View/Edit iSCSI Target LUNs module

- **Add LUN**

New LUNs can be added here by providing various parameters like Target Name, Target Device and RAM Disk Size etc. RW (Read-Write) and RO (Read Only) are the two kinds of permissions that can be set. If Ram Disk is selected, then a minimum of 16 MB should be provided.

Figure 215 - Adding a new LUN

8.7.6. Portal Groups

Portal details for currently connected iSCSI Targets can be viewed and added here. The modules available on this page are as below:

- **View/Edit iSCSI Target Portals**

Portal List on the left displays details of the portal group on which an iSCSI target is listening and the related info is displayed on the right under Portal Details. Selected portals can be deleted.

Figure 216 - View/Edit iSCSI Target Portals module

○ **Add Portal**

New Portals can be added here by choosing the specific target and Portal IP address. The Port number should be 3260.

Figure 217 - Adding a new Portal

8.7.7. ACLs

ACLs configured for currently connected iSCSI Targets can be managed here. The modules available on this page are as below:

○ **View/Edit iSCSI Target ACLs**

This module displays details for all the ACLs configured for an iSCSI Target. Selected ACLs can be deleted.

Figure 218 - Target ACL operations module

- **Add ACL**

New ACLs can be configured by specifying Target name, initiator IQN name, IP address and permission type.

Figure 219 - Adding new ACL

8.8. Hardware Features

The **Hardware** module lists the status of configuration modules under Hardware Features section, running on the agent.

Figure 220 - Hardware module

8.8.1. Filtering (Linux)

Using this page, one can control the traffic from a specific IP. The module available on this page is as below:

T3 Filtering configuration

T3 Filtering options can be set only when offload driver (*t3_tom*) is not loaded.

This module lists the various parameters which can be set while determining filtering options for a system IP. You can set the maximum number of filters and also add/delete filters. A filter with default values (the **Action** field set to **pass**; the **Protocol** field set to **any**) is created at the time of configuring the filtering module. To remove the default filter, enter 0 in the **Set Maximum Filters** field and click on **Set Filters**. The fields **IfName** and **FilterId** are mandatory. The **Action** field is set to **pass** and the **Protocol** field is set to **any** by default. Other possible values for the **Protocol** field are **tcp**, **udp** and **frag**. The **Priority** field can be used to determine the priority of a filter when Vlan ids are same. **Insert at position** features allows user to add a filter at a specified position.

Figure 221 - T3 Filtering Configuration module

ⓘ Note   *Results for actions like adding a new filter or setting maximum filters make some time to reflect. Highlight the system item in the tree menu on the left, and click "Refresh system", to refresh data from the system, in case the updated settings are not being shown.*

T4 Filtering configuration

T4 Filtering options can be set only when offload driver (*t4_tom*) is not loaded.

A list of pre-defined filter selection combinations is displayed. The combination *fragmentation, mpshittype, protocol, vlan, port, fcoe* is active by default. To select a different combination, highlight it in the **Combinations** list by clicking and click "Set Active Combination".

You can create filter rules for any combination in the list. However, filter rule created only for the Active Combination will apply. To create a new rule, select a combination and click "Add a filter rule". The **FILTERID** and **T4 CARD** fields are mandatory. After providing appropriate values for the parameters click "Save Changes".

> ℹ Note   *For a detailed explanation regarding different fields, please refer **cxgbtool** manual by running* `man cxgbtool` *command on Management Agent CLI.*

Figure 222 - T4 Filtering Configuration module

8.8.2. Traffic Management (Linux)

Using this page, one can add/delete/modify offload policies only in the presence of offload driver (*t3_tom* for T3 adapters*; t4_tom* for T4 adapters).

8.8.3. Traffic Management configuration

The **Chelsio Card** section on the left displays all the cards available in the server and their corresponding policies on the right. Policies can be added and deleted. Policy Details displays the primitives (maximum 8) and actions which can be modified. For more details on creating policies, please refer to COP man pages.

Figure 223 - Traffic Management Configuration module

8.8.4. Boot

- ### T4 Save Config File (Linux)

This module displays the current T4 configuration tuning option selected. You can also change the tuning option by selecting the config file for each option located in */ChelsioUwire-x.x.x.xx/src/network/firmware*. For instance, to select *Low latency Networking*, locate the file, *t4-config.txt, in /ChelsioUwire-x.x.x.xx/src/network/firmware/low_latency_config* directory.
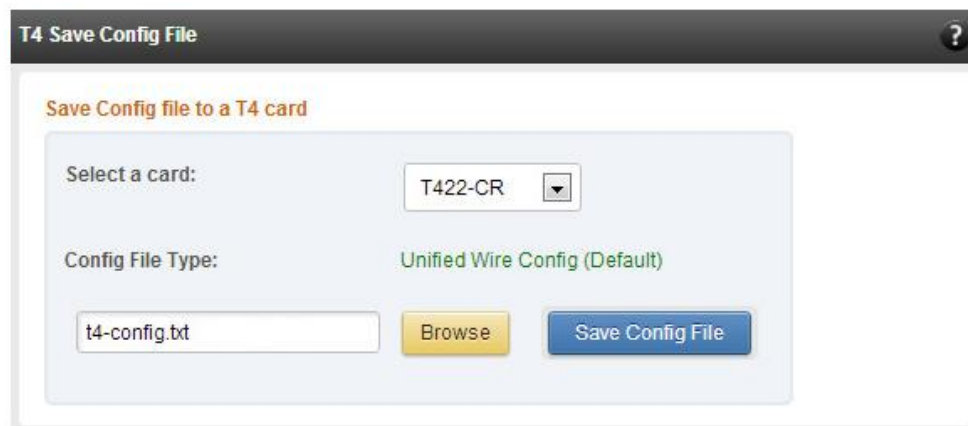
Figure 224 - T4 Save Config File module

8.8.5. T3 iSCSI / T3 PXE Boot Option Rom Management (Linux)

The Chelsio T3 card may be used for PXE or iSCSI boot. This module allows managing the boot capability. The Option ROM (PXE/iSCSI) may be installed to the card, or erased from the card. iSCSI boot Option ROM settings can be configured for the card here. Enable the iSCSI Option ROM only if you are planning to boot the system via iSCSI. Refer the iSCSI boot documentation for more details about its configuration.

Figure 225 - Option ROM management module for a T3 card

8.8.6. T5/T4 Boot Option ROM management

This module allows managing the PXE and FCoE boot capability for Chelsio T5 and T4 cards. The Option ROM (PXE and FCoE) may be installed to or erased from the card. The version of Option ROM flashed can be viewed here.
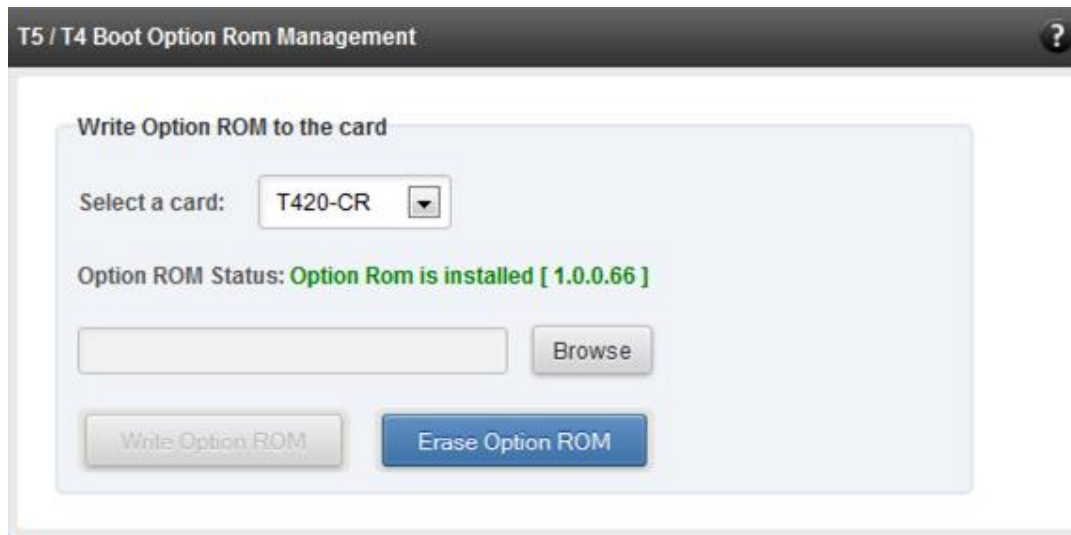
Figure 226 - T5/T4 Option ROM Management module

8.8.7. T5/T4 Boot Configuration

This module can be used to view and configure PXE and FCoE Option ROM settings for Chelsio T5 and T4 cards.

PXE physical functions and order of ports for PXE boot can be selected here. The same module can be used to configure FCoE Option ROM. Here port order for target discovery and discovery timeout can be set. Under the Boot section, LUN can be configured as boot device from the discovered FCoE targets.

Enable the Option ROM only if you are planning to boot the system via PXE or install the operating system on FCoE LUN.

Figure 227 - PXE Boot configuration for T4 CNAs

Figure 228 - FCoE Boot configuration for T4 CNAs

Figure 229 - iSCSI Boot configuration for T4 CNAs

8.8.8. Bypass (Linux)

You can use the **Bypass** page to configure various settings for Chelsio's bypass adapters like setting bypass operation mode, creating rules (filters), starting/stopping BA server, etc. There are two modules available: **Bypass Configuration** and **Redirect Configuration**.

● Bypass Configuration

In the **Bypass Configuration** module, you can view the status and start/stop the BA server accordingly. The adapter will redirect packets using the mode speciffied in the **Default bypass mode** field unless otherwise specified in the **Current bypass mode**.

The **Watchdog** timer is used to ensure that if there is a software failure, the switch will enter the default state. The **Watchdog timeout** value should be provided in milliseconds.

Figure 230 - Bypass Configuration module

◉ **Redirect Configuration**

In the **Redirect Configuration** module, you can set **rules** (filters), based on which the bypass adapter will redirect packets. You can group **rules** into **tables**. You can save the currently configured tables and rules for a bypass adapter into a shell script using the **Download Configuration** button.

The **Table Configuration** tab displays BA server status and the number of tables created. You can create new tables or perform various actions on the existing ones.

- **Delete table**: Delete the selected table and all the rules present in it.
- **Purge table**: Delete all the rules present in the selected table. This action will not delete the table.
- **Activate table**: Enable the selected table.
- **Deactivate table**: Disable the selected table.
- **Create table**: Create a new table. The new table created will be inactive by default. Use the **Activate table** option to enable it. You can cretate upto 5 tables.

In the **Rules Configuration** tab, you can add, delete and configure rules. Use the **Add a Filter row** button to add a new rule by specifying the rule id in the *INDEX* field and providing the required parameters. Finally, click **Save** Changes.

To edit an existing rule, select the corresponding checkbox, change the desired paremeters and click **Save Changes**.

To delete a rule, select the corresponding checkbox and click **Delete a Filter.** Finally, click **Save** Changes. You can delete multiple rules using this method.
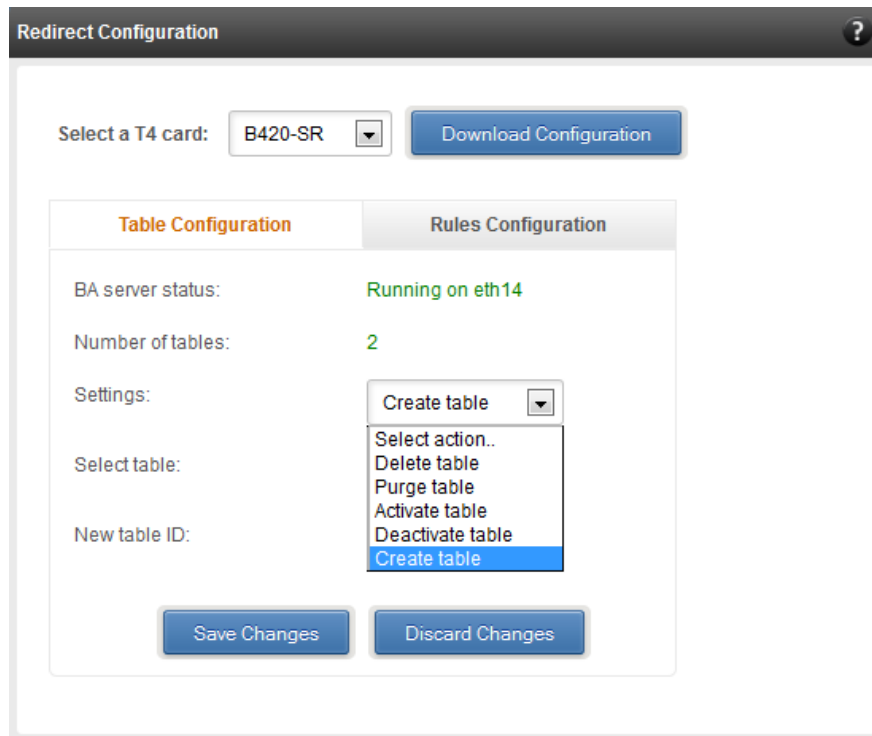

Figure 231 - Redirect Configuration module: Table configuration tab

Figure 232 - Redirect Configuration module: Rules configuration tab

8.8.9. T4 Egress Class Schedulers (Linux)

Schedulers can be set only when T5/T4 network driver (*cxgb4*) is loaded.

● Egress Queue Map

Using this module, you can bind (map) NIC (non-offloaded) Tx queues to Tx Scheduler classes.

Figure 233 - Egress Queue Map module

● Egress Packet Scheduler

Using this module you can configure different scheduler hierarchy levels (i.e.Class Rate Limiting, Class Weighted Round Robin and Channel Rate Limiting). Based on the parameters specified, different scheduler levels can be configured.

Figure 234 - Egress Packet Scheduler module

9. Uninstallation

This section describes the procedure to uninstall UM. For both installer and zip package, the application will have to be uninstalled manually.

i. For Installer, open **Programs and Features** in the **Control Panel**. Double click on *UnifiedWireManager*.

   For zip packages, you can use the **Control Panel** as described above or run the **UnifiedWireManager-x.x.xx-x64** installer application located in *ChelsioUwire-x.x.x.xx/UM* folder.

ii. Select **Modify** to add or remove features or **Remove** to uninstall the application.

Figure 235 - Selecting maintenance option

a. If you select **Modify**, you can select the UM component you want to uninstall by deselecting the appropriate checkbox. To install a component that was skipped during installation, select the corresponding checkbox. Click **Next**.

Figure 236 - Modifying UM components

b. If you select **Remove**, you can uninstall the application completely. Click **Next** and then **Yes** to confirm.

Figure 237 - Uninstalling UM

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

Figure 238 - Finishing uninstallation

IX. UM for Win PE

1. Introduction

Chelsio **UM for Win PE** is a light-weight management tool developed for Windows PE (Windows Preinstallation Environment) platform and provides features to view and configure different aspects of the system, including Chelsio adapter installed.
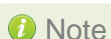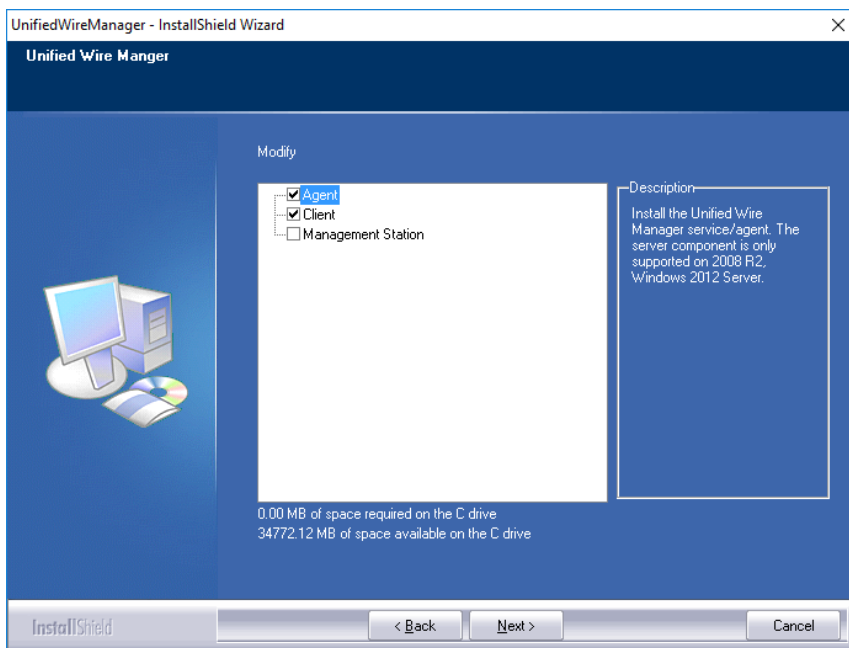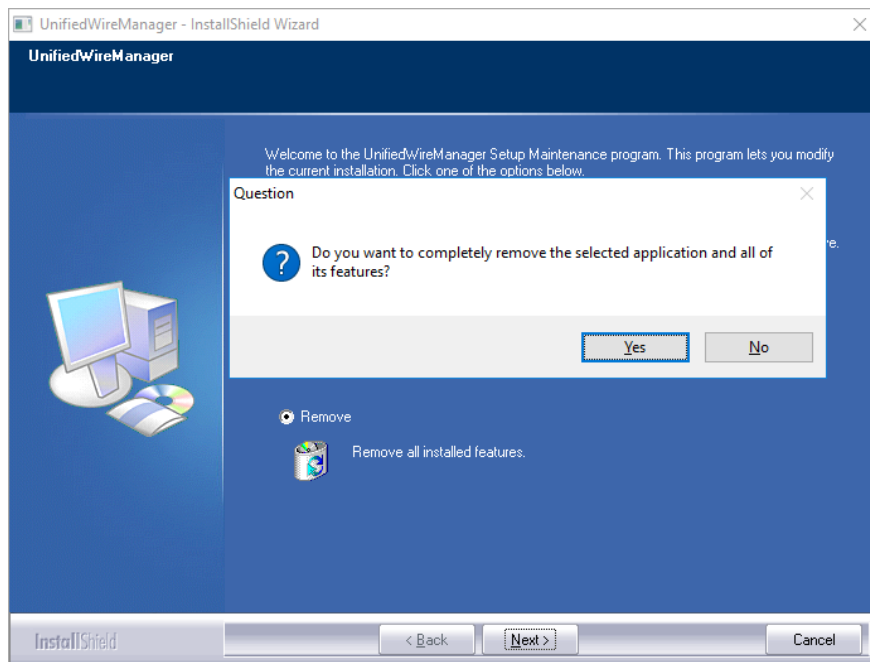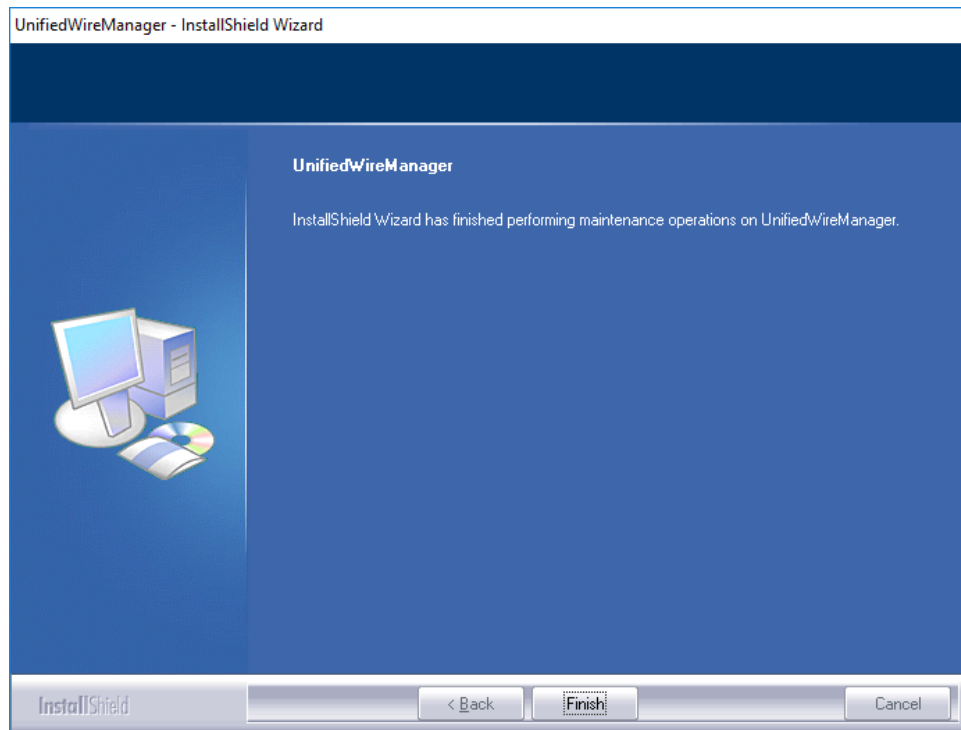
The tool provides command-line options to view system related information like list of drivers installed, Chelsio adapters present, etc. You can also view and configure Chelsio adapter settings like card and port properties, flash and erase option ROM, etc. To view the complete list of commands available, invoke the help by typing `chelsio_uwlite.exe` in the command prompt.

The tool is copied to *<system_drive>\ChelsioUwire\docs* during Unified Wire installation. In order to use the tool, you will have to add it to boot image when creating bootable Windows PE media.

For more information, visit the following link: http://technet.microsoft.com/en-us/library/cc749312%28v=ws.10%29.aspx

> ⓘ Note   *Only local systems can be managed currently using this tool.*

1.1. Hardware Requirements

1.1.1. Supported Adapters

The following are the currently shipping Chelsio adapters that are compatible with Chelsio UM for Win PE:

- T580-CR
- T580-LP-CR
- T580-SO-CR
- T540-CR
- T520-CR
- T520-LL-CR
- T520-SO-CR
- T440-CR
- T440-LP-CR
- T420-CR
- T422-CR
- T420-LL-CR
- T420-CX
- T420-BT
- T404-BT

1.2. Software Requirements

1.2.1. Windows Requirement

Currently Chelsio UM for Win PE is available for the following Windows version(s):

- Server 2012 R2

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

1.3. Examples

1.3.1. Viewing help

- Run the following command to view the list of commands available under a service.

```
X:\windows\system32>chelsio_uwlite.exe <service-name>
```

E.g.

To view commands available under the chelsio service:

```
X:\windows\system32>chelsio_uwlite.exe Chelsio
```

```
X:\windows\system32>chelsio_uwlite.exe chelsio

--------Usage-------

chelsio_uwlite <Service-Name>
                [StackName] <Command> [SubCommand] [Arguments]
Where
        Service-Name                 : SYSTEM:CHELSIO
        Command                      : A command supported by the service
                                       (see service help for a list of commands)
        Arguments                    : Arguments expected by the command
                                       (see command help for more details)
Note:
        All arguments should be in the <key=value> format
        eg: <ip=192.168.1.1>
        Required parameters are enclosed within <> brackets
        Optional parameters are enclosed within [] brackets

Service:
        CHELSIO

Commands
--------
        ListAdapters
        ListAdapterProps
        ListPorts
        GetCardStats
        GetPortStats
        SetCardProperty
        SetPortProperty
        GetProperty
        SetProperty
        FlashRom
        EraseRom
        EraseSettings
        Defaultconfig
```

Figure 239 – 'chelsio' service commands syntax and usage

- Run the following command to view syntax and usage for a sub-command:

```
X:\windows\system32>chelsio_uwlite.exe chelsio <sub-command> /?
```

E.g. To view syntax and usage of *EraseSettings* sub-command under *chelsio* service:

```
X:\windows\system32>chelsio_uwlite.exe chelsio erasesettings /?
```



```
X:\windows\system32>chelsio_uwlite.exe chelsio erasesettings /?

This command not supported

--------Usage-------

chelsio_uwlite <Service-Name>
                [StackName] <Command> [SubCommand] [Arguments]
Where
        Service-Name                    : SYSTEM:CHELSIO
        Command                         : A command supported by the service
                                          (see service help for a list of commands)
        Arguments                       : Arguments expected by the command
                                          (see command help for more details)
Note:
        All arguments should be in the <key=value> format
        eg: <ip=192.168.1.1>
        Required parameters are enclosed within <> brackets
        Optional parameters are enclosed within [] brackets

Service:
        CHELSIO

Note:
        <adapter=adapter index> is the Adapter # obtained from the ListAdapters command

chelsio_uwlite
                Chelsio
                EraseSettings
                <adapter=adapter index>
                <force=Boolean>
```

Figure 240 - 'EraseSettings' sub-command syntax and usage

1.3.2. Flashing Option ROM

i. Run the following command to list all Chelsio CNAs present in the system. The list displays a unique index for each CNA found.

```
X:\windows\system32>chelsio_uwlite.exe chelsio listadapters
```

Figure 241 - Listing Chelsio Adapters

ii. Now, run the following command to flash option ROM onto the adapter:

```
X:\windows\system32>chelsio_uwlite.exe chelsio flashrom adapter=<idx>
path=<option_rom_image>
```

Here, `idx` is the CNA index found in step i (0 in this case) and `option_rom_image` is the path to Option ROM image file, *cuwlbt4.bin*

Figure 242 - Flashing option ROM

1.3.3. Erasing Option ROM

i. Run the following command to list all Chelsio CNAs present on the system. The list displays a unique index for each CNA found.

```
X:\windows\system32>chelsio_uwlite.exe chelsio listadapters
```


Figure 243 - Listing Chelsio Adapters

ii. Now, run the following command to erase option ROM from the adapter:

```
X:\windows\system32>chelsio_uwlite.exe chelsio eraserom adapter=0 force=1
```

Here, `idx` is the CNA index found in step i (0 in this case)

Figure 244 - Erasing option ROM

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