Chelsio Communications

Accelerate

Chelsio T5/T4 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	05/10/2016

TABLE OF CONTENTS

I. Chelsio Unified Wire

1. Introduction

Thank you for choosing Chelsio T5/T4 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 5 (T5) is Chelsio's next generation of highly integrated, hyper-virtualized 40/10GbE controllers. The T5 is 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 true 40Gb 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.

The T4 adapters can fully offload TCP, UDP, iSCSI, iWARP and FCoE over a single Unified Wire. The adapters also fully support SR-IOV, EVB/VNTag, DCB, Traffic Management and Filtering.

Ideal for all data, storage and high performance clustering applications, the T5/T4 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 T5/T4 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 T5 and T4 Unified Wire Adapters. The package provides an interactive installer to install various drivers and utilities.

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)
- NDIS SR-IOV
- Virtual RSS (vRSS)
- iSCSI Storport Miniport
- Data Center Bridging (DCB)
- Unified Wire Manager (UM)
- UM for Win PE

1.2. Hardware Requirements

The Chelsio T5/T4 Unified Wire supports all x64 architectures supporting PCIE (x4, x8) 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.*

1.3. Software Requirements

The Chelsio T5/T4 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.

1.4. Package Contents

Chelsio T5/T4 Unified Wire Software package 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*>*ChelsioT4**docs*\ directory after installing Chelsio Unified Wire.

2. Hardware Installation

1. Shutdown/power off your system.
2. Power off all remaining peripherals attached to your system.
3. Unpack the Chelsio adapter and place it on an anti-static surface.
4. Remove the system case cover according to the system manufacturer's instructions.
5. Remove the PCI filler plate from the slot where you will install the Ethernet adapter.
6. For maximum performance, it is highly recommended to install the adapter into a PCIE x8 slot.
7. 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.
8. Secure the Chelsio adapter with a screw, or other securing mechanism, as described by the system manufacturer's instructions. Replace the case cover.
9. After securing the card, ensure that the card is still fully seated in the PCIE x8 slot as sometimes the process of securing the card causes the card to become unseated.
10. Connect a fiber cable, multi-mode for short range (SR) optics or single-mode for long range (LR) optics, to the 10Gb Ethernet adapter or regular Ethernet cable for the 1Gb Ethernet adapter.
11. Power on your system.
12. Verify if the adapter was installed successfully. To do so, open **Device Manager** in **Control Panel**.
13. 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

14. 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 and T420-BT 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 Installer provides two methods of installation: GUI and 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.

3.1. Pre-requisites

3.1.1. Unified Wire Installer

Please download and install the latest Microsoft Visual C++ Redistributable Packages for Visual Studio (x86 and x64) before running the installer.

3.1.2. NDIS SR-IOV

SR-IOV should be enabled in the machine.

3.1.3. Unified Wire Manager

Chelsio Unified Wire Installer has been designed to install Unified Wire Manager (UM) along with other drivers by default. Based on the Windows version running on the system, the three UM components, i.e. Management Agent, Client and Station, will be installed and hence no separate installation is required. If an older version of UM exists, the Installer will upgrade it to the current version.

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

3.1.4. 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.

3.1.5. Management Station

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

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

3.2. Installing Unified Wire

3.2.1. GUI mode (Installer)

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

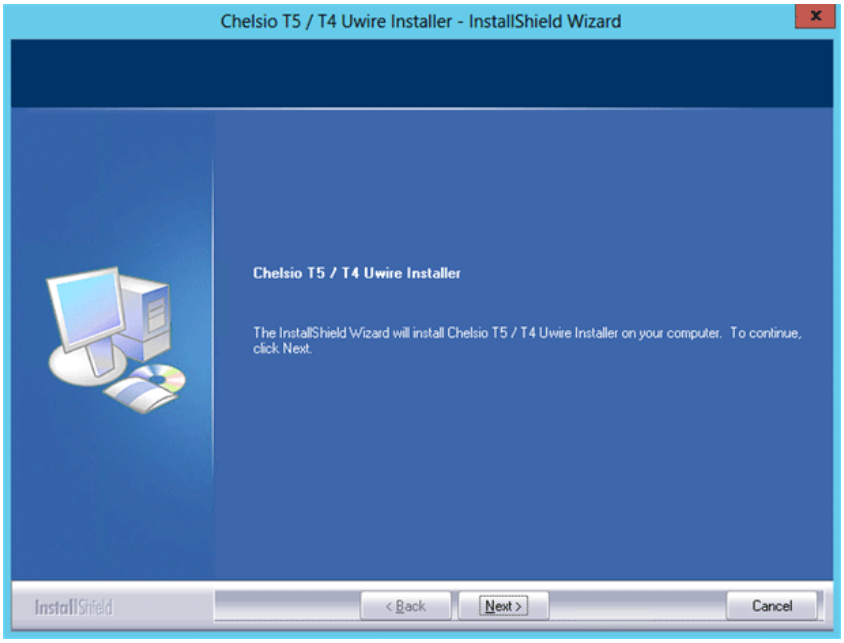

Figure 1 - Unified Wire Installer welcome window

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

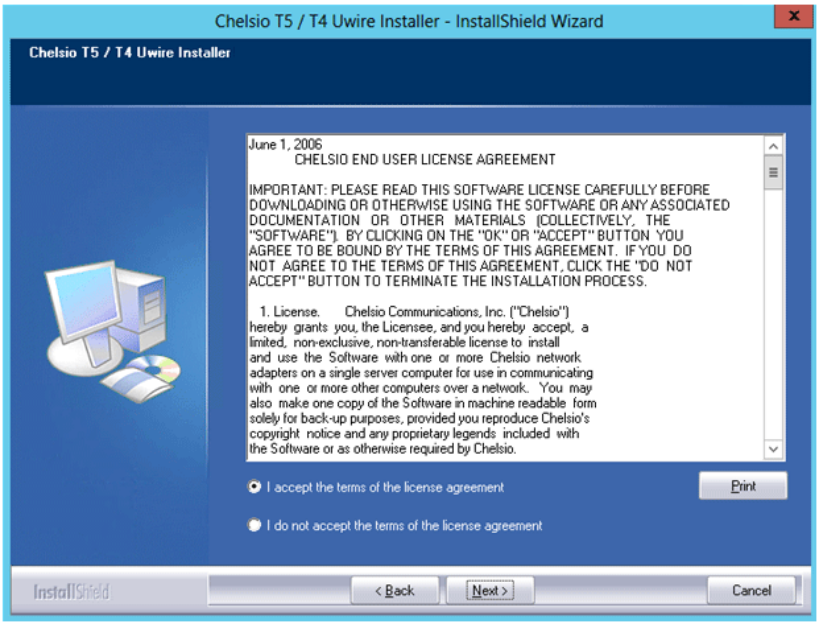

Figure 2 - Chelsio EULA window

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

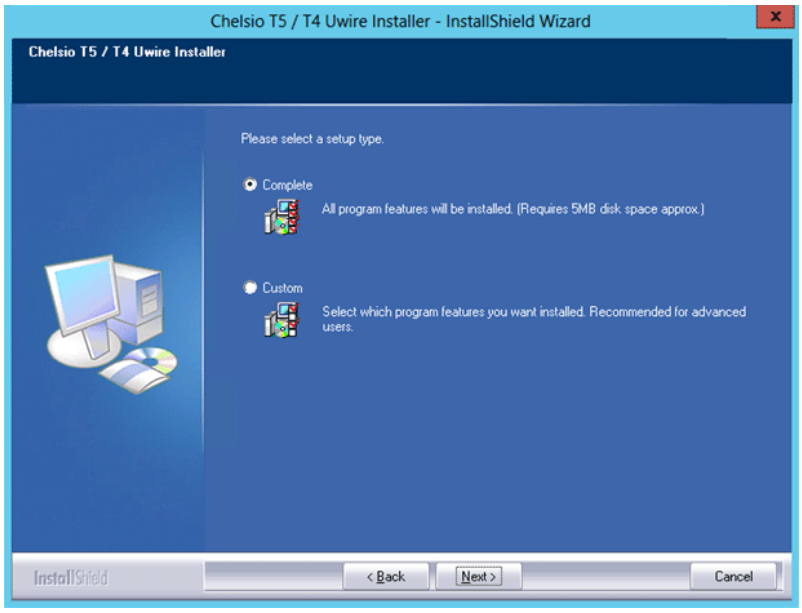

Figure 3 - Select setup (installation) type

Note *Selecting **Complete** will not install iSCSI Storport Miniport driver. To install the driver, select **Custom.***

i. 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.

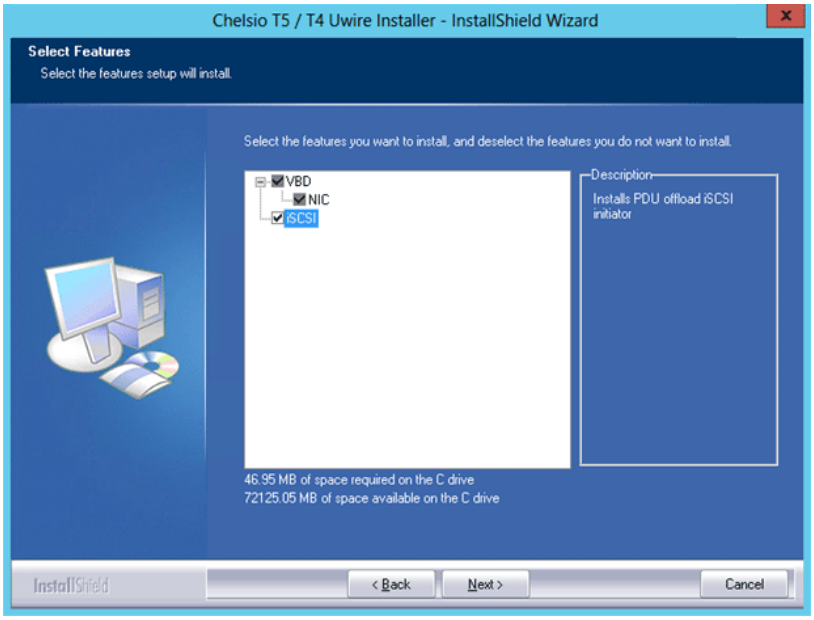

Figure 4 - Customizing the installation

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

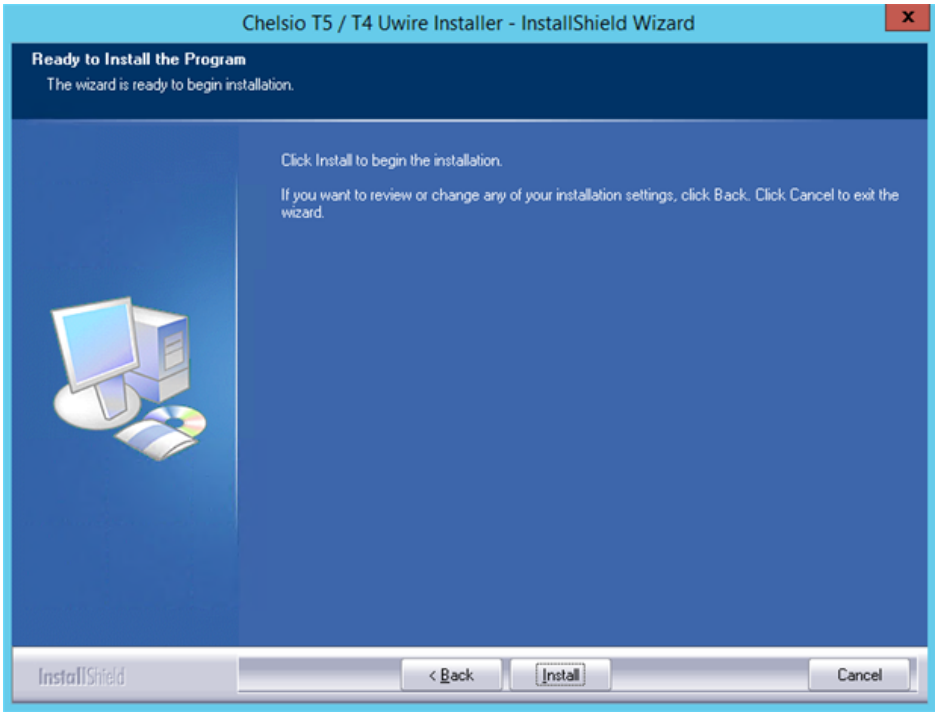

Figure 5 - Start Installation

6. Selected drivers will now be installed.

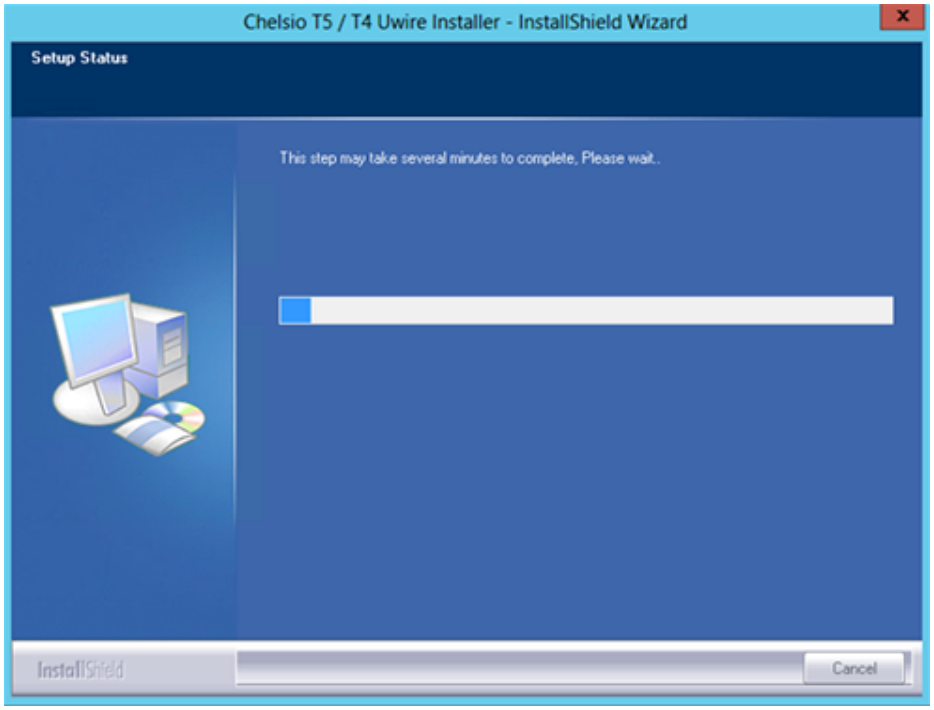

Figure 6 - Installation in progress

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

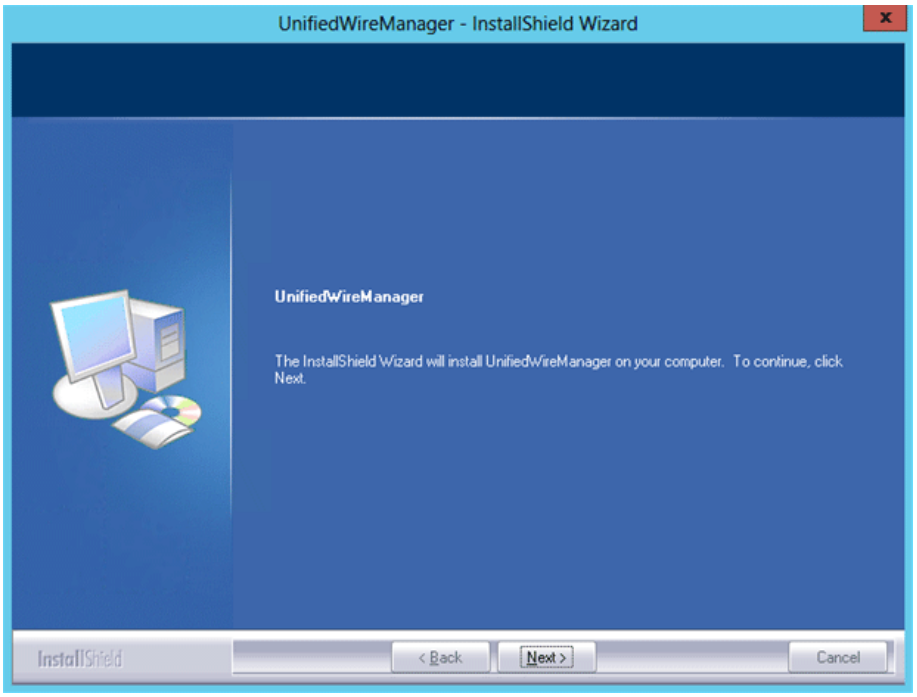

Figure 7 - UM installer welcome window

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

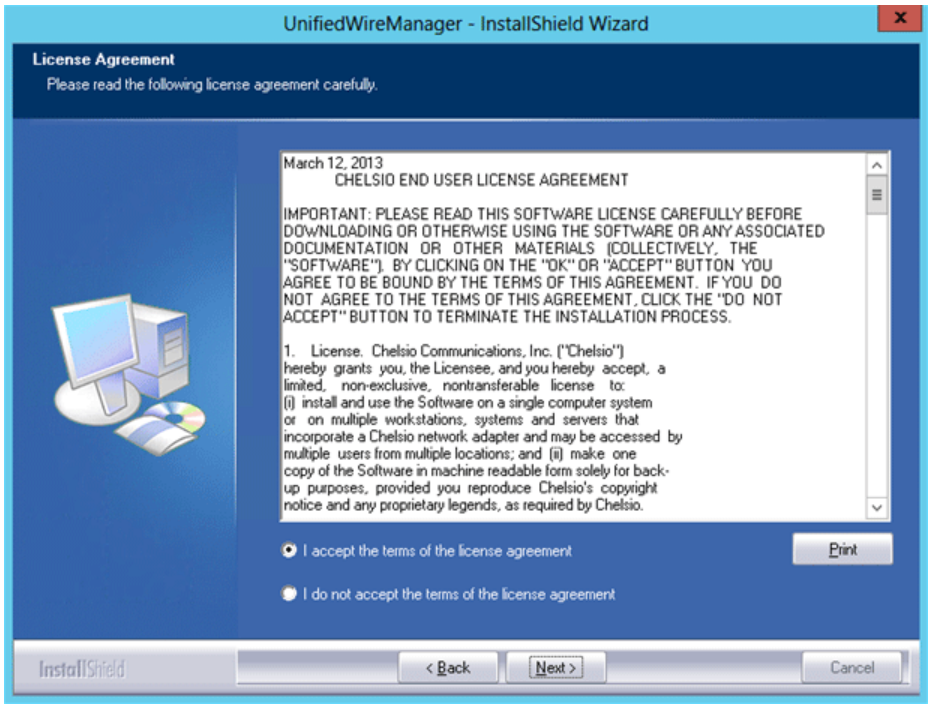

Figure 8 - Chelsio EULA window

9. The next window will display the pre-requisites for various UM components. Ensure that they are met before proceeding. Click Next.

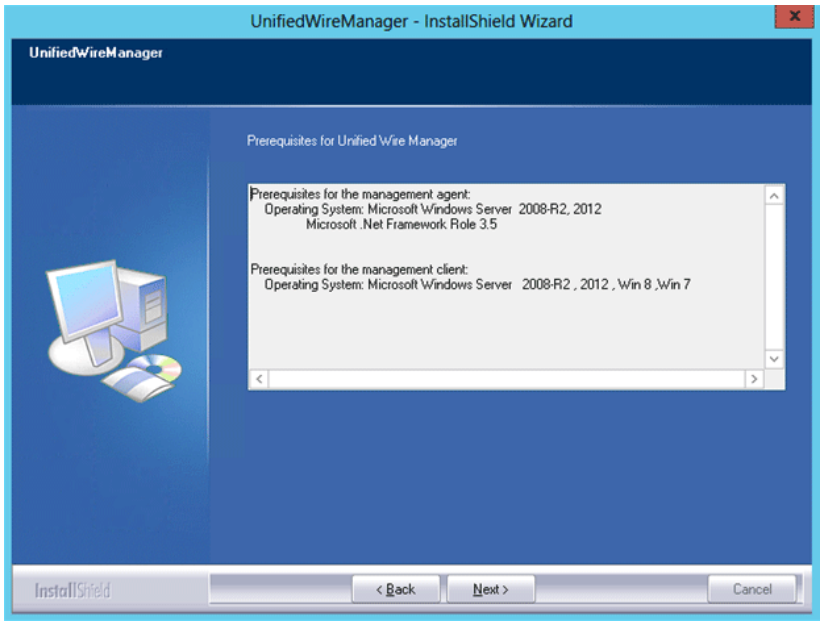

Figure 9 - UM prerequisites window

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

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

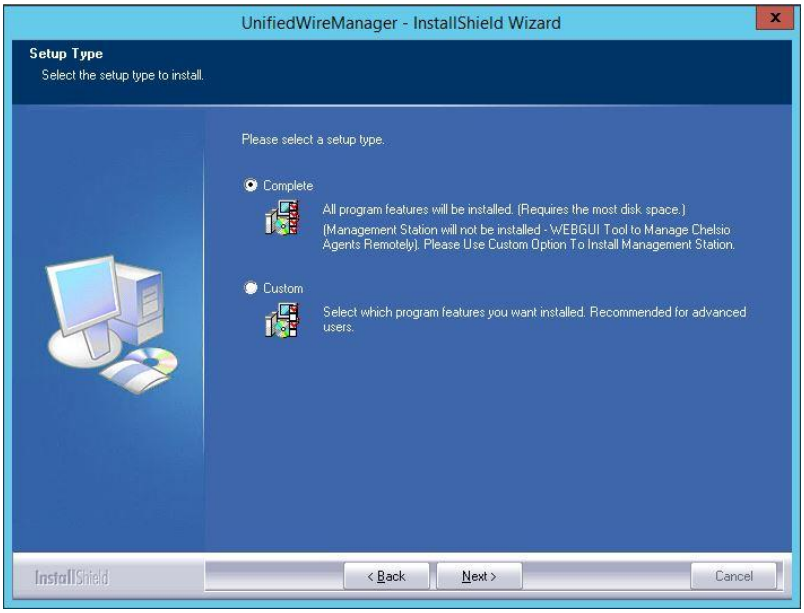

Figure 10 - Select setup (installation) type

i. 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.

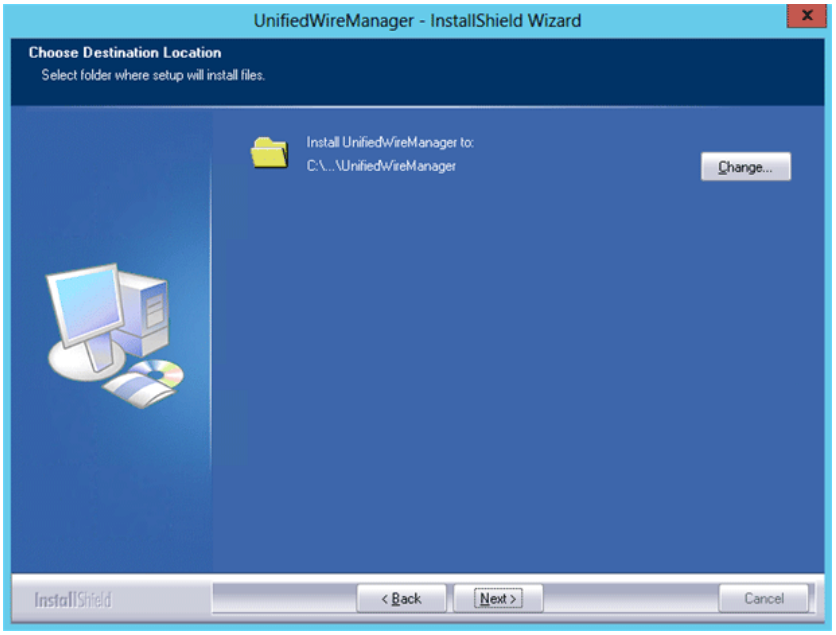

Figure 11 - Changing UM installation path

ii. Next, you can choose the UM components you wish to install. Deselect the components you don't wish to install and click **Next.**

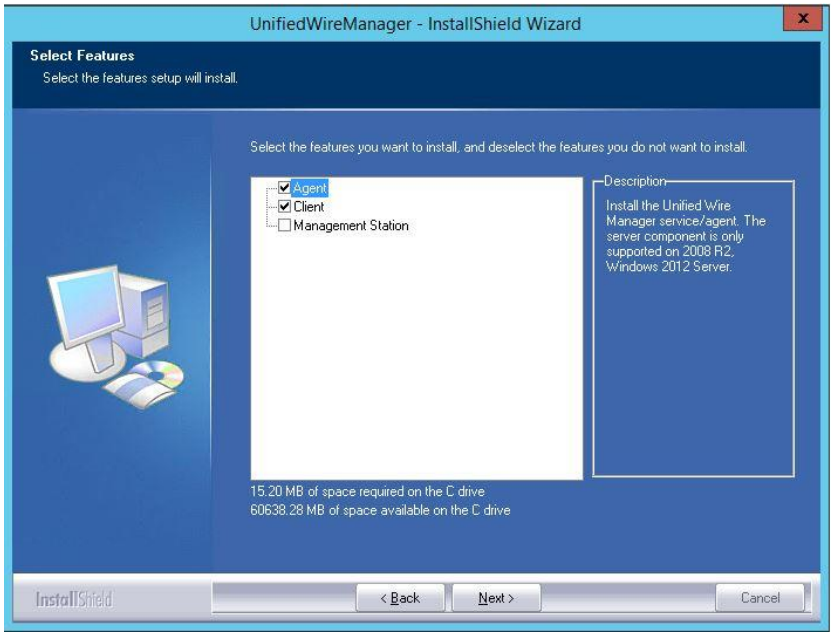

Figure 12 - Selecting UM components to install

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.*

11. Click **Install** to start the installation. Unified Wire Manager will now be installed with the selected options.

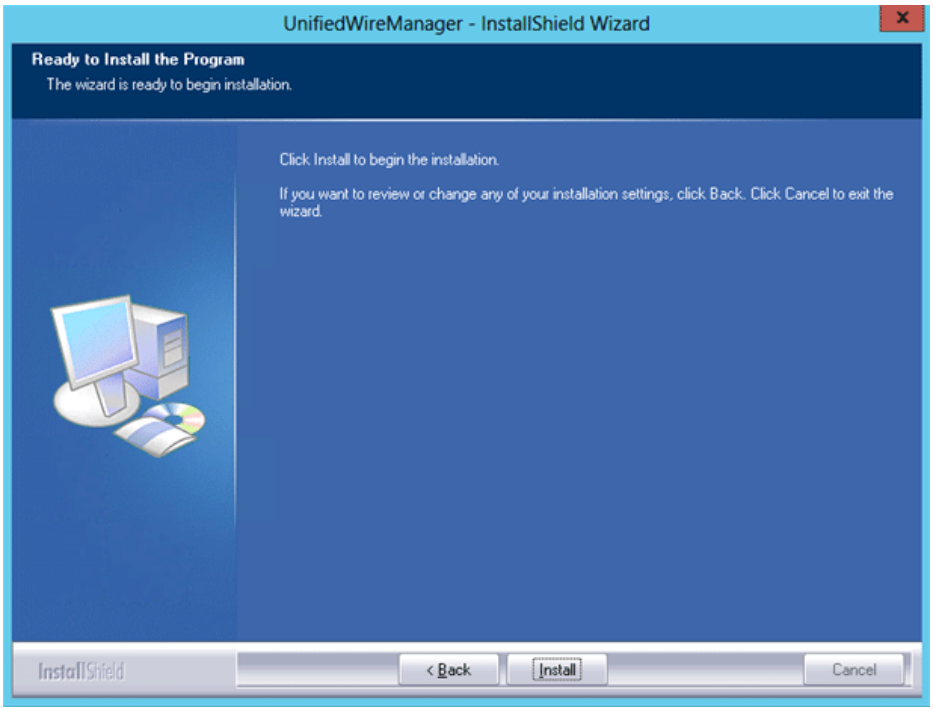

Figure 13 - Starting UM installation

12. Select **Finish** to exit the UM Installer.

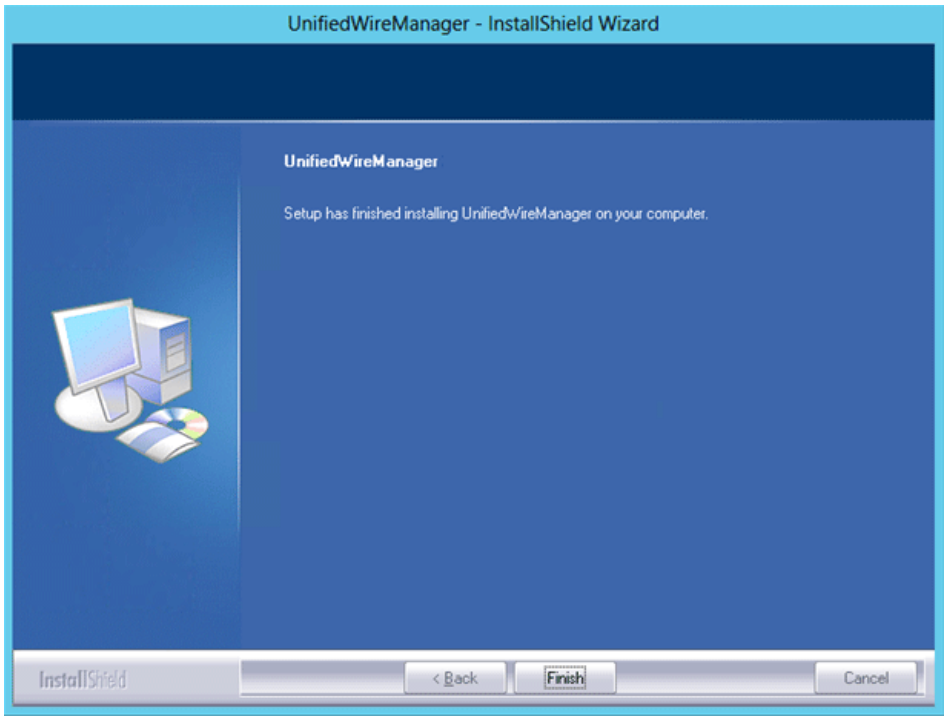

Figure 14 - Finishing UM installation

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

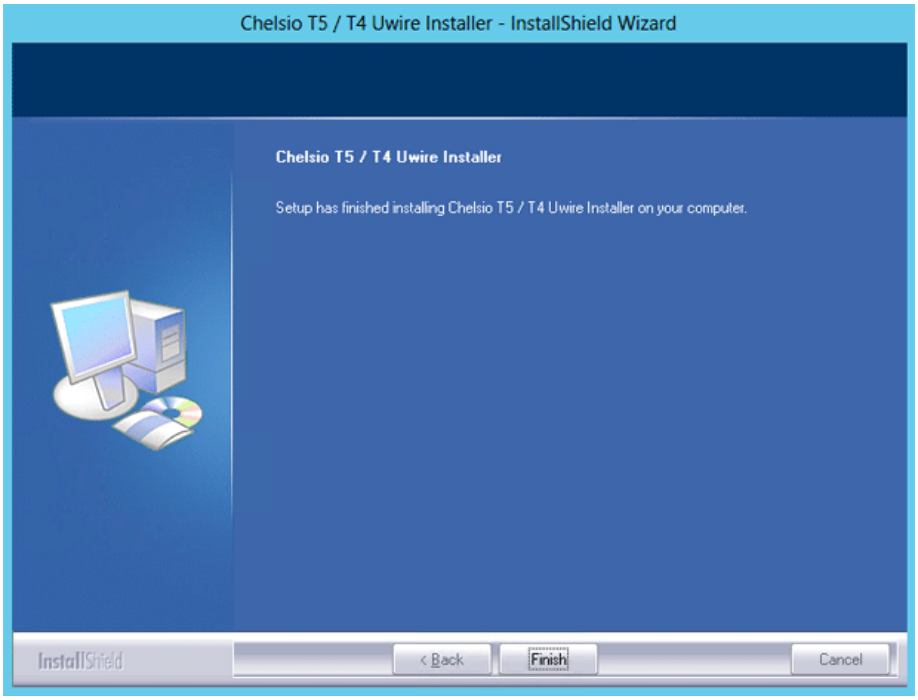

Figure 15 - Finishing Unified Wire installation

3.2.2. 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.*

4. 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 (T5, T4 or both). These machines can be connected directly (back-to-back) or with a switch.

4.1. Configuring 40G adapters

Chelsio T5 40G adapters can be configured in the following three operational modes:

i. **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).
ii. **SPIDER (4X10G)**: In this mode, port 0 functions as 4 10Gbps links and port 1 is disabled.
iii. **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).

To configure Chelsio adapter in any of the three modes mentioned above, use the *chelsio_adapter_config.ps1* configuration script. The script will be present in *<system_drive>\Windows\system32* folder. Follow the steps mentioned below:

i. Open PowerShell with administrative privileges.
ii. Run the adapter configuration script and select *Windows Technical Preview GUI* (option 1) as the Windows version. Hit [Enter].

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

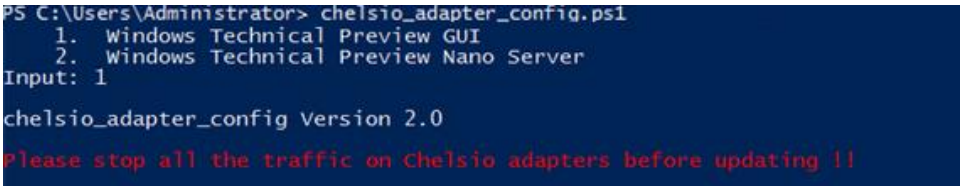

Figure 16 - Adapter Configuration Utility

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

Figure 17 - Selecting adapter

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

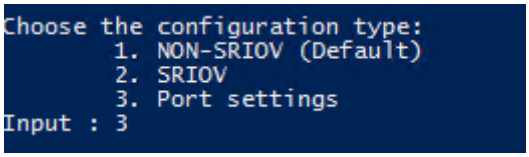

Figure 18 - Port Settings

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

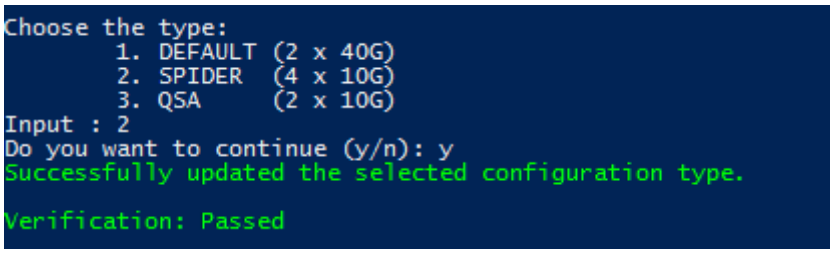

Figure 19 - Operation mode

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

4.2. 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.

5. Mass Deployment

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 has to be installed on the host machine and the Agent component on the remote nodes.

5.1. 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:

1. Copy the Unified Wire Installer (ChelsioUwire-x.x.x.xx.exe) to a shared location accessible to all the nodes.
2. 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
```

3. 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

4. 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.*

5.2. 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.**

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.*

5.2.1. Deploying UM components and Flashing Option ROM together

1. 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>
```

2. Click the **Next** button for the Chelsio End User License Agreement Window.

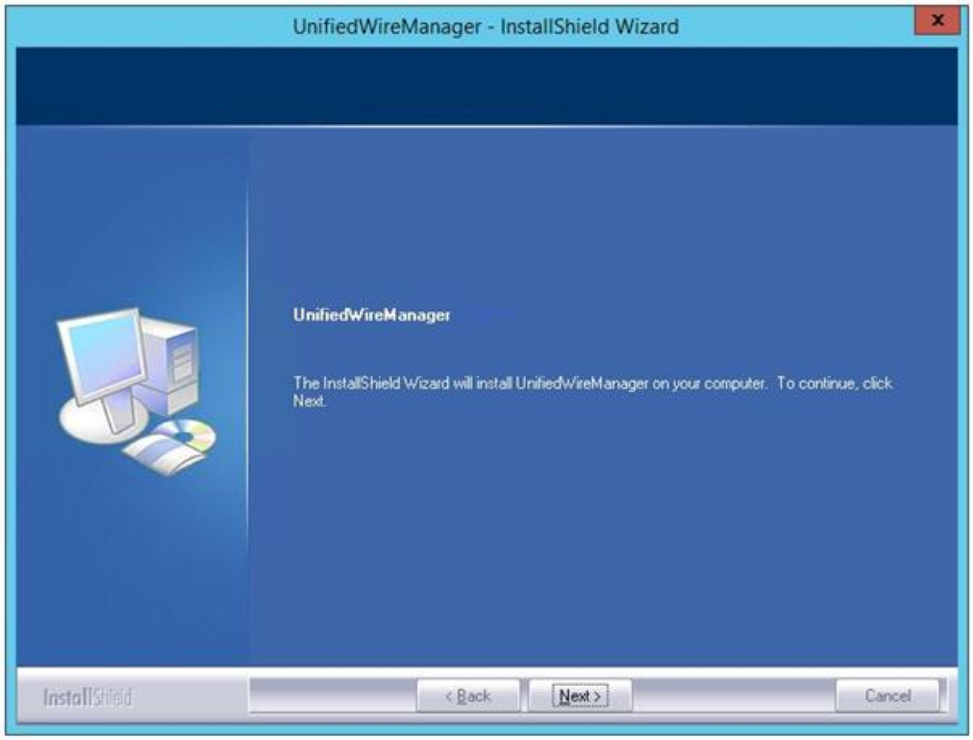

Figure 20 - UM Installer welcome window

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

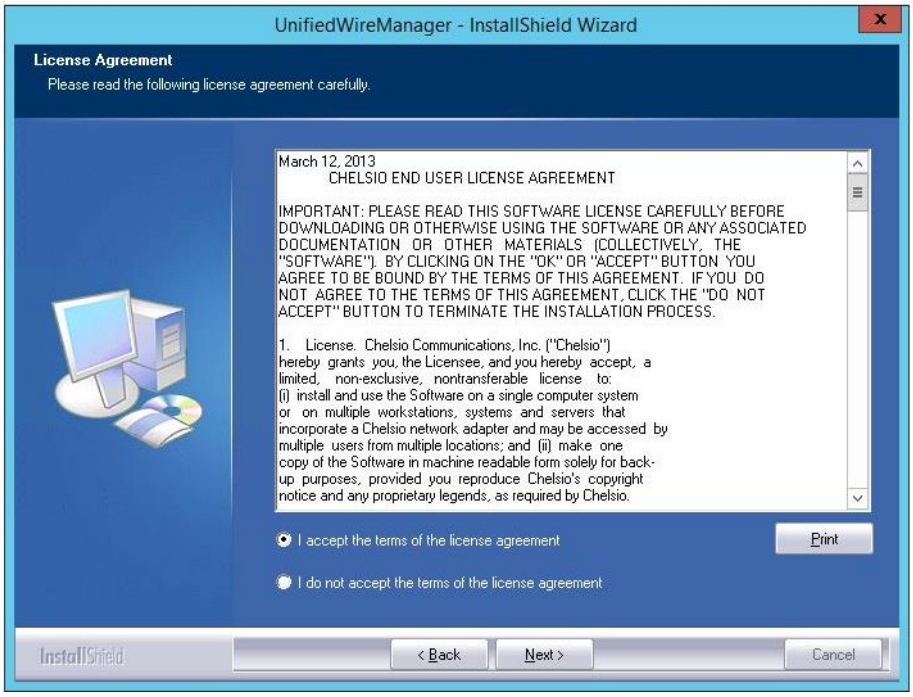

Figure 21 - Chelsio EULA Window

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

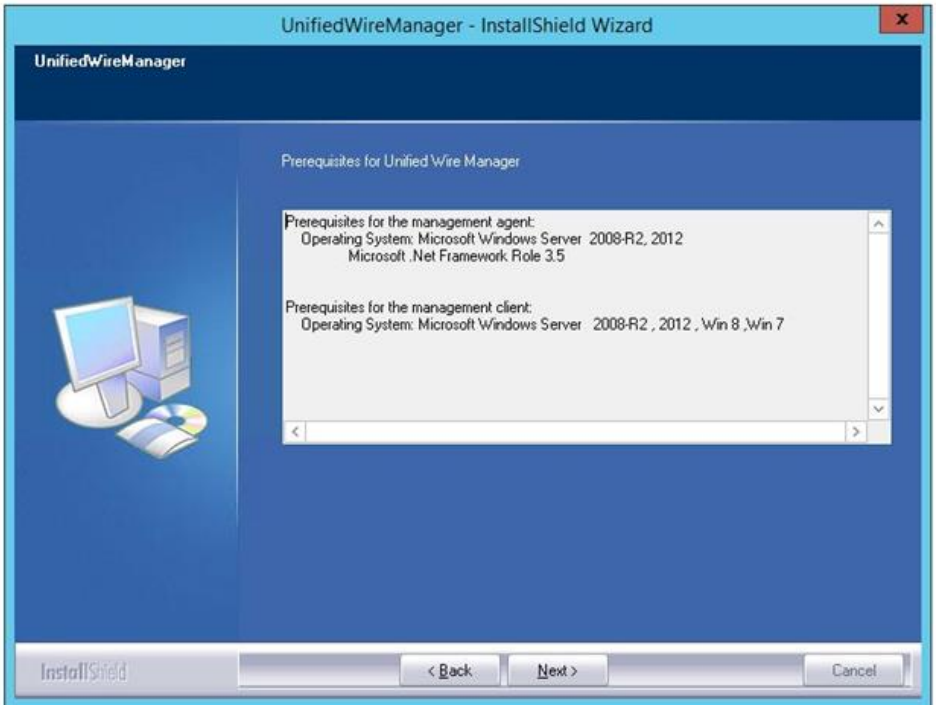

Figure 22 - UM prerequisites window

5. Now, select **Custom** radio button to customize the installation. Click **Next.**

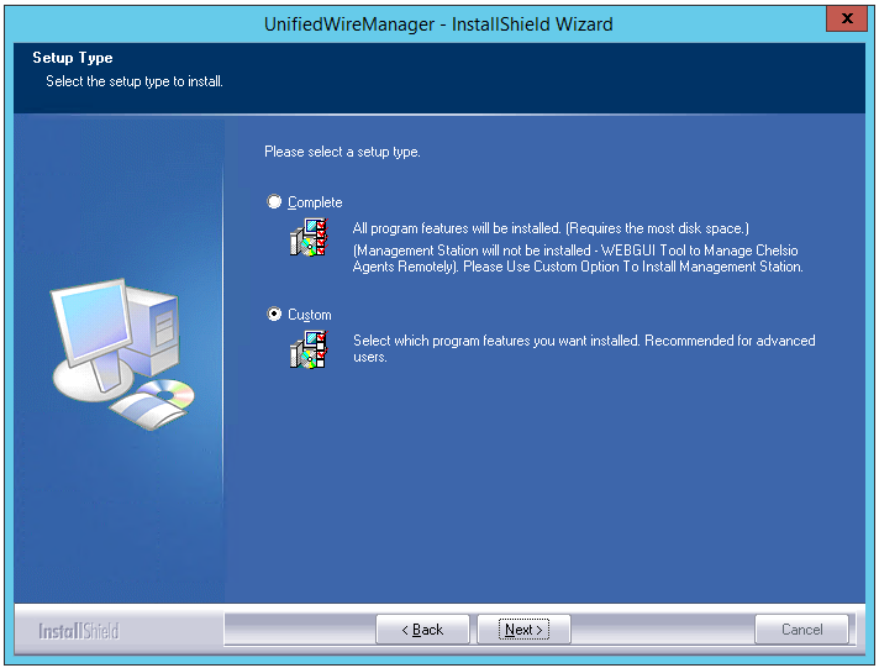

Figure 23 - UM prerequisites window

i. 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.

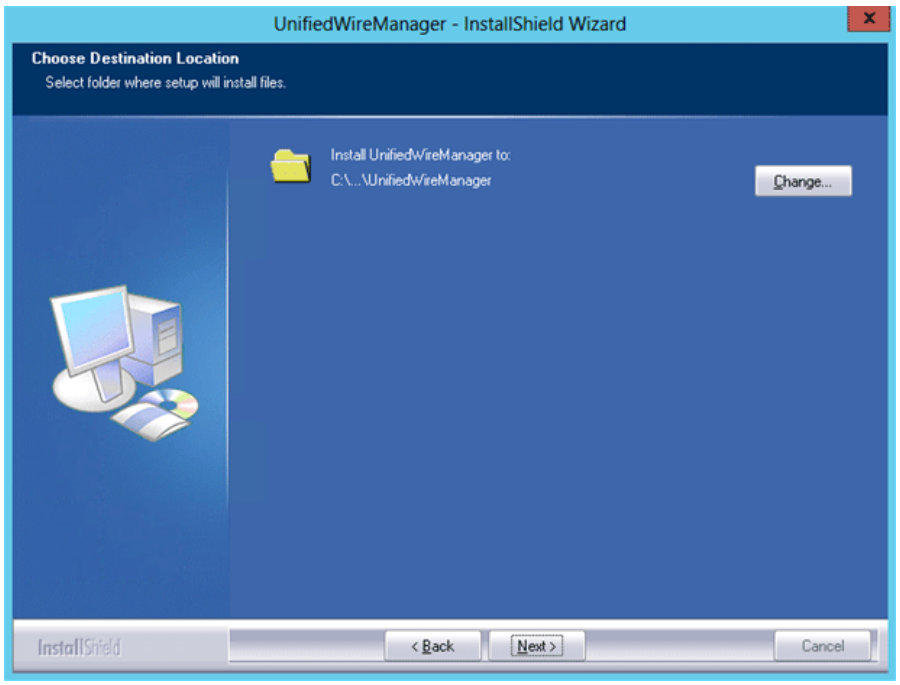

Figure 24 - Changing UM installation path

ii. Next, ensure that only **Client** is selected under features to be installed. Click **Next.**

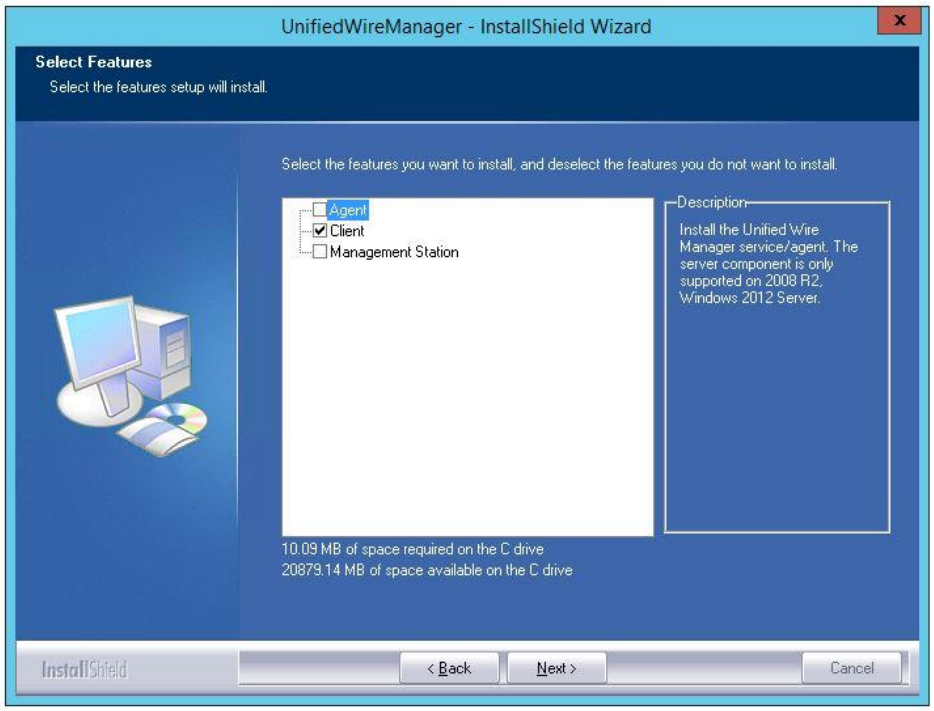

Figure 25 - Selecting UM components to install

6. Click **Finish** to complete Client installation.

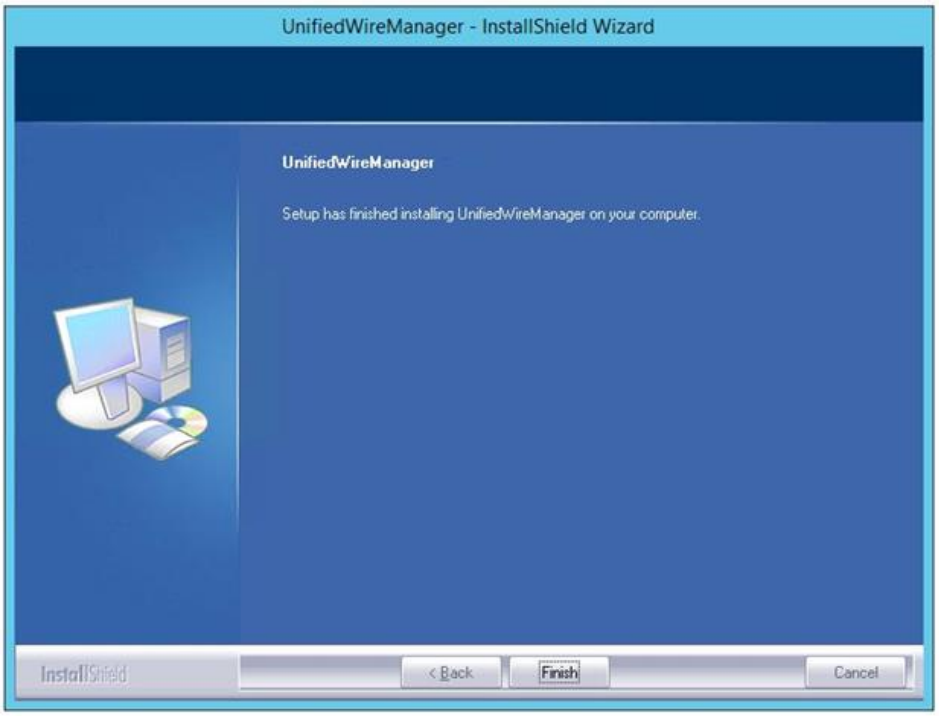

Figure 26 - Finishing UM installation

7. Now, installation of UM Agent component on remote nodes will commence.

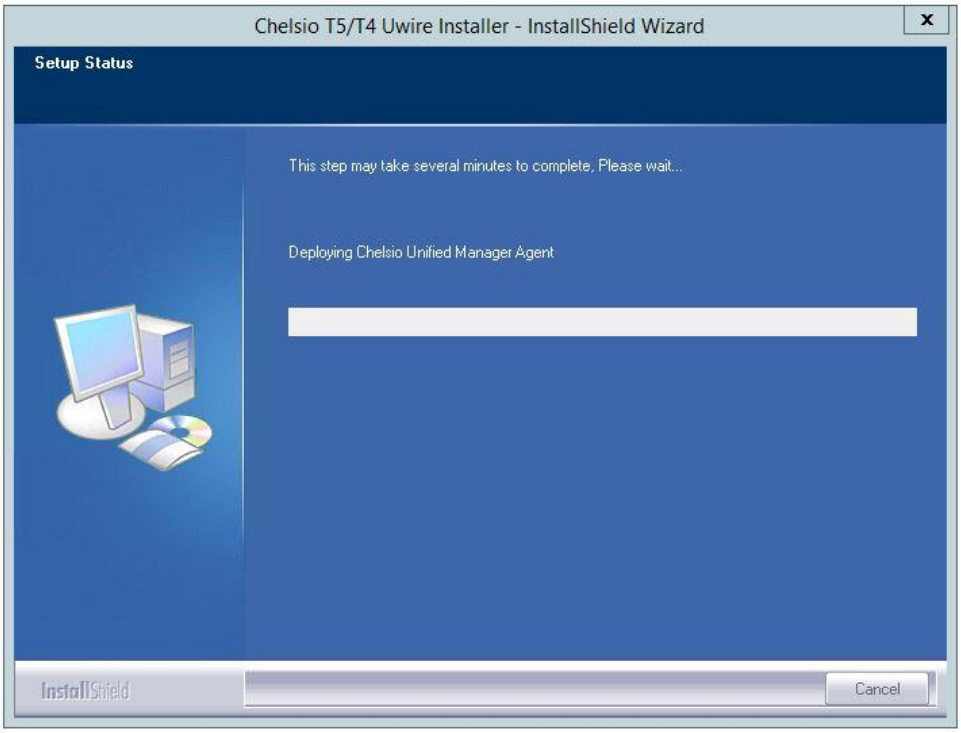

Figure 27 - Starting Agent installation

8. Click **Agree** on the **PsExec License Agreement** Window that appears. This window will appear only during first installation.

Figure 28 - PsEXec License Agreement

9. 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*.

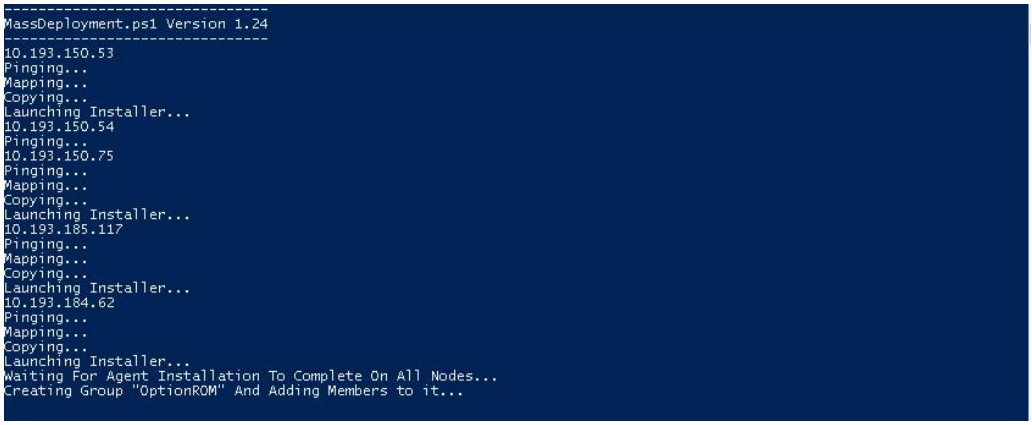

Figure 29 - Agent installation on remote nodes

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.*

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

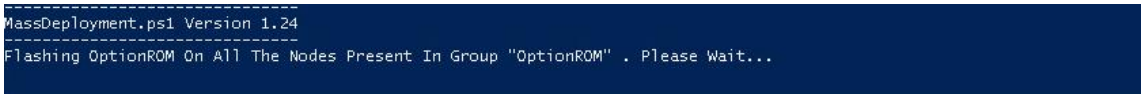

Figure 30 - Flashing option ROM

11. 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.

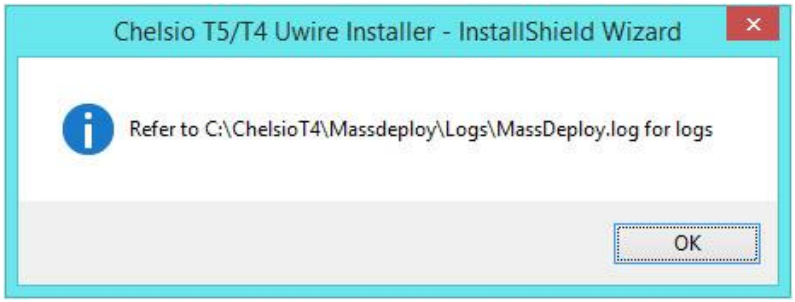

Figure 31 - Finishing option ROM flashing

5.2.2. Deploying UM components and Flashing Option ROM separately

- **Deploying UM Components**

1. 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>
```

2. Follow steps 2-9 in the **Deploying UM components and Flashing Option ROM together** section.

3. 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.

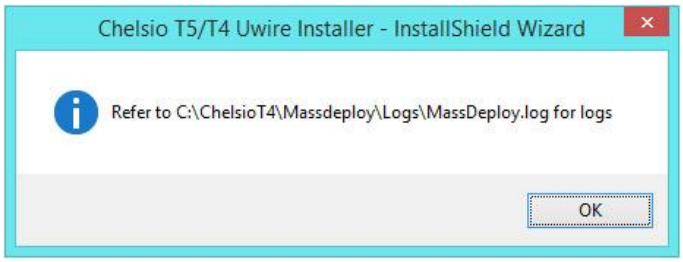

Figure 32 - Finishing UM components deployment

- **Flashing Option ROM**

1. Run the following command to flash Option ROM:

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

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

Figure 33 - Starting option ROM flashing

3. Follow steps 10 and 11 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.*

5.3. 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.*

1. To erase Option ROM from adapters on all the nodes, run the following command:

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

2. Erasing Option ROM is possible only if the adapters were flashed using this installer. Click **Yes** to continue.

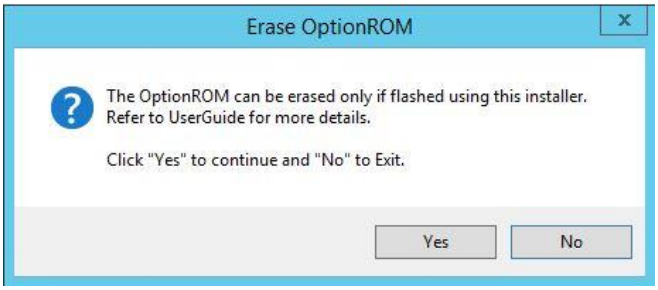

Figure 34 - Starting option ROM erasing

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

Figure 35 - Erasing option ROM on remote nodes

4. 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.

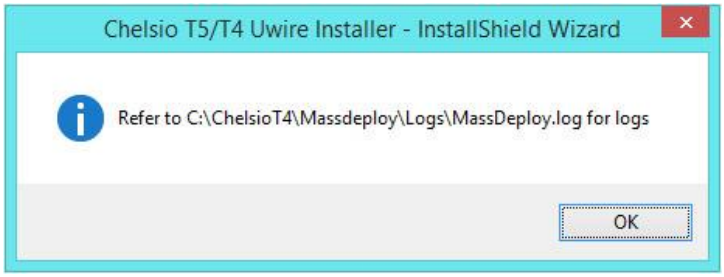

Figure 36 - Finishing option ROM erasing

5.4. Configuring using UM CLI

5.4.1. 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
```

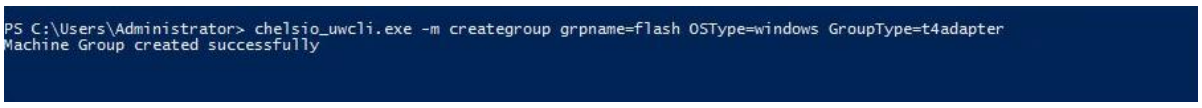

Figure 37 - Creating group

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,*,*,*,*,*>”
```

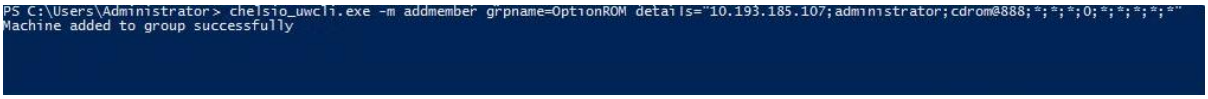

Figure 38 - Adding member to group

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
```

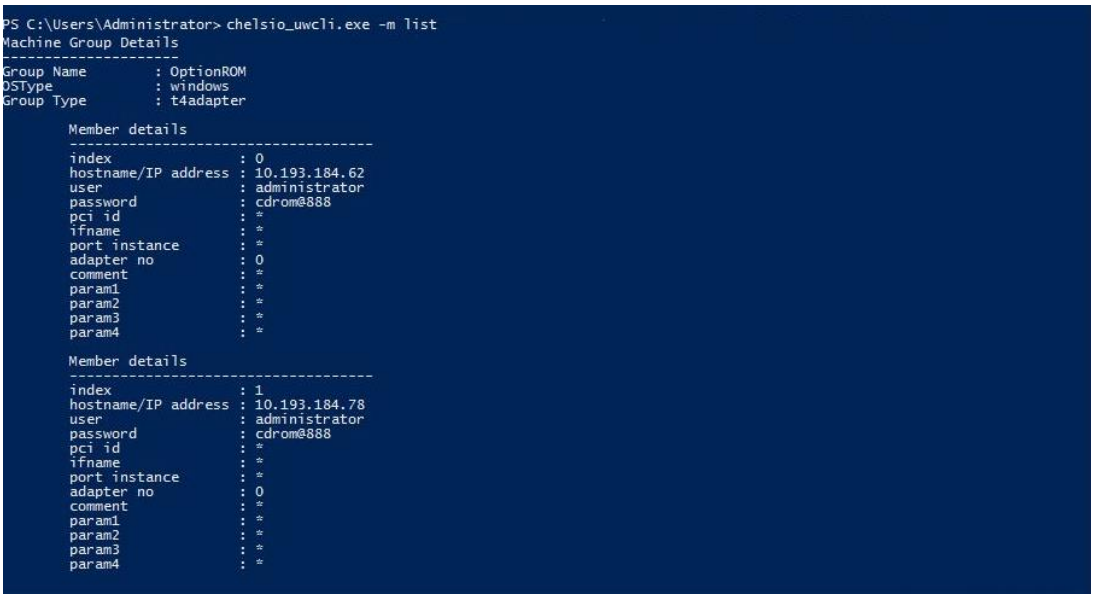

Figure 39 - Viewing group and member details

- **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
```

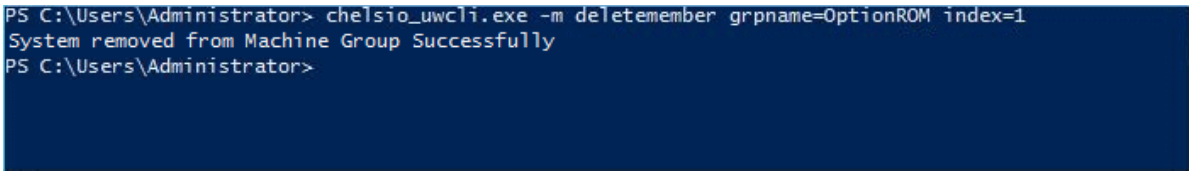

Figure 40 - Deleting member from group

- **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
```

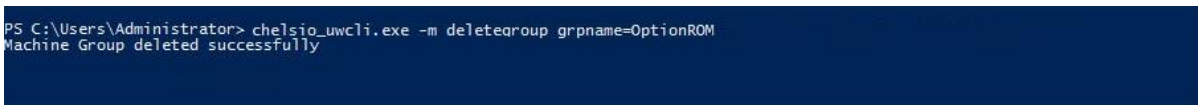

Figure 41 - Deleting group

5.4.2. 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
```

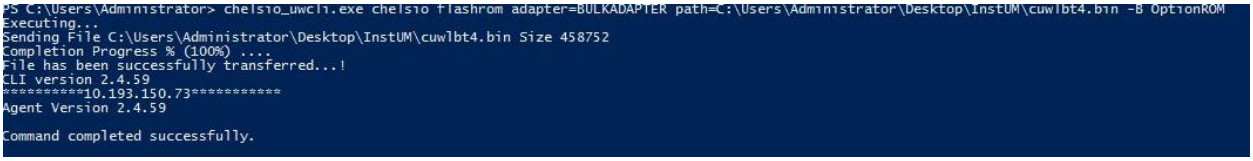

Figure 42 - Flashing option ROM using UM CLI

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
```

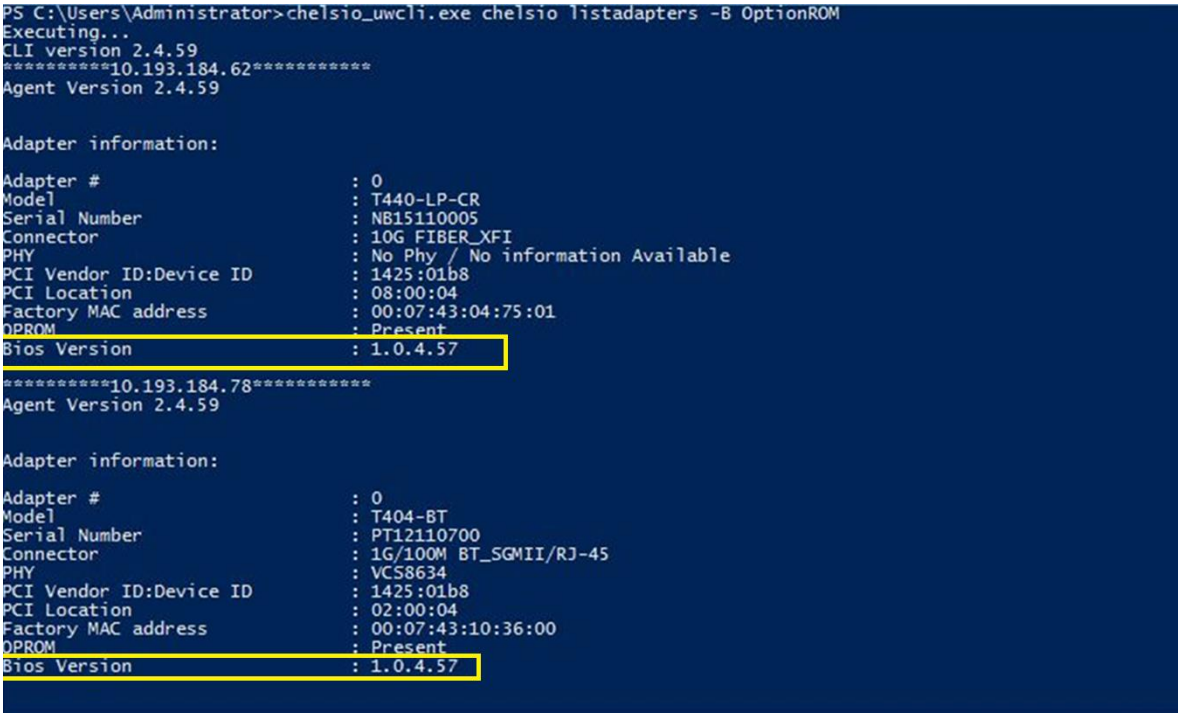

Figure 43 - Viewing status

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 -
```

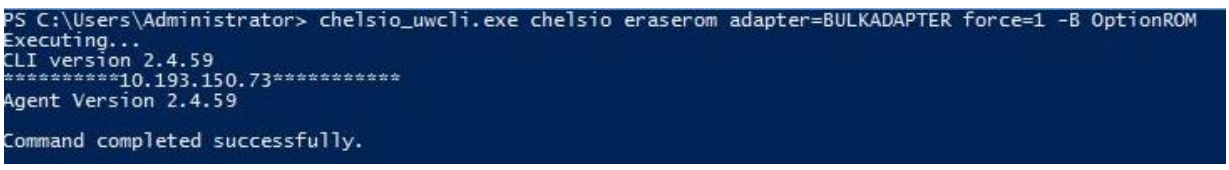

Figure 44 - Erasing option ROM using UM CLI

5.4.3. Help

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

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

6. *cxgbtool* help

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

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

Definitions

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

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

Syntax

cxgbtool [*vbdInterface*|*nicInterface*][*parameters*][(*optionalParameters*)]

- **-h**

Description: Displays help

Syntax: cxgbtool -h

- **cim_la**

Decription: Displays results of logic analyzer trace.

Syntax: cxbtool [*vbdInterface*] cim_la

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cim_la
Status   Data      PC    LS0Stat   LS0Addr            LS0Data
.
.
c0   00000001 1ffffc4a 00000000 00000000 00000000000000000000000000000000
c0   00000001 1ffffc4a 00000000 00000000 00000000000000000000000000000000
fe   1ffffc4a 1ffffc4a 00000000 00000000 00000000000000000000000000000000
c0   00000001 1ffffc4d 00000000 00000000 00000000000000000000000000000000
fe   1ffffc4d 1ffffc4d 00000000 00000000 00000000000000000000000000000000
.
.
```

context

Description: Shows an SGE context.

Syntax: cxgbtool [*vbdInterface*] context [*contextType*] [*queueId*]

Context Type Parameters:

Egress: Egress queue context.
fl: Free list manager context.
response: Response queue context.
ingress: Ingress queue context.
cq: RDMA completion queue context.
cong: Congestion context.

Example:

```
C:\Users\Administrator>cxgbtool vbd0 context egress 0
Response Data:
00000000    00000000 00000000 00000000 00000000
00000010    00000000 00000006 00000074 00000000
00000020    00000001 00000050 00000000 00000000
00000030    00000000 00000000 00000000 00000000
00000040    00000000 00000000 00000000 00000000
00000050    00000000 00000000 00000000 00000000
00000060    00000000 00000000 00000000 00000000
00000070    00000000 00000000 00000000 00000000
In show_t4_ctxt: p->mem_id = 0
StatusPgNS:        0
StatusPgRO:        0
FetchNS:           0
FetchRO:           0
Valid:             1
PCIeDataChannel:   0
DCAEgrQEn:         0
DCACPUID:          0
FCThreshOverride:  0
.
.
.
```

filter

Description: Displays list of configured hardware filters

Syntax: cxgbtool [*nicInterface*] filter

Example:

```
C:\Users\Administrator>cxgbtool nic0 filter
Ftid     Prot     FPORT    LPORT    Type     Que_Id   DMAC_Idx  Locked   Hits
```

loadfw

Description: Loads firmware image.

Syntax: cxgbtool [*vbdInterface*] loadfw [*firmwareImage*]

Note *The Firmware input file used must be a binary and not a header file.*

Example:

```
C:\Users\Administrator>cxgbtool vbd0 loadfw t5fw-1.11.18.0.bin
```

loadcfg

Description: Loads firmware configuration file or clears configuration flash region.

Note *The configuration file used must be in text format and not a header file.*

Syntax: cxgbtool [*vbdInterface*] loadcfg [[*firmwareConfigurationFile*]|clear]

Example:

Loading T5 firmware configuration file:

```
 C:\Users\Administrator>cxgbtool vbd0 loadcfg t5-config.txt
 Flashing configuration file C:\Users\Administrator\Desktop\t5-config.txt
of   size 22607 ...
```

Clearing the configuration file region in flash

```
C:\Users\Administrator>cxgbtool vbd0 loadcfg clear
Clearing configuration file in flash...
```

loadphy

Description: Loads phy firmware.

Syntax: cxgbtool [*vbdInterface*] loadphy [*phyFile*]

Example:

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

loadboot

Description: Flashes the Option ROM image.

Syntax: cxgbtool [*vbdInterface*] loadboot [*bootImageFile*] [pf {0|1|2|…|7}|offset {*val*}|clear]

Parameters:

pf{val}: Flash Option ROM image to the offset defined by the PFs EXPROM_OFST in the serial configuration.

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.

Example:

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

```
C:\Users\Administrator>cxgbtool vbd0 loadboot cuwlbt4.bin pf 0
```

Flashing Option ROM image to user specified offset:

```
C:\Users\Administrator>cxgbtool vbd0 loadboot cuwlbt4.bin offset 0x100
```

loadboot-cfg

Description: Loads boot configuration file.

Syntax: cxgbtool [*vbdInterface*] loadboot-cfg [[*bootConfigImageFile*]|clear]

Parameters:

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

Example:

```
C:\Users\Administrator>cxgbtool vbd0 loadboot-cfg .\boot-config.bin
cxgb_CmdLoadBootCfg: BootROM length: 1664, value2add:0
```

mdio

Description: Reads/writes MDIO register.

Syntax: cxgbtool [*vbdInterface*] mdio
[*physicalAddress*][*manageableDevicesAddress*][*registerAddress*] [(*writeValue*)]

Example

Read MDIO register

```
C:\Users\Administrator>cxgbtool vbd0 mdio 0 0 0
```

Write MDIO register

```
C:\Users\Administrator>cxgbtool vbd0 mdio 0 0 0 0
```

meminfo

Description: Displays memory info.

Syntax: cxgbtool [*vbdInterface*] meminfo

Example:

```
C:\Users\Administrator>cxgbtool vbd0 meminfo
EDC0:          0-0x2fffff [3.00 MiB]
EDC1:          0x300000-0x5fffff [3.00 MiB]

RQUDP region:  0xffffffff-0xfffffffe [0 B]
iSCSI region:  0xffffffff-0xfffffffe [0 B]
Tx payload:    0-0x17ffffff [384 MiB]
Rx payload:    0-0x7affffff [123 MiB]
IMSG contexts: 0x17b280-0x23b27f [768 KiB]
ULPTX state:   0x23b280-0x2443bf [36.3 KiB]
ULPRX state:   0x2443c0-0x248bbf [18.0 KiB]
Pstructs:      0x248bc0-0x2c89bf [511 KiB]
Rx FL:         0x2c89c0-0x2c9e3f [5.12 KiB]
.
.
.
```

mtus

Description: Prints hardware MTU table.

Syntax: cxgtool [vbdInterface] mtus

Example:

```
C:\Users\Administrator>cxgbtool vbd0 mtus
Path mtus : 88 256 512 576 808 1024 1280 1488 1500 2002 2048 4096 4352 8192
9000 9600
```

qsets

Description: Reads # of qsets

Syntax: cxgbtool [*nicInterface*] qsets

Example:

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

QueType    AbsId      RelId      Fl0Id      Fl1Id      Msix       QDepth

TxEth      8          8          n/a        n/a        n/a        1024
TxEth      9          9          n/a        n/a        n/a        1024
TxEth      10         10         n/a        n/a        n/a        1024
TxEth      11         11         n/a        n/a        n/a        1024
TxCtrl     12         12         n/a        n/a        n/a        1024
TxRdma     13         13         n/a        n/a        n/a        1024
RxIng      1          1          0          0          6          1023
RxEth      2          2          0          0          8          3071
RxEth      3          3          1          0          2          3071
RxEth      4          4          2          0          4          3071
RxEth      5          5          3          0          14         3071
RxRdma     6          6          4          0          16         511
RxRdma     7          7          5          0          10         511
RxRdma     8          8          6          0          12         511
RxRdma     9          9          7          0          22         511
```

qstats

Description: Displays statistics for each Tx & Rx queue.

Syntax: cxgbtool [*nicInterface*] qstats [*queueType* [(clr)]]

Queue Type parameters:
txeth: Tx tunnel queue statistics.

rxeth: Rx tunnel queue statistics.
txvmq: Tx VM queue statistics.
rxvmq: Rx VM queue statistics.
txtoe: Chimney Tx queue statistics.
rxtoe: Chimney Rx queue statistics.
txrdma: RDMA Tx queue statistics.
rxrdma: RDMA Rx queue statistics.
txctrl: Chimney control queue statistics.
txfwd: Chimney forwarding queue statistics.
txnvgre: Tx NVGRE statistics.
rxnvgre: Rx NVGRE statistics.
clr: Clear Queue statistics.

Example of rxeth qstats:

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

StatsType   RxEth#2     RxEth#3     RxEth#4     RxEth#5

UcPkts      31          27          81          0
UcBytes     2980        3216        10860       0
McPkts      579         612         543         0
McBytes     43866       45371       42950       0
BcPkts      29          102         1422        0
BcBytes     2472        24786       131544      0
CoalPkts    0           0           0           0
CoalBytes   0           0           0           0
DropPkts    0           0           0           0
RecvNbls    639         741         2046        0
RxCsumGood  600         735         2025        0
VLANex      0           0           0           0
RssProc     1           2           3           1
RssGroup    0           0           0           0
```

- ## reg

Description: Reads/writes register.

Syntax:

Register Read: cxgbtool [*vbdInterface*] reg [*readAddress*]

Register Write: cxgbtool [*vbdInterface*] reg [[*writeAddress*] = {val}]

Example:

Register read:

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

Register write:

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

regdump

Description: Displays registers of a hardware module. Not specifying any parameter will display registers for all the modules. Not all modules are available on all adapters.

Syntax:

Display registers for all available modules: cxgbtool [*vbdInterface*] regdump

Display registers for specific module: cxgbtool [*vbdInterface*] regdump [*registerModule*]

Register module parameters:

Module parameter	Description
sge	Scatter-Gather DMA Engine common register set.
pci	PCI Express Interface common register set. This module implements the PCI-Express SR-IOV physical logical, data link, and transaction layers.
dbg	Debug Engine Common register set.
mc0	Memory controller 0 common register set. This module implements the memory controller for the optional external DDR-II/DDR-III SDRAM.
mc1	Memory controller 1 common register set. This module implements the memory controller for the optional external DDR-II/DDR-III SDRAM.
ma	Memory Arbiter common register set. This module implements the arbitration of memory requests from the various on-chip sources to the memory hierarchy consisting of on-chip eDRAM, external DDR2/DDR3 memory, and host memory that is accessed through the PCIe.
edc0	eDRAM and Controller 0 common register set. This is the on-chip eDRAM and controller.
edc1	eDRAM and Controller 1 common register set. This is the on-chip eDRAM and controller.
cim	CIM common register set. This module implements the CPU interface and µP is the embedded microprocessor. The CIM incorporates functions to improve the performance of CPU accesses to external memory.
tp	Transport Protocol Engine common register set. This module implements the main packet processing pipeline.
ulp_rx	Ingress Upper Layer Protocol common register set. This module implements the upper layer protocol processing in the ingress direction for protocols that are layered on top of TCP, such as iSCSI and RDMA.

ulp_tx	Egress Upper Layer Protocol common register set. This module provides Upper Layer support for RDMA and iSCSI offload in the transmit direction, and also implements LSO/TSO functionality.
pmrx	Ingress Payload Manager common register set. These modules implement the payload manager for receive/ingress.
pmtx	Egress Payload Manager common register set. These modules implement the payload manager for transmit/egress.
mps	Multi-port support common register set. This module implements the multi-port support for T4, and switches egress packets to the ingress path when their Ethernet DA (Destination Address) matches an address in the exact match Ethernet Address database, or if the outer-VLAN indicates that the packet is destined to another virtual machine connected to the T4, or if another virtual machine is subscribing to an L2 multicast group that is the MAC destination address of the packet.
cplsw	CPL Switch common register set. This module implements a configurable switch for ingress CPL messages to the SGE and/or CIM. The embedded µP is assigned a receive queue number and can be assigned one or more MAC addresses and IP addresses, and any of these can be used to switch ingress packets to the µP for processing.
smb	System Management Bus common register set. This module implements an SMBus Master/Slave for system management.
i2c	I2C-Bus Master common register set. This module implements an I2C-Bus Master for PHY management and I/O expansion.
mi	MI common register set. This module implements an MDIO Master for PHY management.
uart	
pmu	Power Management Unit common register set. Power management unit.
sf	Serial Flash controller common register set. This module implements the serial flash controller. It interfaces to the external serial flash device.
pl	PIO Local Bus controller common register set. This module implements the PIO Local Bus controller. It is physically distributed across T4/T5.
le	Lookup Engine Common register set. The Lookup Engine implements the connection, filter and ACL databases. This module includes a TCAM Memory controller that interfaces with the on-chip TCAM memory array, and it implements the interface to the external memory that is used to scale the support of the connection database to 1M connections. The LE stores ACL rules, it stores routing information to handle routing for SYN-cookie mode offloaded listening servers, and it stores tuple information for offloaded connections, and FCoE exchanges.
ncsi	Network Controller sideband Interface common register set. The module implements the NCSI (Network Controller Sideband Interface) protocol.
xgmac	
mac	MAC common register set.
hma	

T4 Register Modules	T5 Register Modules
sge	sge
pci	pci
dbg	dbg
mc	mc0
ma	mc1
edc0	ma
edc1	edc0
cim	edc1
tp	cim
ulp_rx	tp
ulp_tx	ulp_rx
pmrx	ulp_tx
pmtx	pmrx
mps	pmtx
cplsw	mps
smb	cplsw
i2c	smb
mi	i2c
uart	mi
pmu	uart
sf	pmu
pl	sf
le	pl
ncsi	le
xgmac	ncsi
	mac
	hma

Example:

Displaying all available module registers:

```
C:\Users\Administrator>cxgbtool vbd0 regdump
[0x1e000] SGE_PF_KDOORBELL                                  0          0
    31:15 QID                                               0          0
    14:14 Priority                                          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

.
.
.
```

Displaying specific (*mps)* module registers:

```
C:\Users\Administrator>cxgbtool vbd0 regdump mps
[ 0x9000] MPS_CMN_CTL                                   0x1         1
    4:4 LpbkCrdtCtrl                                    0           0
    3:3 Detect8023                                      0           0
    2:2 VFDirectAccess                                  0           0
    1:0 NumPorts                                        0x1         1

.
.
.
```

- **tcb**

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

Syntax: cxgbtool [*nicInterface*] tcb [*tid*]

Example:

```
C:\Users\Administrator>cxgbtool nic0 tcb 5070
HW TCB
ulp_type                : 0x7
ulp_raw                 : 0x0
l2t_ix                  : 0xf40
smac_sel                : 0x36
tf_suspend              : 0x0
tf_non_offload          : 0x0
tf_lock_tid             : 0x0
tf_keepalive            : 0x0
tf_dack                 : 0x0
tf_dack_mss             : 0x0
tf_dack_not_acked       : 0x0
tf_nagle                : 0x0
.
.

.
```

tpi

Description: Displays TP indirect registers.

Syntax: cxgbtool [*vbdInterface*] tpi

Example:

```
C:\Users\Administrator>cxgbtool vbd0 tpi
TP_PIO
[   0x20] TP_RX_SCHED_MAP                          0x55555555 1431655765
    31:24 S_RXMAPCHANNEL3                          0x55       85
    23:16 S_RXMAPCHANNEL2                          0x55       85
    15:8 S_RXMAPCHANNEL1                           0x55       85
     7:0 S_RXMAPCHANNEL0                           0x55       85
 [   0x21] TP_RX_SCHED_SGE                          0xf        15
    15:12 S_RXSGEMOD1                              0          0
     11:8 S_RXSGEMOD0                              0          0
      3:3 S_RXSGECHANNEL3                          0x1        1
 .
 .
 .
```

sgedbg

Description: Displays sge debug indirect registers.

Syntax: cxgbtool [*vbdInterface*] sgedbg

Example:

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

SGE_DEBUG_DATA_HIGH00                          0x3000000  50331648
SGE_DEBUG_DATA_HIGH01                          0          0
SGE_DEBUG_DATA_HIGH02                          0          0
SGE_DEBUG_DATA_HIGH03                          0          0
SGE_DEBUG_DATA_HIGH04                          0          0
SGE_DEBUG_DATA_HIGH05                          0          0
SGE_DEBUG_DATA_HIGH06                          0          0
SGE_DEBUG_DATA_HIGH07                          0          0
SGE_DEBUG_DATA_HIGH08                          0x26020000 637665280
SGE_DEBUG_DATA_HIGH09                          0          0
SGE_DEBUG_DATA_HIGH10                          0x3        3
SGE_DEBUG_DATA_HIGH11                          0          0
.
.
.
```

dumpctx

Description: Displays Adapter Context.

Syntax: cxgbtool [*nicInterface*] dumpctx

Example:

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

Type            Value

Link State      Connected
Link Speed      10 Gbps
Hw Addr         00:07:43:04:7d:60
Cur Addr        00:07:43:04:7d:60
Port No         0
NetIfIdx        18
Mtu             1500
.
.
.
```

version

Description: Displays the Adapter Part Number, Serial Number, Device ID, Firmware Version, TP, NDIS Driver version, and VBD version.

Syntax: cxgbtool [*vbdInterface*] version

Example:

```
C:\Users\Administrator>cxgbtool vbd0 version
P/N: 11011605002
S/N: RE14130227
NIC DeviceID 5401 (T520-CR)
NIC FW Version 1.14.4.0
NIC TP Version 0.1.4.8
Ndis Driver Version 5.5.11.0
VBD Version 5.5.11.0
```

fwtoc

Description: Converts Firmware binary file to source file header.

Syntax: cxgbtool [*vbdInterface*] fwtoc [*firmwareFile*] filename= [C*FileName*].bin

Example:

```
C:\Users\Administrator>cxgbtool vbd0 fwtoc .\Chelsio-Uboot-
1.0.0.68\t4fw- 1.12.14.0.bin filename=t4fw-1.12.14.0.bin
```

- ## inst

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

Syntax: cxgbtool [*vbdInterface*] inst [*infFilePath*]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 inst .\chnetx64.inf
ENTER:  DriverPackageInstallA
ENTER:  DriverPackageInstallW
Installing INF file '.\chnetx64.inf' (Plug and Play).
Looking for Model Section [Chelsio.NTamd64.6.2]...
Installing devices with Id "CHT5BUS\chnet" using
INF     "C:\Windows\System32\DriverStore\FileRepository\chnetx64.inf_amd64_
dd8d3a55aa5c4536\chnetx64.inf".
Installation did not occur because the current driver on the device is
the   same or better.
No drivers installed. Drivers contained
in   'C:\Windows\System32\DriverStore\FileRepository\chnetx64.inf_amd64_dd8d3
a55a a5c4536\chnetx64.inf' are not better than current one's.
RETURN: DriverPackageInstallW (0x103)
RETURN: DriverPackageInstallA (0x103)
```

- ## uninst

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

Syntax: cxgbtool [*vbdInterface*] uninst [*infFile*]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 uninst .\chnetx64.inf
Removing device instance CHT<4|5>BUS\CHNET\5&39B01C76&1&0000
```

- ## update

Description: Updates driver package.

Syntax: cxgbtool update [*infFile*]

Example:

```
C:\Users\Administrator>cxgbtool update C:\ChelsioT4\driver\2k12\amd64\
Driver for HwId:CHT5BUS\CHNET updated successfully!!!
Driver for HwId:PCI\VEN_1425&DEV_5610&SUBSYS_00001425&REV_00 updated
successfully!!!
Driver update for HwId:PCI\VEN_1425&DEV_5409&SUBSYS_00001425&REV_00 failed
No more data is available.
```

- ## rescan all

Description: Scans for hardware changes in the device manager.

Syntax: cxgbtool [*vbdInterface*] rescan all

Example:

```
C:\Users\Administrator>cxgbtool vbd0 rescan all
```

- ## trace

Description: Enables driver debug prints.

Syntax: trace [*nicInterface*|*vbdInterface*] trace [nic|vbd] [(flags)][(level)]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 trace vbd 8 7
Tracing vbd flags=0x8 level=7
Press 'q' to quit tracing:
```

- ## seeprom

Description: Reads/Writes SEEPROM (init+VPD) data.

Syntax:

Read SEEPROM data to a bin file: cxgbtool [*vbdInterface*] seeprom read

Display SEEPROM data in console: cxgbtool [*vbdInterface*] seeprom dump

Write SEEPROM data: cxgbtool [*vbdInterface*] seeprom write [*seepromFile*]

Verify SEEPROM data: cxgbtool [*vbdInterface*] seeprom verify [*seepromFile*]

Example:

Read SEEPROM data:

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

Write SEEPROM data:

```
C:\Users\Administrator>cxgbtool.exe vbd0 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 T580-LP-CR
(SN:PT02141264,PN:110116450B0,NA:000743291420) card? (y/n) : y
Hardware configuration changed successfully.
Please reboot for the changes to take effect
```

Verify SEEPROM data:

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

verification started
Verification : PASSED
```

- ## serialinit

Description: Reads/Writes serialinit data.

Syntax:

Read serialinit data to a bin file: cxgbtool [*vbdInterface*] serialinit read

Display serialinit data in console: cxgbtool [*vbdInterface*] serialinit dump

Write serialinit data: cxgbtool [*vbdInterface*] serialinit write [*initdataFile*]

Verify serialinit data: cxgbtool [*vbdInterface*] serialinit verify [*initdataFile*]

Example:

Read serialinit data

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

Write serialinit data

```
C:\Users\Administrator>cxgbtool vbd0 serialinit write C:\ChelsioT4\Adapter
Configuration\SRIOV\t520_cr_init_gen3_250_825_Pf4vf.bin
Changing the init/vpd can cause the card to become inaccessible if the
operation is interrupted
Do you want to flash your T520-CR
(SN:PT05140526,PN:110116050D0,NA:0007432962F0) card? (y/n): y
Hardware configuration changed successfully.
```

Verify serialinit data

```
C:\Users\Administrator>cxgbtool vbd0 serialinit verify C:\ChelsioT4\Adapter
Configuration\SRIOV\t520_cr_init_gen3_250_825_Pf4vf.bin
Verifying Serial Init data against C:\ChelsioT4\Adapter
Configuration\t520_cr_init_gen3_250_825_Pf4vf.bin
verification started
Verification: PASSED
```

- ### vpd

Description: Reads/writes VPD data.

Syntax:

Read VPD data: cxgbtool [*vpdInterface*] vpd

Write VPD data to serial EEPROM on chip: cxgbtool [*vpdInterface*] vpd [*vpdFile*]

Example:

Read VPD Data:

```
C:\Users\Administrator>cxgbtool vbd0 vpd
```

Write VPD Data:

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

vpdparams

Description: Displays the adapters Serial Number, Device ID, EC, Core Clock, Part Number, and Network Address (MAC).

Syntax: cxgbtool [*vbdInterface*] vpdparams

Example:

```
C:\Users\Administrator>cxgbtool vbd0 vpdparams
Serial Number  : RE14130227
Id             : T520-CR
Ec             : 0000000000000000
Core Clock     : 250000
Pn             : 11011605002
Network Addr   : 000743047D60
```

- **wtp**

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

Syntax: cxgbtool [*vbdInterface*] wtp

Example:

```
C:\Users\Administrator>cxgbtool vbd0 wtp
LOW    0x00000000
HIGH   0x00aa0000
LOW    0x0003f000
HIGH   0x00660022
LOW    0x00000000
HIGH   0x0000eeee
LOW    0x00000000
HIGH   0x0000eeee
LOW    0x00000000
HIGH   0x00aa00aa
LOW    0x0070000b
HIGH   0x0000eeee
LOW    0x00000008
HIGH   0x0000eeee
LOW    0x00000013
HIGH   0x00550033
LOW    0x00000016
HIGH   0x0002006e
LOW    0x0000000b
HIGH   0x00880022
LOW    0x00c00001
HIGH   0x00000000
LOW    0x00000001
HIGH   0x00000000
LOW    0x00000000
HIGH   0x00000000
LOW    0x00000000
HIGH   0x00000000
LOW    0x01534745
HIGH   0x45475301
LOW    0xfeacb8ba
HIGH   0xbab8acfe
ifaces = nic0 nic1
************************EGGRESS (TX) PATH *********************************
MOD :  core---->PCIE---->SGE<-|    #Ring Doorbell
SOP        ?      ???         |
EOP        ?      ???         |
MOD |<-core<----PCIE<----SGE<-|    #Request Work Request
SOP |    1fe        a
EOP |     ?        a
MOD |->core---->PCIE---->SGE------>CIM/uP->| #Work Request to uP
```

```
SOP      1fe        a       0              |
EOP      1fe        a       0              |
MOD |<-core<----PCIE<----SGE<------UTX<----|    #Request DATA
SOP |    3fc        2      1c
EOP |      ?       2      1c
MOD |->core-->PCIE-->SGE-->UTX---->TPC------->TPE---->MPS---->XGMAC--->wire
SOP      3fc     1c    1c    19 (19) 19   ff  12 (182) 182   3fc   fffffffc
EOP      3fc     1c    1c    19 (19) 19   ff  12 (182) 182   3fc   fffffffc
DROP: ???      ???      ???      ???       0        0       ???
************************INGRESS (RX) PATH *********************************
MOD   core<-PCIE<---SGE<--CSW<-----TPC<-URX<--TPE<-----MPS<----XGMAC<---wire
SOP          3fc   1c     6    5   19    2    2    2  92  92   3fc    fffffc
EOP          3fc   1c     6    5   19    2    2    2  92  92   3fc
DROP: ???      ???      ???        0(mib)   0(err)  0(oflow) f8(cls)
INTS:   f<-  e     f<-  6     f<-  0     f<-  0 (PCIE<-SGE, channels 0 to 3)
```

- **stats**

Description: Prints MAC statistics for a given port or TP MIB statistics.

Syntax: cxgbtool [*vbdInterface*] stats [mac{*portInstance*}|mib]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 stats mac0
LOW    0x00000000
HIGH   0x00000066
LOW    0x0003f000
HIGH   0x00aa00ee
LOW    0x00000000
HIGH   0x0000eeee
LOW    0x00000000

HIGH   0x0000eeee
LOW    0x00000000
HIGH   0x00000000
LOW    0x0070000b
HIGH   0x0000eeee
LOW    0x00000013
HIGH   0x0000eeee
LOW    0x00000013
HIGH   0x001199cc
LOW    0x00000016
HIGH   0x00020491
LOW    0x0000000b
.
.
.
```

cpl_stats

Description: Displays CPL Request and Response Statistics for all channels.

Syntax: cxgbtool [*vbdInterface*] cpl_stats

Example:

```
C:\Users\Administrator>cxgbtool vbd0 cpl_stats
                 channel 0  channel 1  channel 2  channel 3
CPL requests:            4          3          0          0
CPL responses:           3          3          0          0
```

debugfs

Description: When used in conjunction with other options, *debugfs* displays useful information regarding Chelsio adapters.

Syntax: cxgbtool [*vbdInterface*] debugfs [option]

***debugfs* options:**

- **cctrl**

Description: Displays congestion control table.

Syntax: cxgbtool [*vbdInterface*] debugfs cctrl

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs cctrl
0:   24  108  236  268  384  492  620  724
    730  981 1004 2028 2156 4076 4480 4780     1 0.5
1:    8   36   78   89  128  164  206  241
    243  327  334  676  718 1358 1493 1593     1 0.5

.
.
.
```

- **cim_pif_la**

Description: Displays CIM PIF logic analyzer trace.

Syntax: cxgbtool [*vbdInterface*] debugfs cim_pif_la

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs cim_pif_la
Cntl ID DataBE   Addr                 Data
 01  01  f000  e1003060 400040444000404440004044400044044
 81  00  f000  e1003060 000040440000404400004044000044044
 81  00  f000  e1003060 400030444000304440003044400030044

.
.
.
```

- **cim_ma_la**

Description: Displays results of CIM MA logic analyzer trace.

Syntax: cxgbtool [*vbdInterface*] debugfs cim_ma_la

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs cim_ma_la
.
.
00000000000000000000000000000000000
00000000000000000000000000000000000

Cnt ID Tag UE       Data       RDY VLD
    0  0   0    0 00000000000000000  0   0
    0  0   0    0 00000000000000000  0   0
    0  0   0    0 00000000000000000  0   0
    0  0   0    0 00000000000000000  0   0
    0  0   0    0 00000000000000000  0   0
    0  0   0    0 00000000000000000  0   0
.
.
```

- **cim_qcfg**

Description: Displays CIM queue configuration details.

Syntax: cxgbtool [*vbdInterface*] debugfs cim_qcfg

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs cim_qcfg

 Queue      Base     Size    Thres    RdPtr    WrPtr      SOP      EOP    Avail
   TP0         0     2048        0       40       40        2        2     2048
   TP1       800     2048        0        0        0        0        0     2048
   ULP      1000     2048        0        0        0        0        0     2048
  SGE0      1800     2048        0      7e0      7e0      476      476     2048
  SGE1      2000     2048        0        0        0        0        0     2048
 NC-SI      2800     2048        0        0        0        0        0     2048
  ULP0         0     2048              130      130      805      805     2048
  ULP1       800     2048               80       80      648      648     2048
  ULP2      1000     2048                0        0        0        0     2048
  ULP3      1800     2048                0        0        0        0     2048
   SGE      2000     2048              340      340      136      136     2048
 NC-SI      2800     2048                0        0        0        0     2048
```

- **clk**

Description: Displays the core clock.

Syntax: cxgbtool [*vbdInterface*] debugfs clk

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs clk
Core clock period: 4 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: 960 us
Persist timer max: 9765440 us
Keepalive idle timer: 7031116800 us
Keepalive interval: 73240800 us
Initial SRTT: 137437100448 us
FINWAIT2 timer: 9765440 us
```

- **ddp_stats**

Description: Displays DDP(Direct Data Placement) statistics.

Syntax: cxgbtool [*vbdInterface*] debugfs ddp_stats

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs ddp_stats
Frames: 0
Octets: 0
Drops:  0
```

- **cudbg**

Description: Redirects all hardware debug logs to a file.

Syntax: cxgbtool [*vbdInterface*] debugfs cudbg [*fileName*]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs cudbg cudbg_debug
cxgbtool: This may take a while. Please be patient
cxgbtool: Writing cudbg block to file cudbg_debug... size: 12006043 bytes
cxgbtool: Done writing cudbg data to file cudbg_debug
```

- **edc0**

Description: Redirects EDC0 memory details to a file.

Syntax: cxgbtool [*vbdInterface*] debugfs edc0 [*fileName*]

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

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs edc0 edc0

cxgbtool: This may take a while. Please be patient
cxgbtool: Reading MEM_EDC0 (0) memory of size 3145728 bytes
cxgbtool: Writing memory block to file edc...
cxgbtool: Done writing memory block to file edc0
```

- **edc1**

Description: Redirects EDC1 memory details to a file.

Syntax: cxgbtool [*vbdInterface*] debugfs edc1 [*fileName*]

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

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs edc1 edc1
cxgbtool: This may take a while. Please be patient
cxgbtool: Reading MEM_EDC1 (1) memory of size 3145728 bytes
cxgbtool: Writing memory block to file edc1...
cxgbtool: Done writing memory block to file edc1
```

- **flash**

Description: Redirects Flash memory details to a file.

Syntax: cxgbtool [vbd*Interface*] debugfs flash [*fileName*]

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

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs flash 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 flash...
cxgbtool: Done writing memory block to file flash
```

- **ibq_tp**

Description: Displays CIM TP inbound queue.

Syntax: cxgbtool [*vbdInterface*] debugfs ibq_tp[{0|1}]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs ibq_tp0
000000: 078003ff ffc04102 2e8003ff 00000000
0x0010: 2effc041 00000000 00000000 00000000
0x0020: 078103ff ffc04202 2e8103ff 00000000
0x0030: 2effc042 00000000 00000000 00000000
.
.
```

- **ibq_ulp**

Description: Displays CIM ULP inbound queue.

Syntax: cxgbtool [*vbdInterface*] debugfs ibq_ulp

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs ibq_ulp
000000: 00000000 00000000 00000000 00000000
0x0010: 00000000 00000000 00000000 00000000
0x0020: 00000000 00000000 00000000 00000000
0x0030: 00000000 00000000 00000000 00000000
.
.
.
```

- **ibq_sge**

Description: Displays CIM SGE inbound queue.

Syntax: cxgbtool [*vbdInterface*] debugfs ibq_sge[{0|1}]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs ibq_sge0
000000: 82000003 0000002a 00000001 2eae1676
0x0010: 00000052 00000077 00000001 29acd250
0x0020: 00000001 29acd2b4 00000000 00000000
0x0030: 08000010 c0901005 01ca01ca 00000000
.
.
.
```

- **ibq_ncsi**

Description: Displays CIM NCSI inbound queue.

Syntax: cxgbtool [*vbdInterface*] debugfs ibq_ncsi

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs ibq_ncsi
000000: 00000000 00000000 00000000 00000000
0x0010: 00000000 00000000 00000000 00000000
0x0020: 00000000 00000000 00000000 00000000
0x0030: 00000000 00000000 00000000 00000000
.
```

- **mc**

Description: Displays MC memory.

Syntax: cxgbtool [*vbdInterface*] debugfs mc[{0|1}] [*dumpFileName*]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs mc1 mc1
cxgbtool: This may take a while. Please be patient
cxgbtool: Reading MEM_MC1 (3) memory of size 1073741824 bytes
cxgbtool: Writing memory block to file mc1...
cxgbtool: Done writing memory block to file mc1
```

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

- **mps_tcam**

Description: Displays MPS TCAM configuration.

Syntax: cxgbtool [*vbdInterface*] debugfs mps_tcam

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs mps_tcam
Idx  Ethernet address     Mask     Vld Ports PF  VF Repl P0 P1 P2 P3  ML
  0 01:80:c2:00:00:0e ffffffffffff  Y   0x3   7  92   Y   0  0  0  0  0
  1 00:07:43:29:0a:b0 ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
  2 33:33:00:00:00:01 ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
  3 01:00:5e:00:00:01 ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
  4 33:33:00:01:00:03 ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
  5 33:33:00:00:00:0c ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
  6 33:33:ff:00:00:11 ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
  7 33:33:ff:b8:3f:b8 ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
  8 01:00:5e:7f:ff:fa ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
  9 01:00:5e:00:00:fc ffffffffffff  Y   0x1   4  65   N   0  0  0  0  0
 10         -
 11         -
 12         -
 13         -
```

- **mps_trc_rd trace**

Description: Reads MPS trace filter.

Syntax: cxgbtool [*vbdInterface*] debugfs mps_trc_rd trace[*traceQueueId*]

Example:

```
 C:\Users\Administrator>cxgbtool vbd0 debugfs mps_trc_rd trace0
 tracer is capturing nic0 Tx, snap length: 0, min length: 0
 packets captured match filter
 filter
pattern:   00000000000000000000000000000000000000000000000000000000000000000
00000000000 00000000000000000000000000000000000000000000000000000000000000000
000000000000 00000000000000000000000000000000000000000000000000000000000000000
000000000/000 00000000000000000000000000000000000000000000000000000000000000000
00000000000000 00000000000000000000000000000000000000000000000000000000000000000
000000000000000 00000000000000000000000000000000000000000000000000000000000000000
0000000000@0
```

- **mps_trc_wr trace**

Description: Sets MPS trace filter.

Syntax:

cxgbtool [*vbdInterface*] debugfs 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 debugfs mps_trc_wr trace0 tx0
```

- **mbox**

Description: Displays the last command in each mailbox.

Syntax: cxgbtool [*vbdInterface*] debugfs mbox[{0|1|2|…|7}]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs mbox0
mailbox[0] owned by driver
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
```

- **obq_ulp**

Description: Displays ULP outbound queue.

Syntax: cxgbtool [*vbdInterface*] debugfs obq_ulp[{0|1|2|3}]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs obq_ulp0
000000: 00000000 00000000 00000000 00000000
0x0010: 82000002 00000022 00000001 09119152
0x0020: 00000134 00000000 00000001 168966c0
.
.
```

- **obq_sge**

Description: Displays SGE outbound queue.

Syntax: cxgbtool [*vbdInterface*] debugfs obq_sge

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs obq_sge
000000: 00000020 00000801 00000000 00000000
0x0010: 00000020 00001301 00010000 67400052
0x0020: 00001020 00001301 0000037f e0800606
0x0030: 00000020 00000801 01fd206d 00907803
.
.
.
```

- **obq_ncsi**

Description: Displays NCSI outbound queue.

Syntax: cxgbtool [*vbdInterface*] debugfs obq_ncsi

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs obq_ncsi
000000: 00000000 00000000 00000000 00000000
0x0010: 00000000 00000000 00000000 00000000
0x0020: 00000000 00000000 00000000 00000000
0x0030: 00000000 00000000 00000000 00000000
.
.
```

- **obq_sge_rx_q**

Description: Displays CIM SGE outbound queue.

Syntax: cxgbtool [*vbdInterface*] debugfs obq_sge_rx_q[{0|1}]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs obq_sge_rx_q0
000000: 00000000 00000000 00000000 00000000
0x0010: 00000000 00000000 00000000 00000000
0x0020: 00000000 00000000 00000000 00000000
0x0030: 00000000 00000000 00000000 00000000
.
.
.
```

- **pm_stats**

Description: Displays page memory statistics.

Syntax: cxgbtool [*vbdInterface*] debugfs pm_stats

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs pm_stats

                        Tx cycles    Tx bytes
Read:                           0           0
Write bypass:               53350         396
Write mem:                      0           0
Bypass + mem:                   0           0

                        Rx cycles    Rx bytes
Read:                           0           0
Write bypass:               20820         242
Write mem:                      0           0
Flush:                          0           0
```

- **tcp_stats**

Description: Displays IPv4/IPv6 TCP statistics.

Syntax: cxgbtool [*vbdInterface*] debugfs tcp_stats

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs tcp_stats

                              IP                  IPv6

OutRsts:                       0                     0
InSegs:                        0                     0

OutSegs:                       0                     0
RetransSegs:                   0                     0
```

- **tp_err_stats**

Description: Displays TP error statistics for channel 0-3.

Syntax: cxgbtool [*vbdInterface*] debugfs tp_err_stats

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs tp_err_stats

                 channel 0  channel 1  channel 2  channel 3
macInErrs:               0          0          0          0
hdrInErrs:               0          0          0          0
tcpInErrs:               0          0          0          0
tcp6InErrs:              0          0          0          0
tnlCongDrops:            0          0          0          0
tnlTxDrops:              0          0          0          0
ofldVlanDrops:           0          0          0          0
ofldChanDrops:           0          0          0          0


ofldNoNeigh:    0
ofldCongDefer:  0
```

- **tp_la**

Description: Dumps TP la.

Syntax:

cxgbtool [*vbdInterface*] debugfs tp_la

cxgbtool [*vbdInterface*] debugfs tp_la [{2|3}]

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs tp_la
TP LA Debug Mode: 3

RcfOpCodeOut: 4 State: 0 WcfState: 0 RcfOpcSrcOut: 1 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: 2 RcfRxChannel:0 RcfDataOutSrdy: 0 RxDvld: 0 RxOoDvld: 0
        RxCongestion: 0 TxCongestion: 0
CplCmdIn: 238 MpsVfVld: 0 MpsPf: 0 MpsVf: 0 SynIn: 0 AckIn: 0 FinIn: 0
        RstIn: 0 DataIn: 0 DataInVld: 0 PadIn: 0 RxBufEmpty: 1 RxDdp: 0
        RxFbCongestion: 0 TxFbCongestion: 0 TxPktSumSrdy: 1 RcfUlpType: 0
        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: 1 CpcmdSave: 0
        RxPstructsFull: 0 EpcmdDvld: 0 EpcmdFlush: 0 EpcmdTrimPrefix: 0
        EpcmdTrimPostfix: 0 ERssIp4Pkt: 0 ERssIp6Pkt: 0 ERssTcpUdpPkt: 0
        ERssFceFipPkt: 0

RcfOpCodeOut: 14 State:14 WcfState:0 RcfOpcSrcOut: 0 CRxError: 0 ERxError: 0
        SanityFailed: 0 SpuriousMsg: 0 FlushInputMsg: 0 FlushInputCpl: 0
        RssUpBit: 0 RssFilterHit: 0 Tid: 17 InitTcb: 0 LineNumber: 5 Emsg: 0
        EdataOut: 1 Cmsg: 0 CdataOut: 0 EreadPdu: 1 CreadPdu: 0 TunnelPkt: 0
        RcfPeerFin: 0 RcfReasonOut: 8 TxCchannel: 2 RcfTxChannel: 0
        RxEchannel: 0 RcfRxChannel:0 RcfDataOutSrdy: 0 RxDvld: 0 RxOoDvld: 0
        RxCongestion: 0 TxCongestion: 0

.
.
.
```

- **tid_info**

Description: Displays TID info.

Syntax: cxgbtool [*nicInterface*] debugfs tid_info

Example:

```
C:\Users\Administrator>cxgbtool nic0 debugfs tid_info
TID range: 0..1919/2048..18431, in use: 0
STID range: 1920..1951, in use: 4
ATID range: 0..4095, in use: 0
FTID range: 1952..1967
HW TID usage: 0 IP users, 0 IPv6 users
```

- **tx_rate**

Description: Displays TX rate for NIC and offload traffic.

Syntax: cxgbtool [*vbdInterface*] debugfs tx_rate

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs tx_rate
              channel 0   channel 1   channel 2   channel 3
NIC B/s:              0           0           0           0
Offload B/s:          0           0           0           0
```

- **ulprx_la**

Description: Dumps ULP RX LA.

Syntax: cxgbtool [*vbdInterface*] debugfs ulprx_la

Example:

```
C:\Users\Administrator>cxgbtool vbd0 debugfs ulprx_la
      Pcmd        Type   Message                Data
0000000000000000     0  00000000  00000000000000000000000000000000
0000000000000000     0  00000000  00000000000000000000000000000000
0000000000000000     0  00000000  00000000000000000000000000000000
.
```

- **devlog**

Description: Prints firmware device log information.

Syntax: cxgbtool [*vbdInterface*] devlog

Example:

```
C:\Users\Administrator>cxgbtool vbd0 devlog
        Tstamp         Level        Facility     Message
Seq#
 0      318517         INFO         CORE         log initialized @ 0x20084000
                                                 size 32768 (128 entries)
                                                 fwrev 0x00000000pcie_fw
                                                 0x0014cc10
 1      388309         INFO         CORE         bootstrap firmware took 27
                                                 msecs to run
.
```

rss

Description: Prints RSS info.

Syntax: cxgbtool [*vbdInterface*] rss

Example:

```
C:\Users\Administrator>cxgbtool vbd0 rss
0:   1023     0     0     0     0     0     0     0
8:      0     0     0     0     0     0     0     0
16:  1023     0     0     0     0     0     0     0
24:     0     0     0     0     0     0     0     0

.
.
.
```

rss_config

Description: Prints RSS Configuration.

Syntax: cxgbtool [*vbdInterface*] rss_config

Example:

```
C:\Users\Administrator>cxgbtool vbd0 rss_config
TP_RSS_CONFIG: 0x300003c
    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

.
.
.
```

rss_key

Description: Prints RSS Key.

Syntax: cxgbtool [*vbdInterface*] rss_key

Example:

```
C:\Users\Administrator>cxgbtool vbd0 rss_key
c62ee59b4b9d292c2c8e195c582f8fd032568e579f03cdeb1b66ed910dab1d414c5516e6bdb1
9289
```

rss_pf_config

Description: Prints RSS PF Configuration

Syntax: cxgbtool [*vbdInterface*] rss_pf_config

Example:

```
C:\Users\Administrator>cxgbtool vbd0 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
1  yes  no  no    1    4    7    no  no    no  no    no    0    0
2  yes  no  no    2    4    7    no  no    no  no    no    0    0
3  yes  no  no    3    4    7    no  no    no  no    no    0    0
4  yes  no  no    4    4    7    no  no    no  no    no    0    0
5  yes  no  no    5    4    7    no  no    no  no    no    0    0
6  yes  no  no    6    4    7    no  no    no  no    no    0    0
7  yes yes  no    7    4    7   yes yes   yes yes    no    0    0
```

rss_vf_config

Description: Prints RSS VF Configuration.

Syntax: cxgbtool [*vbdInterface*] rss_vf_config

Example:

```
C:\Users\Administrator>cxgbtool vbd0 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
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
.
.
.
```

coalesce

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

Syntax: cxgbtool [*nicInterface*] 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.

Example:

Setting Rx Coalescing Timer:

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

Setting Rx Coalescing Threshold Packets

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

eps

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

Syntax: cxgbtool [*nicInterface*] eps

Example:

```
 C:\Users\Administrator>cxgbtool nic0 eps
 ep FFFFE00005734AE0 cm_id FFFFE000071C0250 state 1 flags 0x0 stid
1920    backlog 1024 fe80:0000:0000:0000:2c51:690a:9289:f 543:5445
 ep FFFFE00004822810 cm_id FFFFE000049261C0 state 1 flags 0x0 stid
1928   backlog 1024 169.254.245.67:5445
```

qps

Description: Prints queue paris (qps) if NDK/ND is enabled.

Syntax: cxgbtool [*nicInterface*] qps

Example:

```
 C:\Users\Administrator>cxgbtool nic0 qps
 rc qp sq id 1026 in_use 0 rq id 1180 in_use 176 state 1 onchip 0 ep tid
152    state 7 0.0.0.0:4341->0.0.0.0:5445
 rc qp sq id 1027 in_use 0 rq id 1393 in_use 256 state 1 onchip 0 ep tid
114   state 7 0.0.0.0:4322->0.0.0.0:5445
 rc qp sq id 1039 in_use 0 rq id 1823 in_use 256 state 1 onchip 0 ep tid
414   state 7 0.0.0.0:4884->0.0.0.0:5445
 rc qp sq id 1044 in_use 0 rq id 1374 in_use 256 state 1 onchip 0 ep tid
136  state 7 0.0.0.0:4333->0.0.0.0:5445
 rc qp sq id 1048 in_use 0 rq id 2169 in_use 256 state 1 onchip 0 ep tid
78  state 7 0.0.0.0:4304->0.0.0.0:5445
```

rdma_stats

Description: Prints RDMA statistics if NDK/ND is enabled

Syntax: cxgbtool [*vbdInterface*] rdma_stats

Example:

```
 C:\Users\Administrator>cxgbtool vbd0 rdma_stats

   Object:      Total    Current        Max       Fail
     PDID:      65536          0          0          0
      QID:      31744          0          0          0
   TPTMEM:   40265280          0          0          0
   PBLMEM:  273804160          0          0          0
   RQTMEM:   80530624          0          0          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:     589824
```

- **stags**

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

Syntax: cxgbtool [*nicInterface*] stags

Example:

```
C:\Users\Administrator>cxgbtool nic0 stags
Stag(0):0x600
0x83ae200 0000000000000000 0000000000000000 0000000000000000
0000000000000000
0x083ae200: (inactive): stag idx 0x0600 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 0x0b414780 len 0 va
0000000000000000                          bind_cnt 0
Stag(1):0x900
0x83ae260 0000000000000000 0000000000000000 0000000000000000
0000000000000000
0x083ae260: (inactive): stag idx 0x0900 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 0x0b414780 len 0 va
0000000000000000                          bind_cnt 0
Stag(2):0xf00
0x83ae320 0000000000000000 0000000000000000 0000000000000000
0000000000000000
```

```
0x083ae320: (inactive): stag idx 0x0f00 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 0x0b414780 len 0 va
0000000000000000                        bind_cnt 0
```

- ## chim_sock

Description: Prints chimney statistics.

Syntax: cxgbtool [*nicInterface*] chim [tcb {tid}|sock]

tcb: Prints Hardware TCB information for given tid of an offloaded connection.
sock: Prints driver per socket statistics.

Example:

Print Hardware TCB information

```
C:\Users\Administrator>cxgbtool nic0 chim tcb 9210
HW TCB
ulp_type                : 0x5
ulp_raw                 : 0x0
l2t_ix                  : 0x1
smac_sel                : 0x82
tf_suspend              : 0x0
tf_non_offload          : 0x0
tf_lock_tid             : 0x1
```

Print driver per socket statistics:

```
C:\Users\Administrator>cxgbtool nic0 chim sock
102.1.1.56:5001 ======> 102.1.1.62:49169
=========================================
ChRcvBytes              : 1460
ChRcvCredits            : 1460
ChRcvMaxSeq             : 0
ChRcvWndAdjusted        : 324776
Cached flag             : 0x2 :Nagle
Const flag              : 0x6 :SACK:WS
RcvIndicationSize       : 128
RcvNxt                  : 1871353069
.
.
.
```

l2t

Description: Displays l2t table contents.

Syntax: cxgbtool [*nicInterface*] l2t

Example:

```
C:\Users\Administrator>cxgbtool nic0 l2t
IDX     REF_COUNT       MTU     DEST_ADDR       DEST_MAC
```

hw_sched

Description: Displays hardware schedule information

Syntax: cxgbtool [*vbdInterface*] hw_sched

Example:

```
C:\Users\Administrator>cxgbtool vbd0 hw_sched
Scheduler  Mode   Channel  Rate (Kbps)   Class IPG(0.1 ns)   Flow IPG(us)
    0      class     0      29640261          disabled          disabled
    1      class     1      29640261          disabled          disabled
    2      class     2      29640261          disabled          disabled
    3      class     3      29640261          disabled          disabled
    4      class     0      29640261          disabled          disabled
    5      class     0      29640261          disabled          disabled
    6      class     0      29640261          disabled          disabled
    7      class     0      29640261          disabled          disabled
```

mbox_log

Description: Prints firmware mailbox command/reply log information.

Syntax: cxgbtool [*vbdInterface*] mbox_log

Example:

```
C:\Users\Administrator>cxgbtool vbd0 mbox_log
Seq#   Tstamp  Atime  Etime  Command/Reply


42     589890      0      0  08c00400 00000004  020b0000 00000000  020c0000
00000000  00000000 00000000  00000000
00000000  00000000 00000000  00000000
00000000  00000000 00000000
43     589890    0        1  08000400 00000004  020b0000 08d87840  020c0000
0918783f  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
44     589890    0        0  1bc00000 00030002  00000000 00000000  00000000
00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
45     589890    0        1  1b000000 00030002  001fc904 c1cac1ca  05ee0300
0000c0c0  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
46    589890    0         0  14b00400 80000004  00000000 00000000  00000000
00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
47    589890    0         1  14000400 80000004  04c00007 432df7c0  00000000
00000000  00400000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
48    589890    0         0  1bc00001 00030002  001fc904 c1cac1ca  05ee0300
0000c0c0  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
49    589890      0       1  1b000001 00030002  001fc904 c1cac1ca  05ee0300
0000c0c0  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
50   589890       0        0  14b00400 80000004  00000000 00000000  10000000
00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
51   589890       0        1  14000400 80000004  04c10007 432df7c8  10000000
00000000  00400000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
52   589890       0        0  14900400 40000004  04c00000 00000000  00000000
00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
53   589890       0        1  14000400 40000004  04c00000 00000000  00000000
00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
54   589890       0        0  14900400 40000004  04c10000 00000000  00000000
00000000  00000000 00000000  00000000 00000000  00000000 00000000  00000000
00000000  00000000 00000000
```

7. Firmware Update

The driver will auto-load the T5 and T4 firmwares if an update is required. The firmware version can be verified using *cxgbtool*:

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

The Firmware binaries are copied to *<system_drive>\ChelsioT4\Firmware* directory during Unified Wire installation.

8. Software/Driver Uninstallation

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

8.1.1. GUI mode (Installer)

1. Run the **ChelsioUwire-x.x.x.xx.exe** application.
2. 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**.

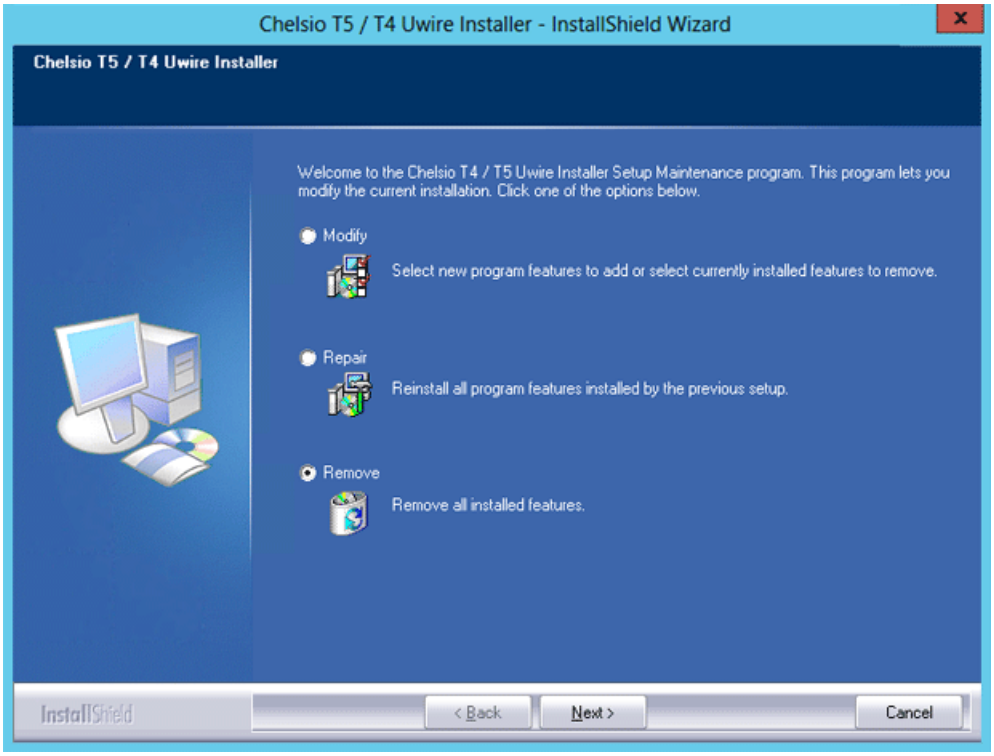

Figure 45 - Selecting maintenance option

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

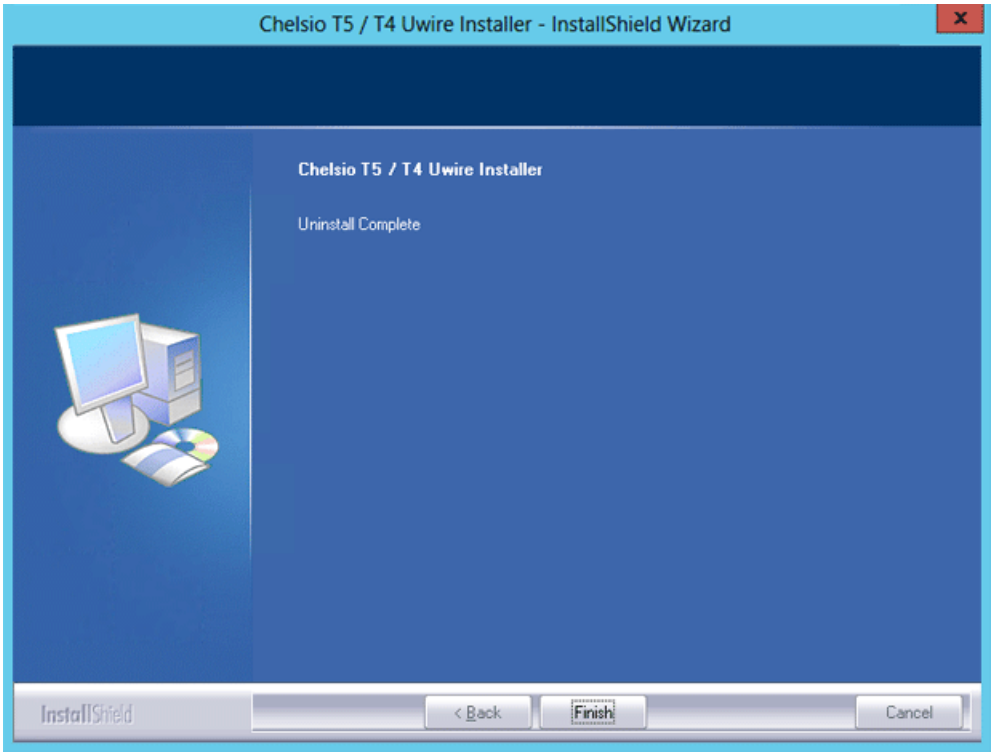

Figure 46 - Finishing uninstallation

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. You will have to manually uninstall it using "Programs and Features" in the Control Panel.*

8.1.2. CLI Mode

To uninstall all the drivers, execute the following command:

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

To uninstall a particular driver, execute the following command:

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

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. Software/Driver Update

For any distribution specific problems, please check README and Release Notes included in the release for possible workaround.

Please visit Chelsio support web site http://service.chelsio.com/ for regular updates on various software/drivers. You can also subscribe to our newsletter for the latest software updates.

II. NDIS Function

1. Introduction

Chelsio's T5/T4 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 LRO - Large Receive Offload).

1.1. Hardware Requirements

1.1.1. Supported Adapters

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

Chelsio Adapter	NDIS components
T520-BT	NIC, NVGRE, VXLAN, vRSS
T520-CR	NIC, NVGRE, VXLAN, PacketDirect, vRSS
T580-CR	NIC, NVGRE, VXLAN, PacketDirect, vRSS
T580-LP-CR	NIC, NVGRE, VXLAN, PacketDirect, vRSS
T580-SO-CR	NIC, NVGRE, VXLAN, vRSS
T520-LL-CR	NIC, NVGRE, VXLAN, vRSS
T520-SO-CR	NIC, NVGRE, VXLAN, vRSS
T540-CR	NIC, NVGRE, VXLAN, vRSS
T420-CR	NIC
T440-CR	NIC
T422-CR	NIC
T420-LL-CR	NIC
T440-LP-CR	NIC
T420-CX	NIC
T420-BT	NIC
T404-BT	NIC

1.2. Software Requirements

1.2.1. Windows Requirement

Currently Chelsio T5/T4 NDIS Function driver is available for the following Windows versions:

- Server 2016 Technical Preview
- 2016 Client
- Nano Server
- 10 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. 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 T5/T4 40G/10G Bus Enumerator**, and then click the **Advanced** tab.

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

2.1.2. NDIS Miniport Driver Parameters

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

- **Encapsulated Task Offload**
  - *Description*: Enable or disable the stateless offloads (checksums, LSO, VMQ and filtering) of encapsulated traffic.
  - *Value*: Disabled | Enabled
  - *Default*: Enabled

Note *This parameter is available only for T5 adapters.*

Encapsulation Overhead

- *Description*: Specify the amount of overhead required in Ethernet frames due to virtual network overlay encapsulation such as VXLAN and NVGRE.
- *Value*: 0-256
- *Default*: 0

Note *This parameter is available only for T5 adapters.*

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

Interrupt Moderation

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

Interrupt Moderation Rate

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

IPv4 Checksum Offload

- *Description*: Allow the checksum to be computed by the adapter for IPV4 packets.
- *Value*:
  - Disabled
  - Rx & Tx Enabled
  - Rx Enabled

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

Jumbo Packet (Maximum Transmission Unit)

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

Large Send Offload V2 (IPv4)

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

Large Send Offload V2 (IPv6)

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

Locally Administered Address

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

Maximum Number of RSS Processors

- *Description*: Change the number of RSS Processors.
- *Value*: 1 | 2 | 4 | 8
- *Default*: 8

Maximum Number of RSS Queues

- *Description*: Change the number of RSS Queues.
- *Value*: 1-8
- *Default*: 4

Maximum RSS Processor Number

- *Description*: Set the number of RSS processors to help the overall performance of the computer.
- *Value*: 0-63
- *Default*: 63

NDIS QoS

- *Description*: Enable or disable NDIS Quality of Service for DCB interface.
- *Value*: Disabled | Enabled
- *Default*: Disabled

NetworkDirect Functionality

- *Description*: Enable or disable Network Direct Functionality.
- *Value*: Disabled | Enabled
- *Default*: Enabled

NetworkDirect Interrupt Moderation

- *Description*: Control the interrupts generated during NDK traffic.
- *Value*: Disabled | Enabled
- *Default*: Enabled

For more information, refer http://msdn.microsoft.com/en-us/library/windows/hardware/ff556017%28v=vs.85%29.aspx

Nvgre Task Offload

- *Description*: Enable or disable offloading processing of NVGRE encapsulated frames.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Packet Direct

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

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 40/10Gb 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 40/10Gb 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*: 0-16
- *Default*: 0

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

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

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

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

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

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

TCP Checksum Offload (IPv4):

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

TCP Checksum Offload (IPv6):

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

Tx Ethernet Queue Size

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

Tx Offload Queue Size

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

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

UDP Checksum Offload (IPv4)

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

UDP Checksum Offload (IPv6)

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

User Mode NetworkDirect

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

Virtual Machine Queues

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

Virtual Switch RSS

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

VLAN Identifier

- *Description*: Specify the VLAN ID to be inserted.
- *Value*: 0-4095
- *Default*: 0

VMQ LookAhead Split

- *Description*: Enable or disable the ability to split receive buffers into lookahead and post-lookahead buffers.
- *Value*: Disabled | Enabled
- *Default*: Enabled

VMQ VLAN ID Filtering

- *Description*: Enable or disable the ability to filter network packets by using the VLAN identifier in the media access control (MAC) header.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Vxlan Task Offload

- *Description*: Enable or disable offloading processing of VXLAN encapsulated frames.
- *Value*: Disabled | Enabled
- *Default*: Enabled

Vxlan UDP Port Number

- *Description*: Specify the current VXLAN UDP destination port number that is currently operational in the NIC.
- *Value*: 1-65535
- *Default*: 4789

Note *This parameter is available only for T5 adapters.*

2.2. NVGRE Offload

Chelsio's T5 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 T5 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 T5 based adapters offer extensive support for this feature leading to a higher throughput performance. This feature is enabled by default.

2.1. Virtual RSS (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 RSS or vRSS solves this bottleneck by distributing the network traffic across multiple virtual processors. This feature is disabled by default and can be enabled using the Device Manager.

III. 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, Terminator 5 based 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 driver:

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

1.2. Software Requirements

1.2.1. Windows Requirement

Currently Chelsio T5/T4 SMB Direct driver is available for the following Windows versions:

- Server 2016 Technical Preview
- 2016 Client
- Nano Server
- Server 2012 R2

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

2. Software/Driver Configuration and Fine-tuning

2.1. Enabling SMB Direct

SMB Direct is installed and enabled by default. RDMA functionality on Chelsio adapters will be enabled after installing the driver from the package.

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:

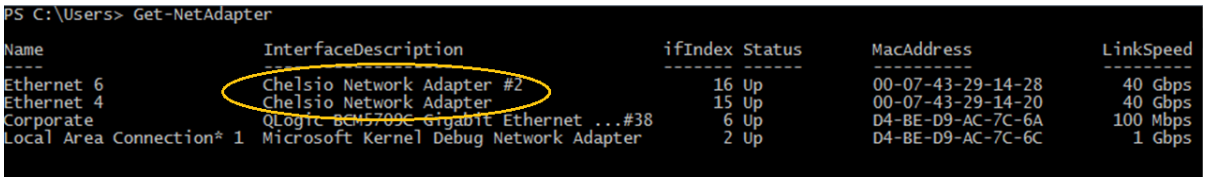

Figure 47 - Verifying Chelsio driver

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

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

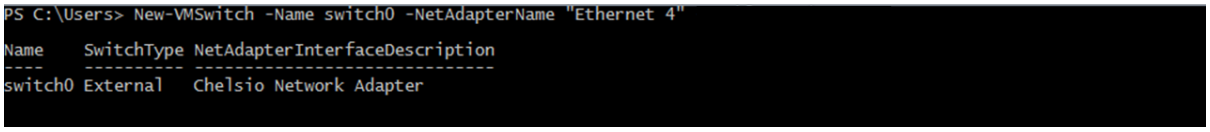

Figure 48 - Creating virtual switch

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.

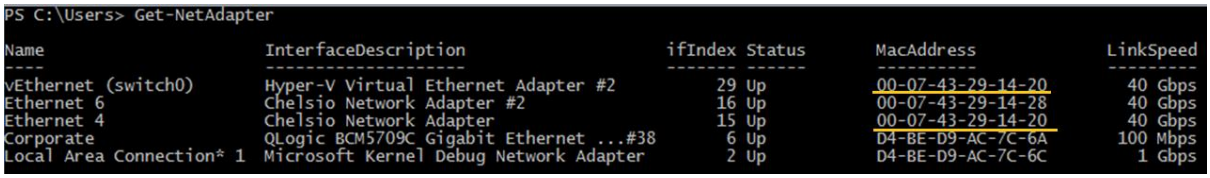

Figure 49 - Verifying virtual switch

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

Figure 50 - Enabling RDMA on virtual switch

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

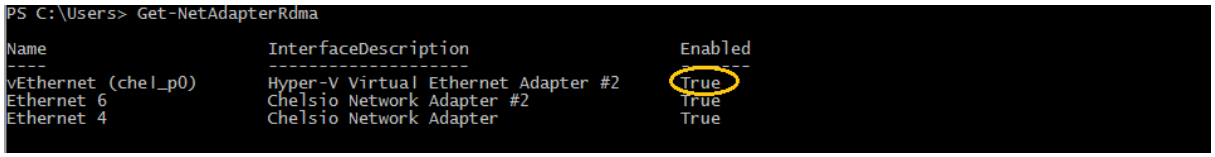

Figure 51 - Verifying RDMA on virtual switch

2.4. Troubleshooting

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

- Operating system version should be Windows Server Technical Preview (and Client) or Server 2012 R2.
- 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.

IV. 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 T5 and T4 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 driver:

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

1.2. Software Requirements

1.2.1. Windows Requirements

Currently, the NDIS SR-IOV driver is available for the following Windows versions:

- Server 2016 Technical Preview
- Nano Server
- 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

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. The script will be copied to *<system_drive>\Windows\system32* folder during Unified Wire installation. Open PowerShell with administrative privileges and follow the steps mentioned below:

i. Check if the machine is SR-IOV capable using the following command:

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

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

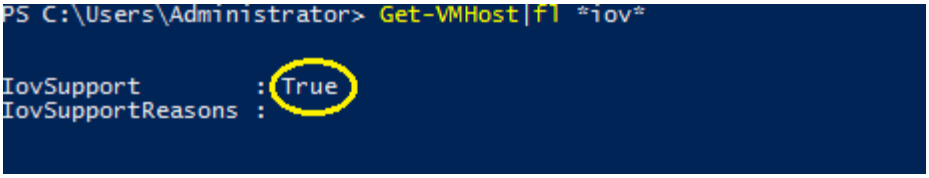

Figure 52 - Checking SR-IOV capability

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

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

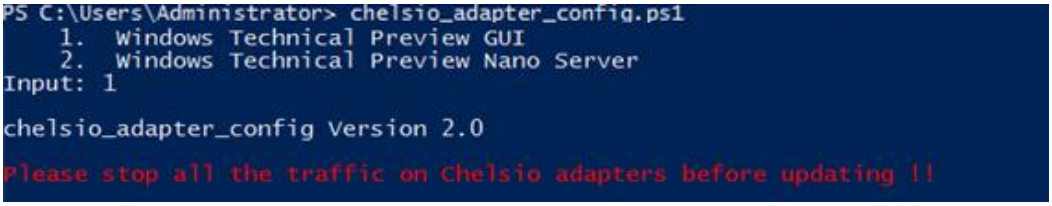

Figure 53 - Adapter Configuration Utility

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

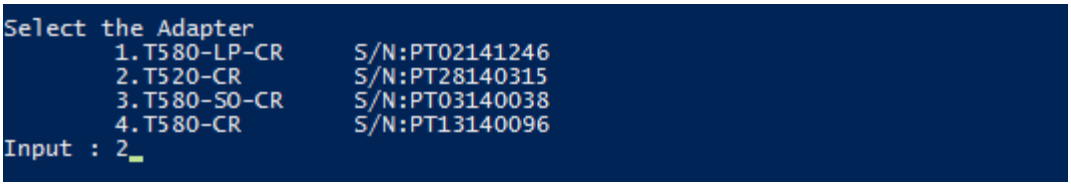

Figure 54 - Selecting adapter

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

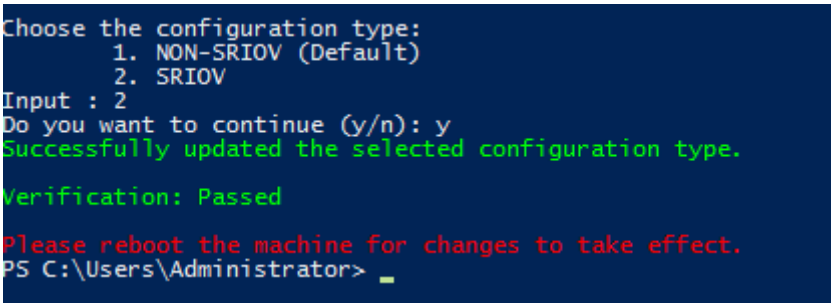

Figure 55 - Setting the configuration type

v. Reboot system for changes to take effect.

2.2. vSwitch Configuration

i. Create a vSwitch with SR-IOV enabled using Chelsio adapter.
ii. Assign a virtual network adapter with SR-IOV enabled to the VM.
iii. Bring up the VM.

2.3. Guest (VM) Configuration

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

Note *Custom installation not supported on VM.*

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

V. 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:

- 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 Technical Preview
- 2016 Client
- 10 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 T5 iSCSI Initiator** and click on **Properties**.

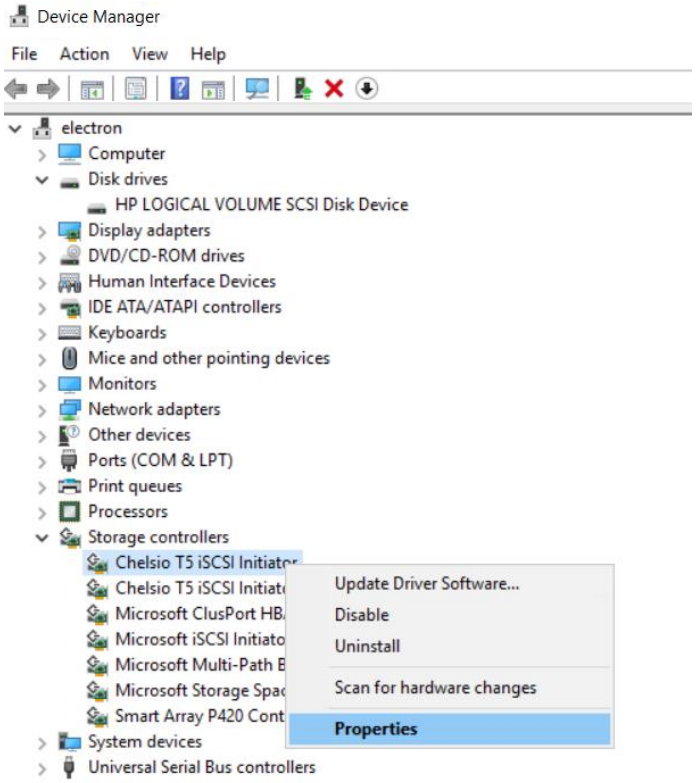

Figure 56 - Device Manager

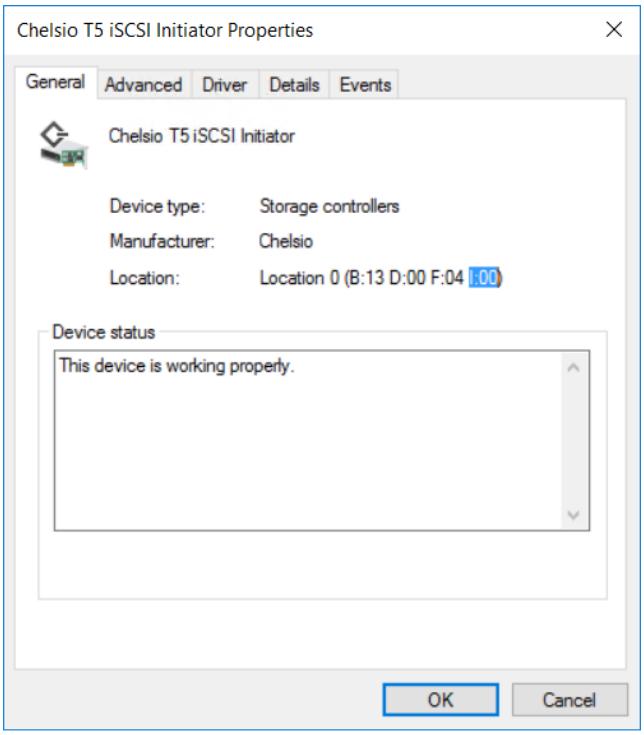

Figure 57 - Chelsio T5 physical port properties

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

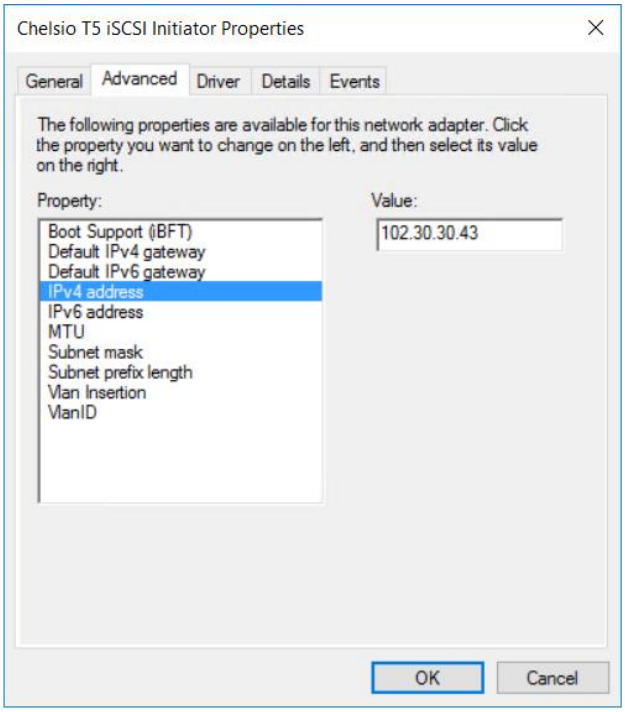

Figure 58 - Assigning IPv4 address

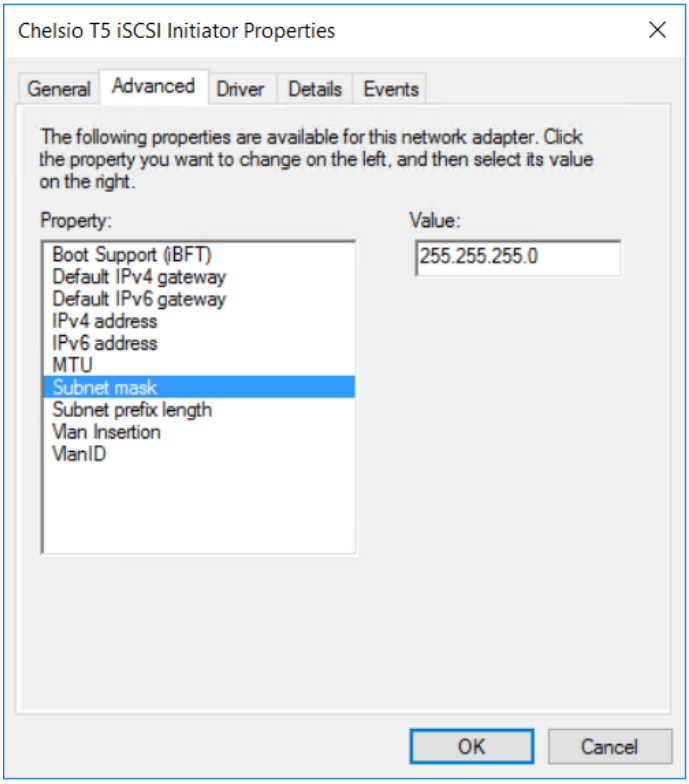

Figure 59 - Adding Subnet mask

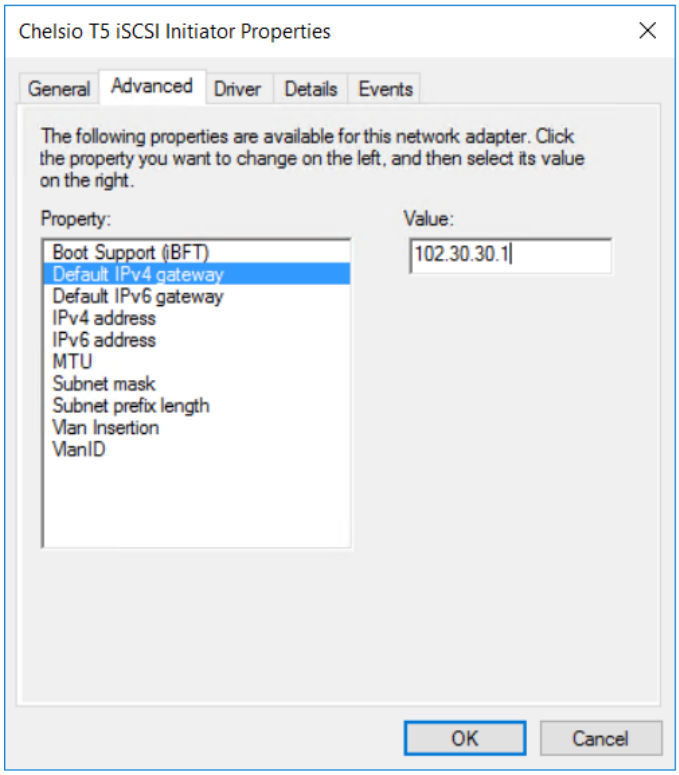

Figure 60 - Adding Default IPv4 gateway

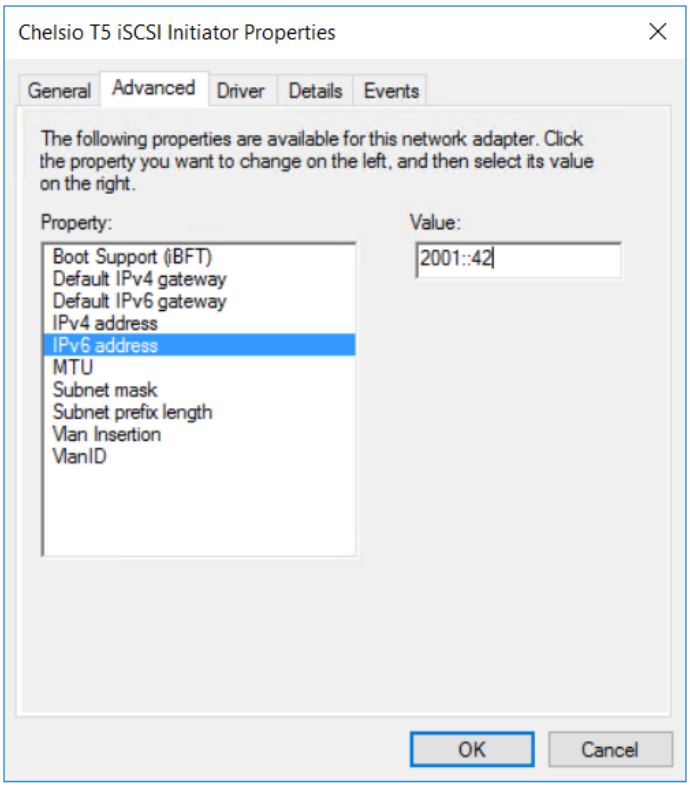

Figure 61 - Assigning IPv6 address

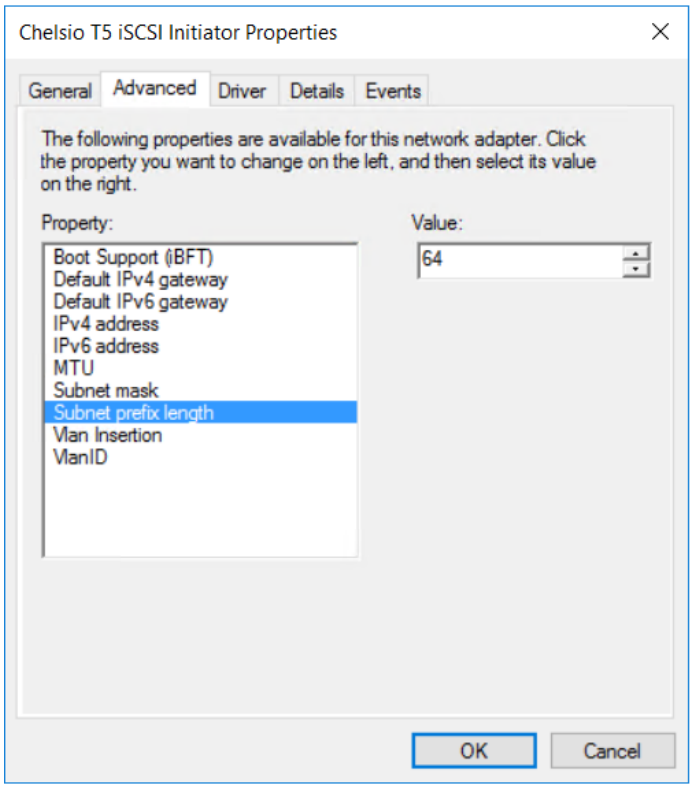

Figure 62 - Adding Subnet prefix length

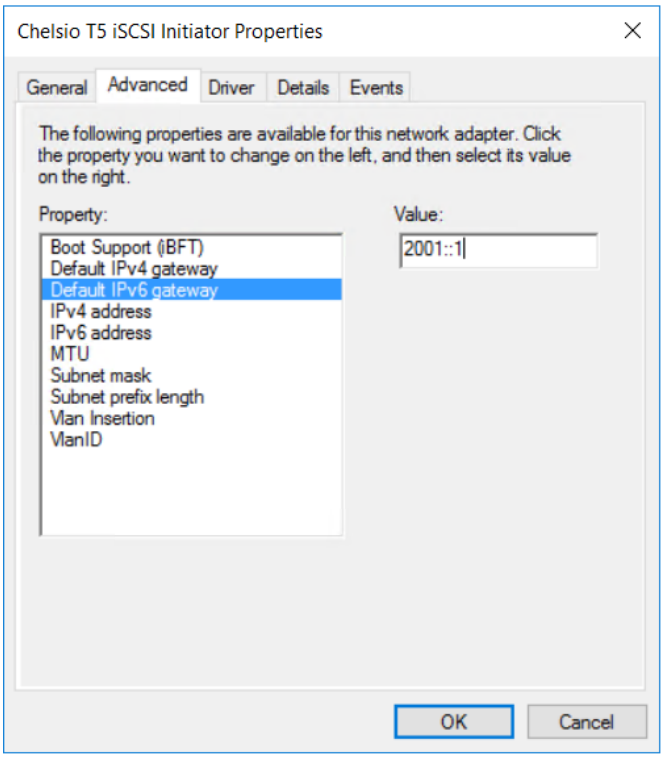

Figure 63 - Adding Default IPv6 gateway

3. Select MTU and assign value as per requirement.

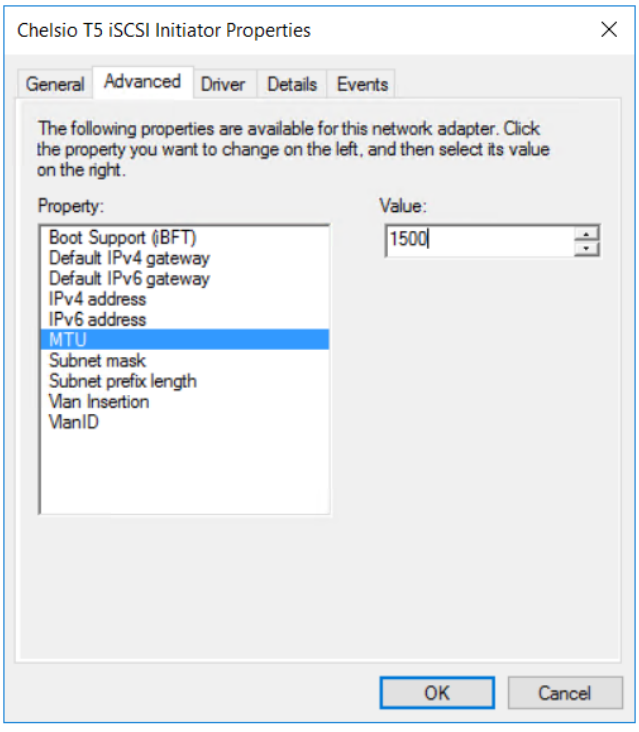

Figure 64 - Setting MTU

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

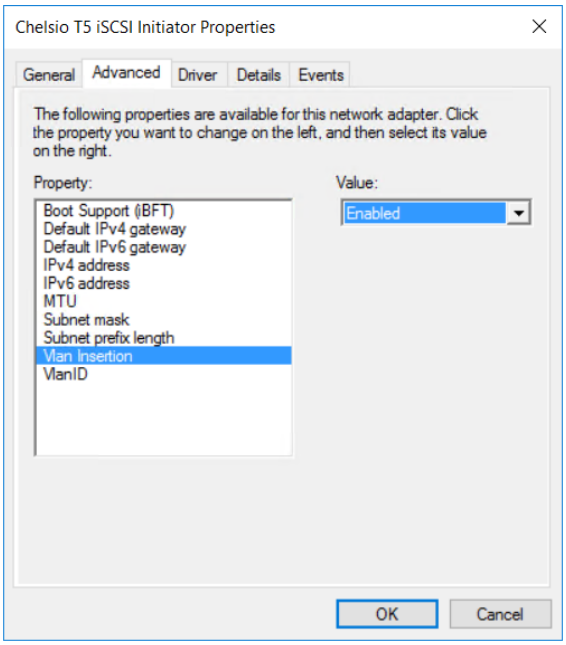

Figure 65 - Enabling VLAN

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

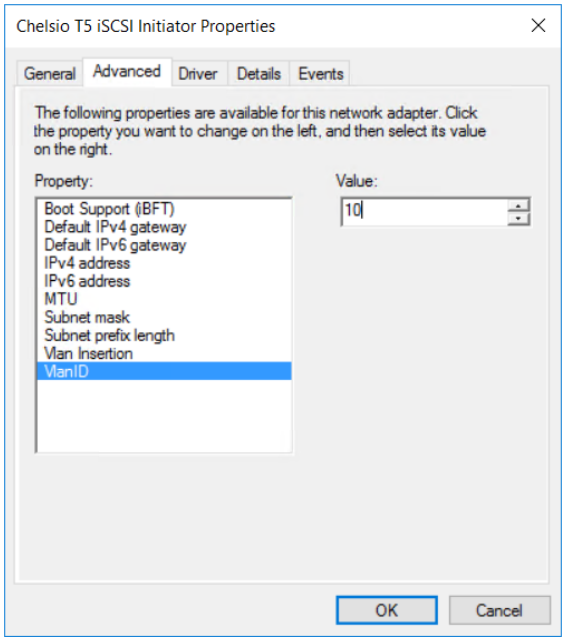

Figure 66 - Assigning VLAN Id

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

2.2. iSCSI Target Discovery and Login

Before target discovery, make sure a static IP address is assigned in the Device Manager to the respective Chelsio T5 iSCSI node.

Following are the set of instructions to discover the target and login to it using the Chelsio T5 iSCSI interface:

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

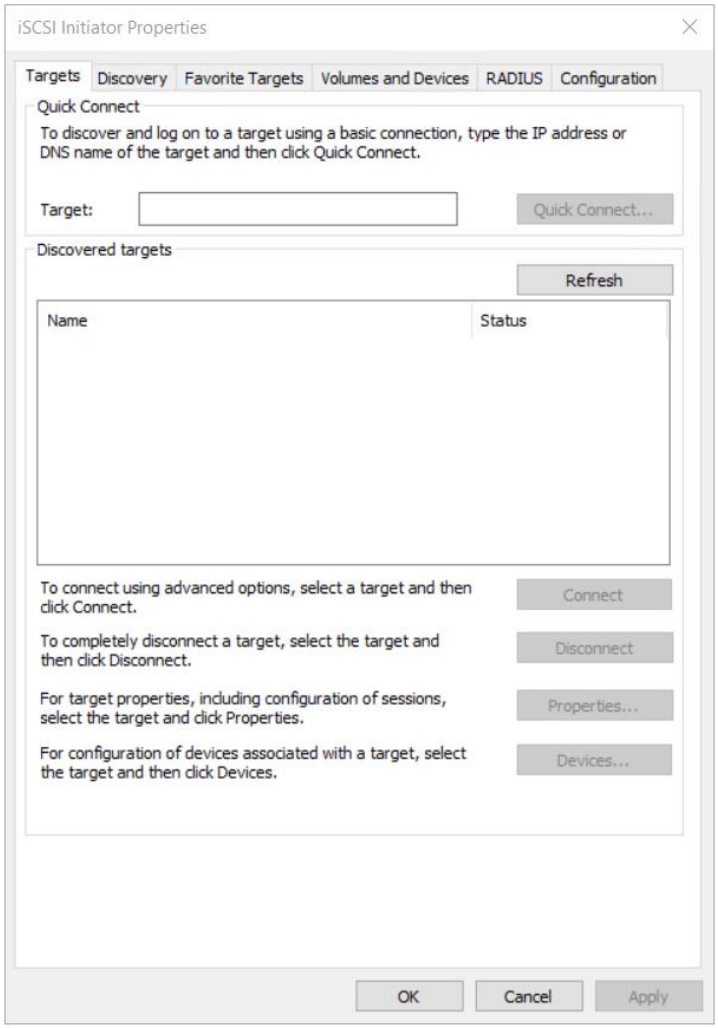

Figure 67 - iSCSI Initiator Properties: Targets tab

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

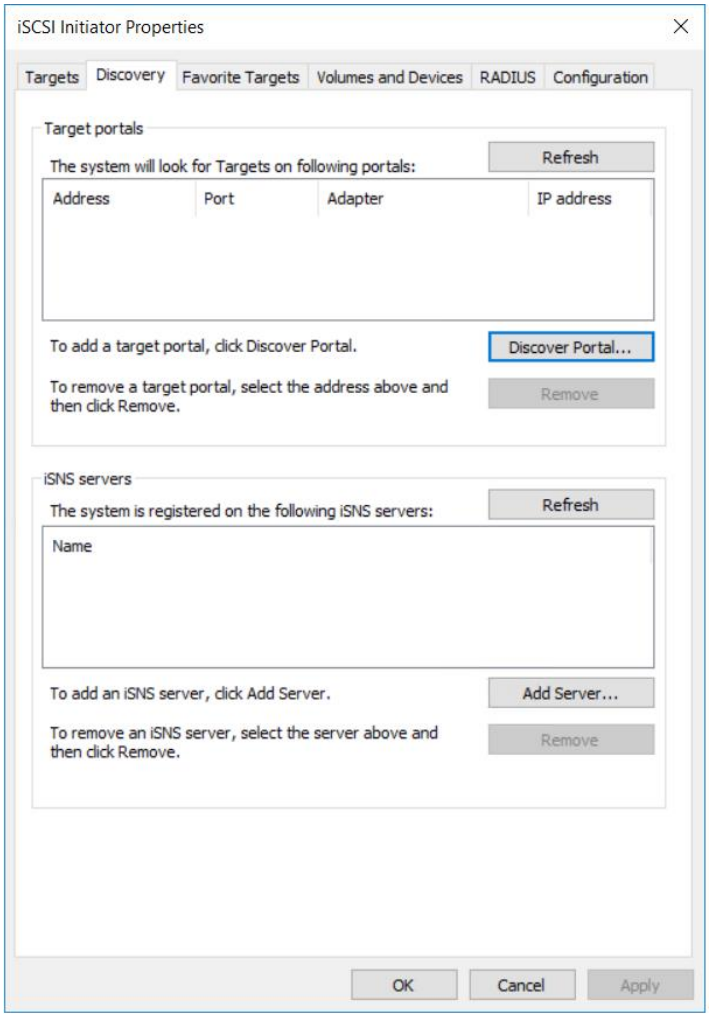

Figure 68 - iSCSI Initiator Properties: Discovery tab

3. Click on the **Advanced** button.

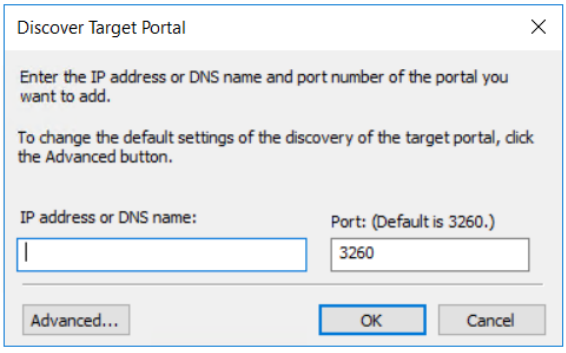

Figure 69 - Discovery Target Portal window

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

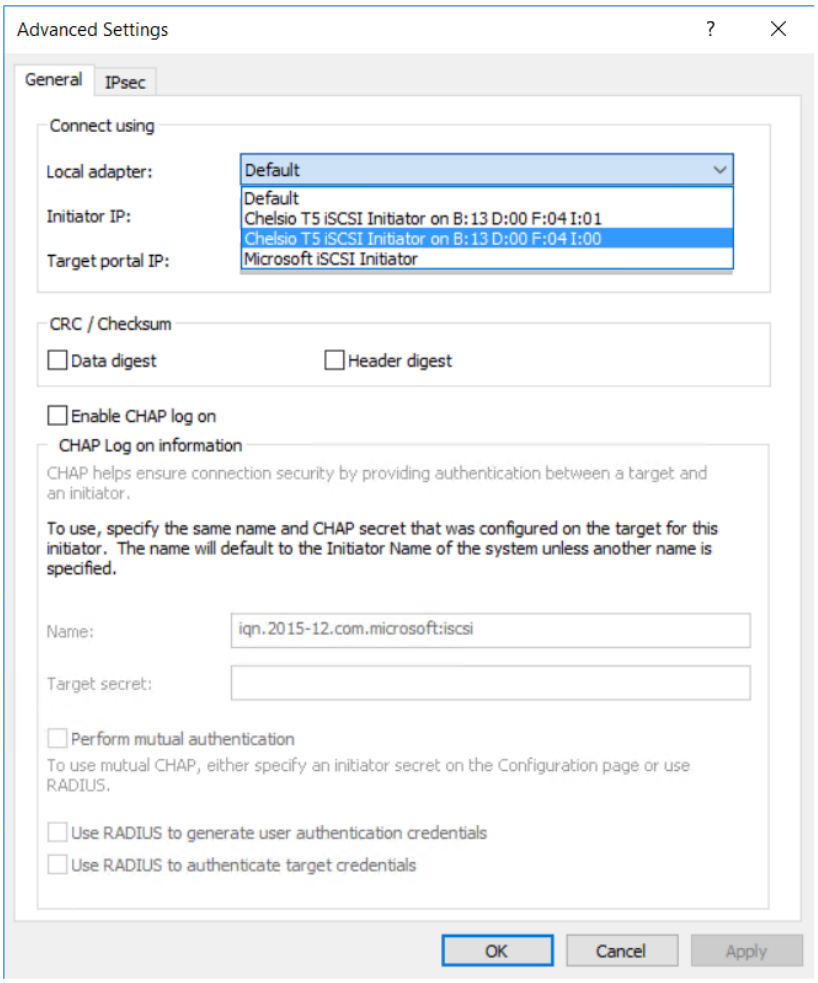

Figure 70 - Discovery Target Portal Advanced Settings: Selecting Local adapter

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

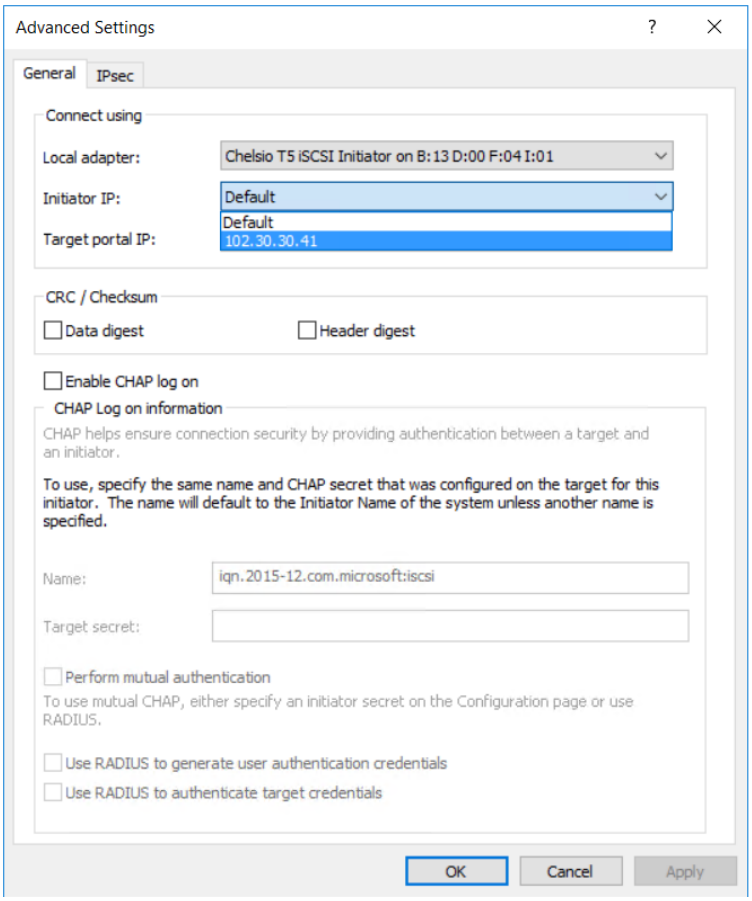

Figure 71 - Discovery Target Portal Advanced Settings: Specifying Target portal IP

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

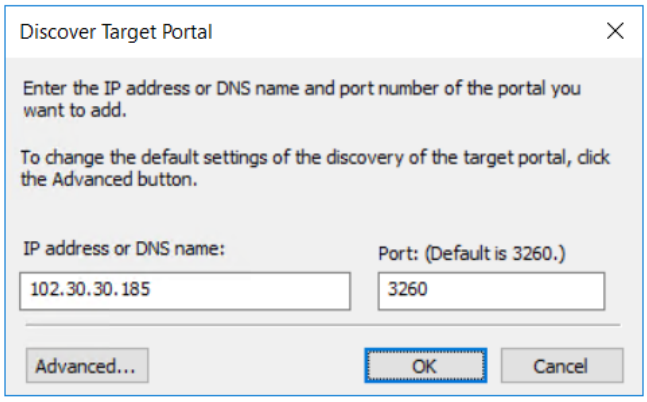

Figure 72 - Adding Target portal

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

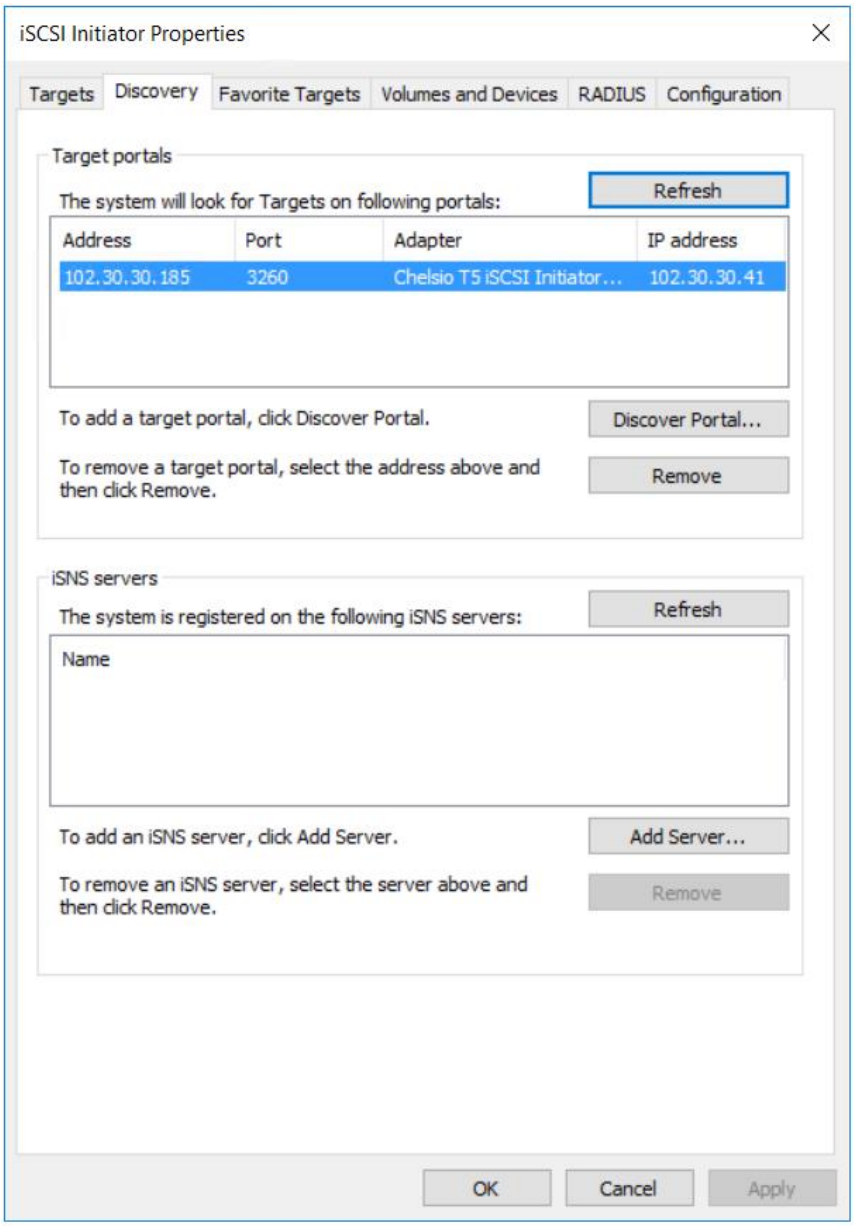

Figure 73 - Target portal added

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

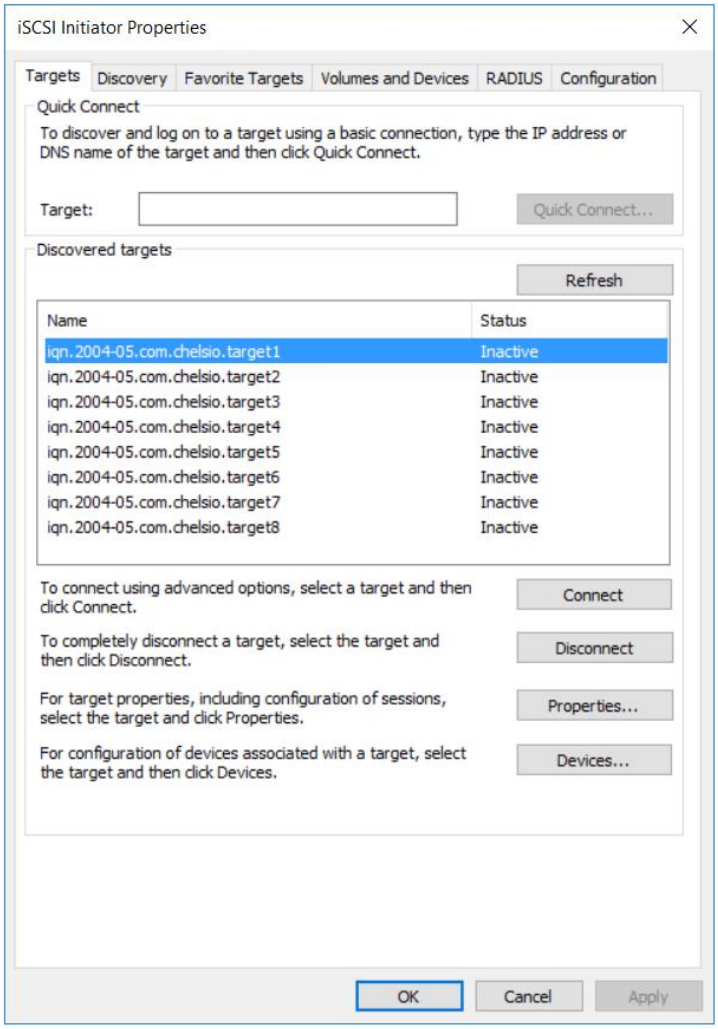

Figure 74 - Targets tab displaying list of available targets

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

Figure 75 - Connecting to Target

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

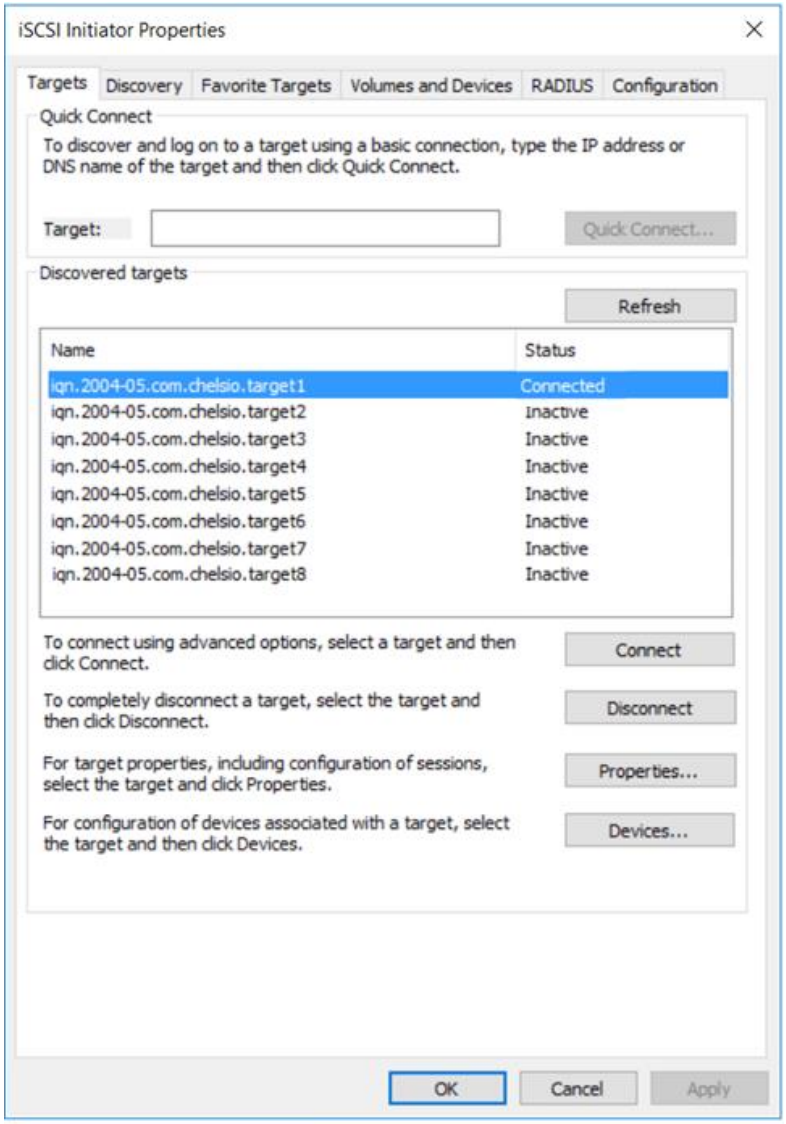

Figure 76 - iSCSI target connected

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

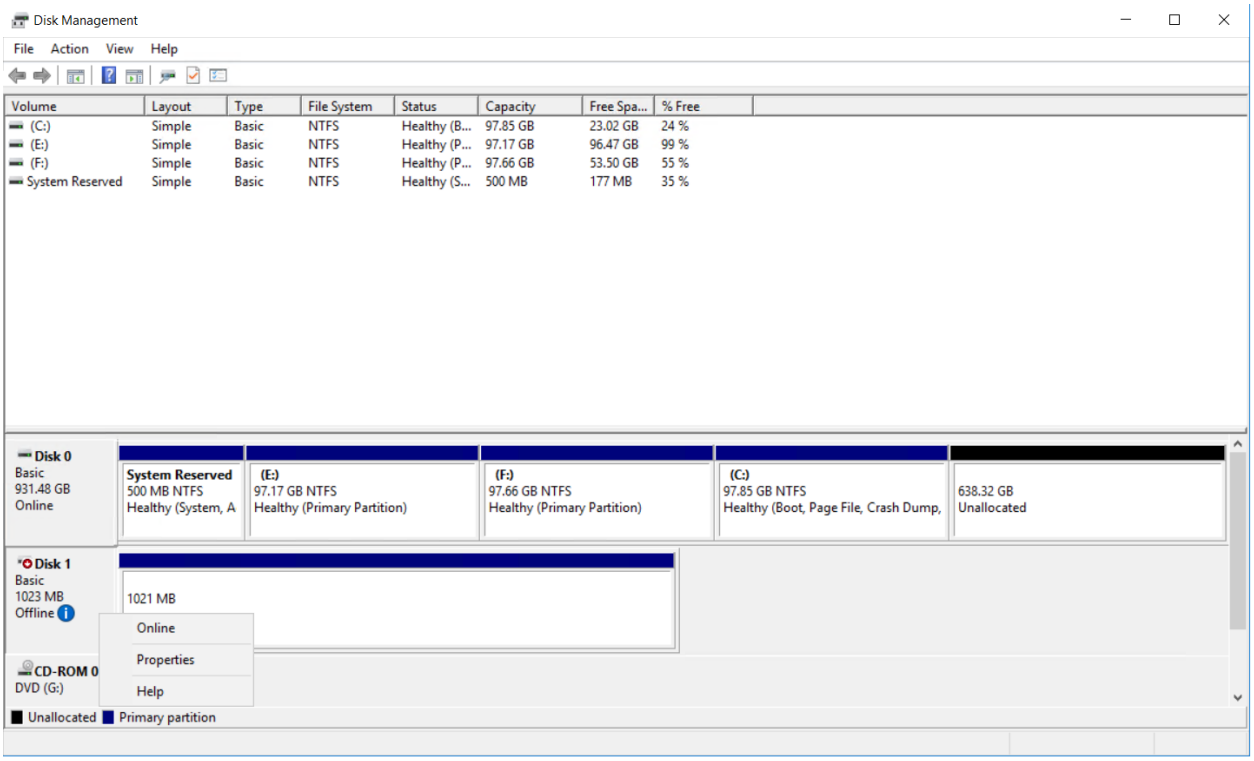

Figure 77 - Disk Management: making volume online

VI. 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:

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

1.2. Software Requirements

1.2.1. Windows Requirements

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

- Server 2016 Technical Preview
- 2016 Client
- Nano Server
- Server 2012 R2

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

2. Software/Driver Configuration and Fine-tuning

2.1. Installing DCB

To install DCB on the host machine, open PowerShell with administrative privileges and run the following command:

```
PS C:\Users\Administrator> Install-WindowsFeature -Name "Data-Center-
Bridging" -IncludeAllSubFeature -IncludeManagementTools
```

Although not mandatory, it is recommended to reboot the machine after installation.

2.2. Quality of Service (QoS)

Quality of Service (QoS) is an advanced Windows feature used to prioritize and manage the outgoing network traffic. The feature ensures efficient usage of resources and minimizes the impact of bandwidth congestion.

2.2.1. Enabling QoS

QoS can be enabled by using either Device Manager or PowerShell:

- **Using Device Manager**

i. Open the **Device Manager** (Start > Control Panel -> System & Security-> System -> Device Manager), click on the **Network adapters**, double click on any **Chelsio Network Adapter**, and then click the **Advanced** tab.
ii. Select the **NDIS QoS** property and change the value to *Enabled*.

- **Using PowerShell**

Open PowerShell with administrative privileges and run the following command:

```
PS C:\Users\Administrator> Get-NetAdapterRdma | where
{$_.InterfaceDescription -like "Chelsio*"} | Enable-NetAdapterQos'
```

2.2.2. Verifying QoS

To verify if QoS is enabled, use the PowerShell command *Get-NetAdapterQos* and observe the results. You should see a similar output:

```
PS C:\Users\Administrator> Get-NetAdapterQos
Name          : Chelsio_p1
Enabled       : True
Capabilities :                        Hardware     Current
                                      --------     -------
               MacSecBypass         : NotSupported NotSupported
               DcbxSupport          : IEEE         IEEE
               NumTCs(Max/ETS/PFC)  : 8/8/8        8/8/8
```

2.3. Configuring Dell Force10 Switch

2.3.1. Enabling DCB

To enable DCB functions on a Dell Force10 switch, follow the steps mentioned below:

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
```

2.3.2. Verifying

To verify remote and operational QoS configurational parameters, run the *Get-NetAdapterQos* command on the host machine. You should see a similar output:

```
PS C:\Users\Administrator> Get-NetAdapterQos
Name                        : Chelsio_p1
Enabled                     : True
Capabilities                :                          Hardware     Current
                                                       --------     -------
                              MacSecBypass          : NotSupported NotSupported
                              DcbxSupport           : IEEE         IEEE
                              NumTCs(Max/ETS/PFC)   : 8/8/8        8/8/8

OperationalTrafficClasses   : TC TSA    Bandwidth Priorities
                              -- ---    --------- ----------
                               0 ETS    80%       5-7
                               1 ETS    20%       0-4

OperationalFlowControl      : Priorities 5-7 Enabled
OperationalClassifications  : Not Available
RemoteTrafficClasses        : TC TSA    Bandwidth Priorities
                              -- ---    --------- ----------
                               0 ETS    80%       5-7
                               1 ETS    20%       0-4

RemoteFlowControl           : Priorities 5-7 Enabled
RemoteClassifications       : Not Available
```

VII. 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.

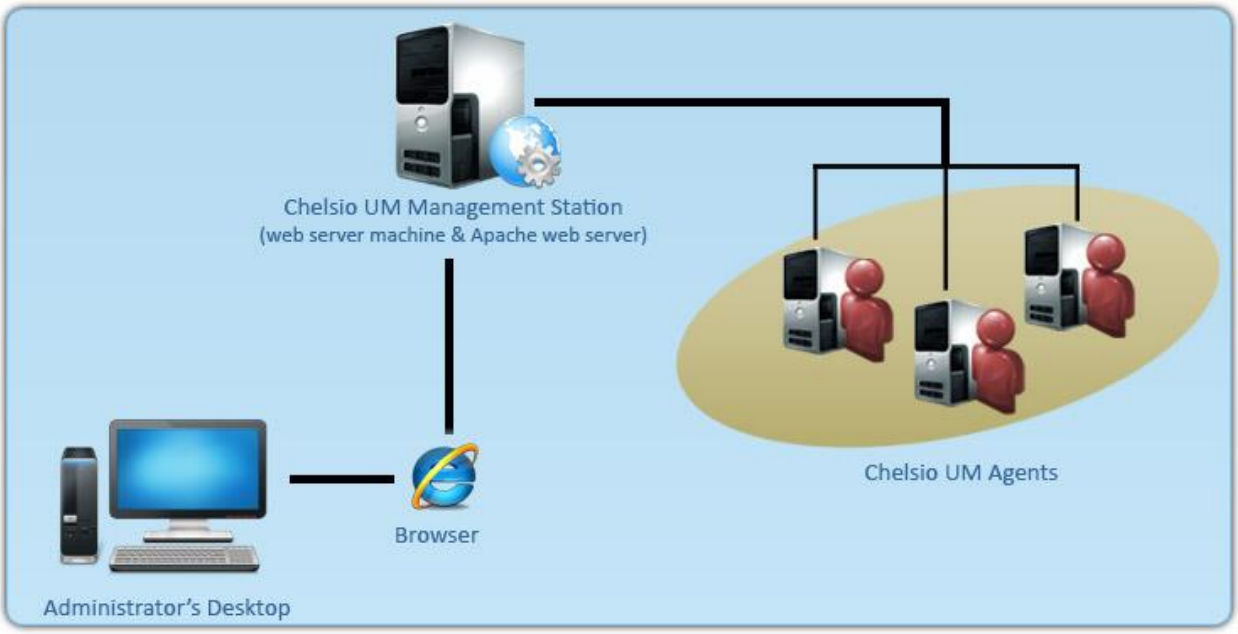

Figure 78 - Chelsio Unified Wire Manager with web interface (Web GUI)

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:

- **T5 Adapters**
  - T520-CR
  - T580-CR
  - T580-LP-CR
  - T580-SO-CR
  - T520-LL-CR
  - T520-SO-CR
  - T540-CR

- **T4 Adapters**
  - T420-CR
  - T440-CR
  - T422-CR
  - T404-BT
  - T420-BT
  - T420-LL-CR
  - T440-LP-CR
  - T420-CX

2.2. Platform/Component Matrix

The table below lists the Windows versions and the 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 T5/T4 driver(s) and their supported versions:

Chelsio driver	Version
NIC	6.1.17.0

3. Installing Unified Wire Manager

3.1. Pre-requisites

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

3.1.1. 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

3.1.2. Management Station

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

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

3.2. Installation

The following section describes the procedure to install UM. If an older version of the software exists, the Installer will upgrade it to the current version.

1. Run the **UnifiedWireManager-x.x.xx-x64** installer application.
2. Click the **Next** button for the Chelsio End User License Agreement Window.

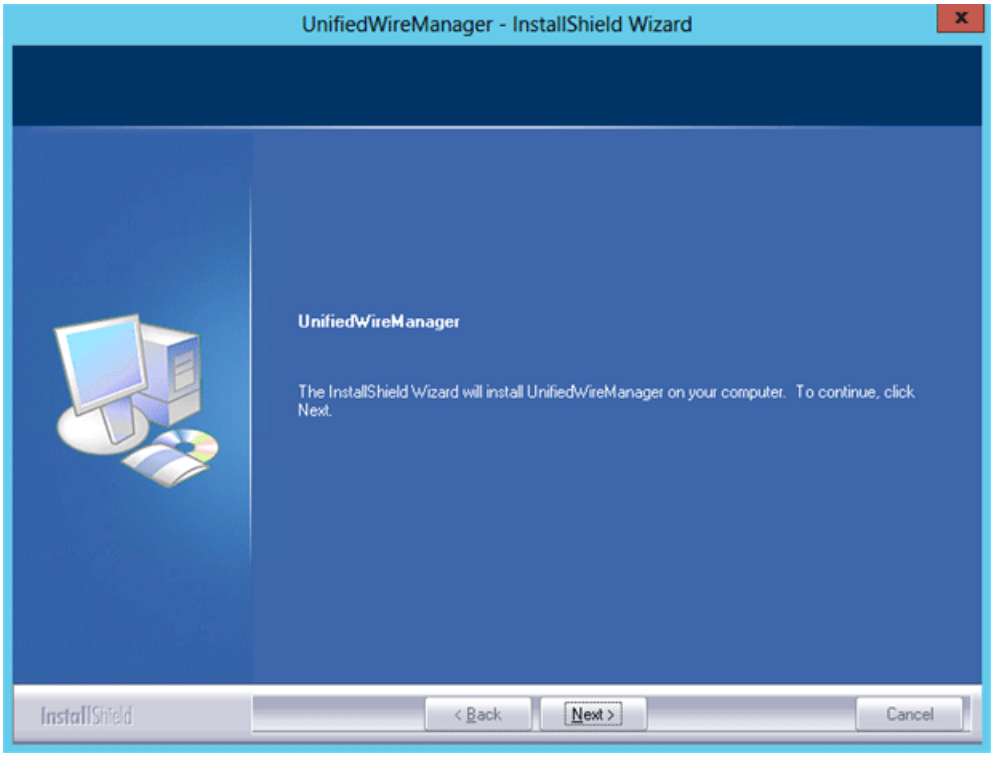

Figure 79 - UM installer welcome window

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

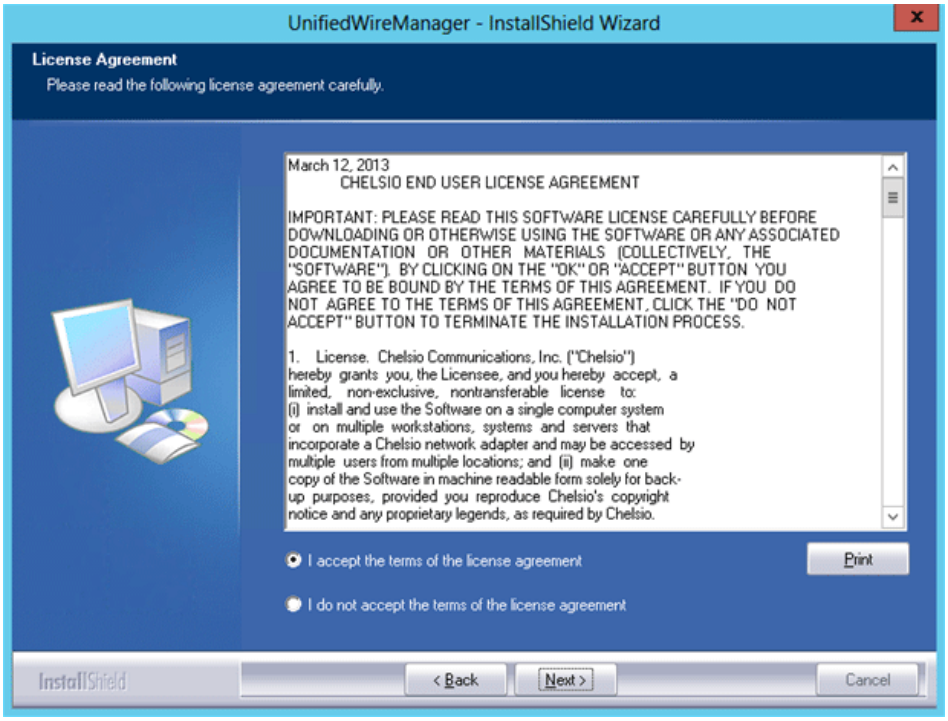

Figure 80 - Chelsio EULA window

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

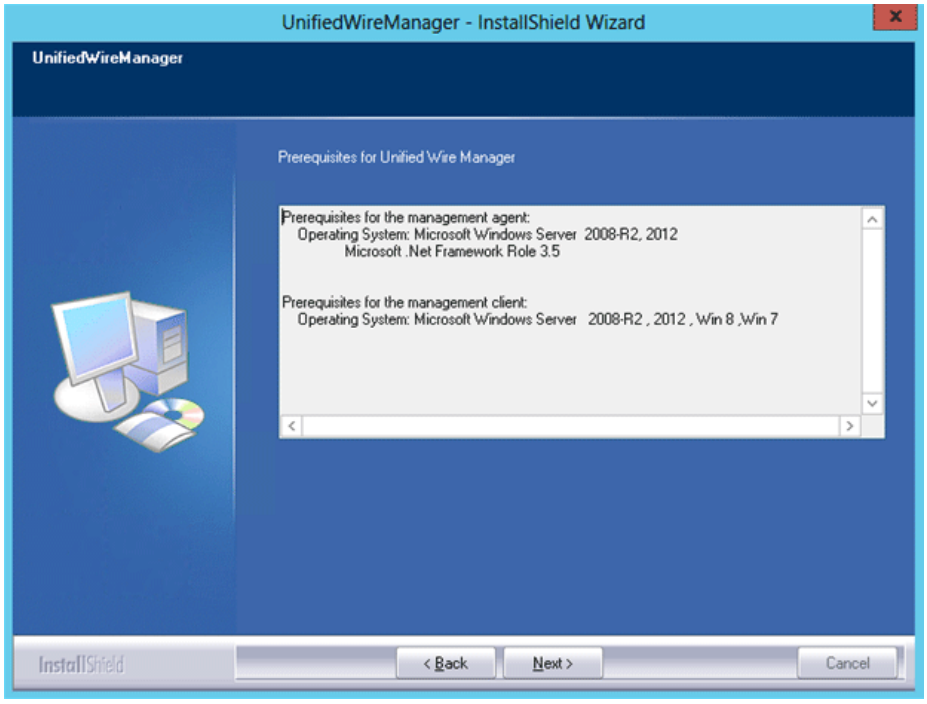

Figure 81 - UM prerequisites window

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

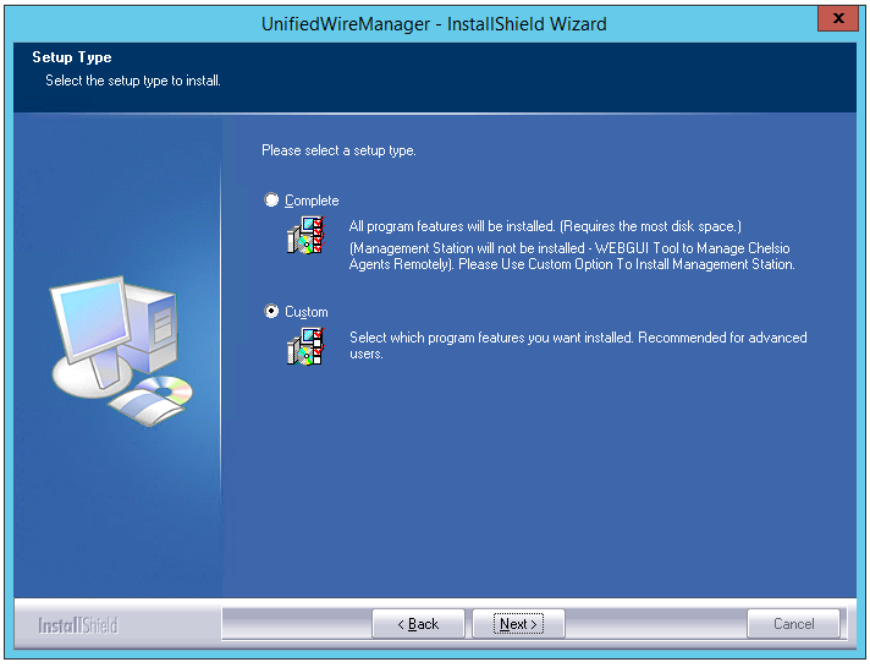

Figure 82 - Select setup (installation) type

iii. 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.

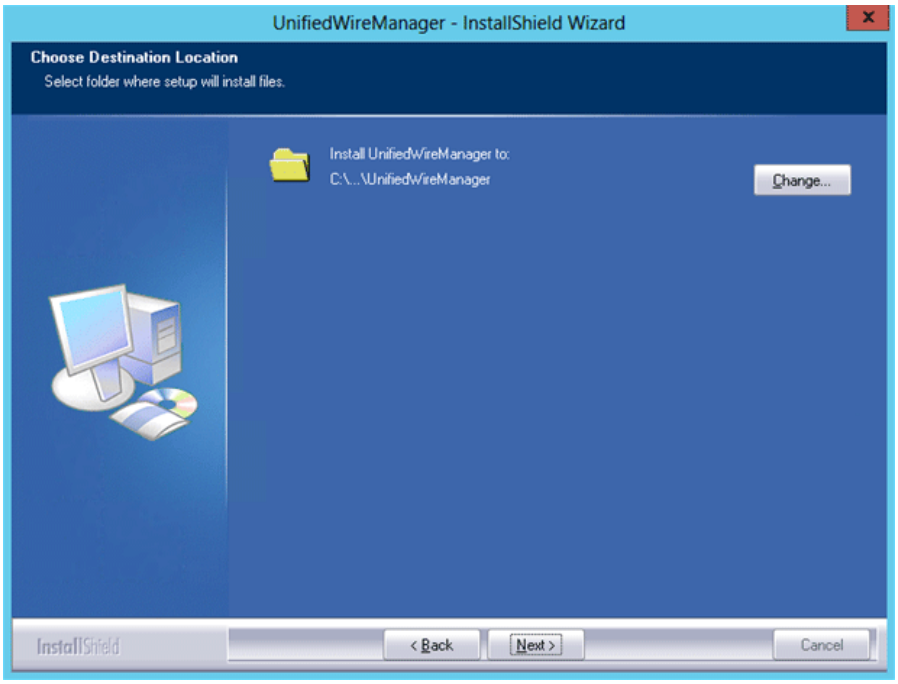

Figure 83 - Changing UM installation path

iv. Next, you can choose the UM components you wish to install. Deselect the components you don't wish to install and click **Next.**

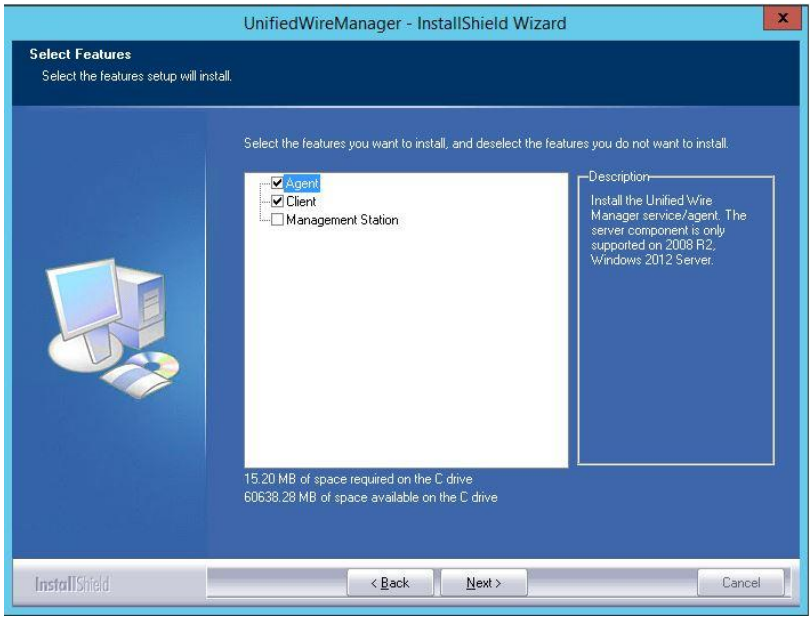

Figure 84 - Selecting UM components to install

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.*

6. Click **Install** to start the installation. Unified Wire Manager will now be installed with the selected options.

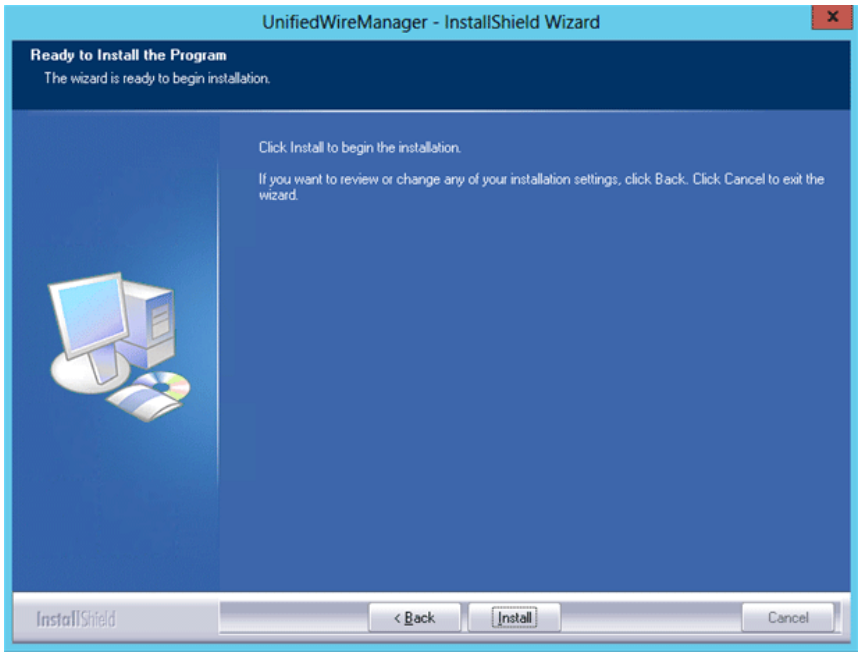

Figure 85 - Starting UM installation

7. Select **Finish** to exit the UM Installer.

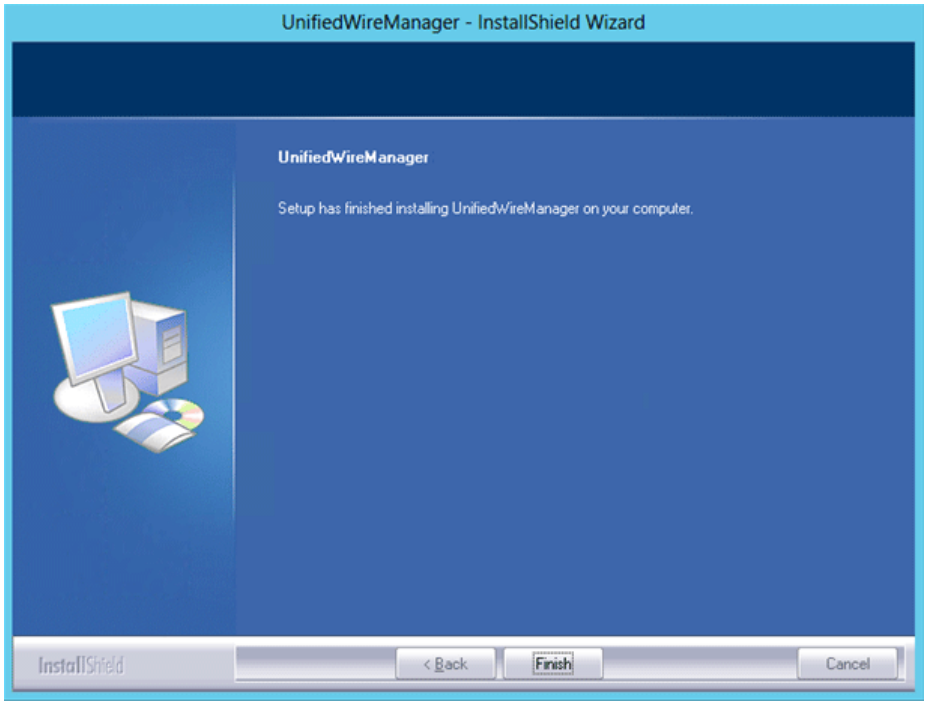

Figure 86 - Finishing UM installation

4. Configuring Unified Wire Manager

4.1. Configuring Management Station

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

4.1.1. Apache HTTP Server

1. 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
```

2. Perform the following tasks in *C:\Program Files (x86)\Apache Software Foundation\Apache2.2\Conf\httpd.conf* file
   i. Change "DocumentRoot" to “C:/Program Files (x86)/Apache Software Foundation/Apache2.2/htdocs/chelsio”
   ii. Uncomment "LoadModule rewrite_module modules/mod_rewrite.so"
   iii. Uncomment "Include conf/extra/httpd-ssl.conf"
   iv. Uncomment "LoadModule ssl_module modules/mod_ssl.so"
   v. Add the following lines:
   RewriteEngine On
   RewriteCond %{HTTPS} !=on
   RewriteRule ^/?(.) https://%{SERVER_NAME}/$1 [R,L]*

4.1.2. SSL Certificate

1. 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
```

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

3. Open the file *httpd-ssl.conf* and make the following modifications:

   i. Modify "SSLCertificateFile" to “C:\Program Files (x86)\Apache Software Foundation\Apache2.2\conf\blarg.crt”
   ii. 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*

i. Uncomment the following line
SSLSessionCache "dbm:C:/Program Files (x86)/Apache Software Foundation/Apache2.2/logs/ssl_scache"

ii. 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

1. On the system running Management Agent, launch **Services** from the Control Panel. You can also type *services.msc* in the **Run** command**.** This will open the **Services** window.

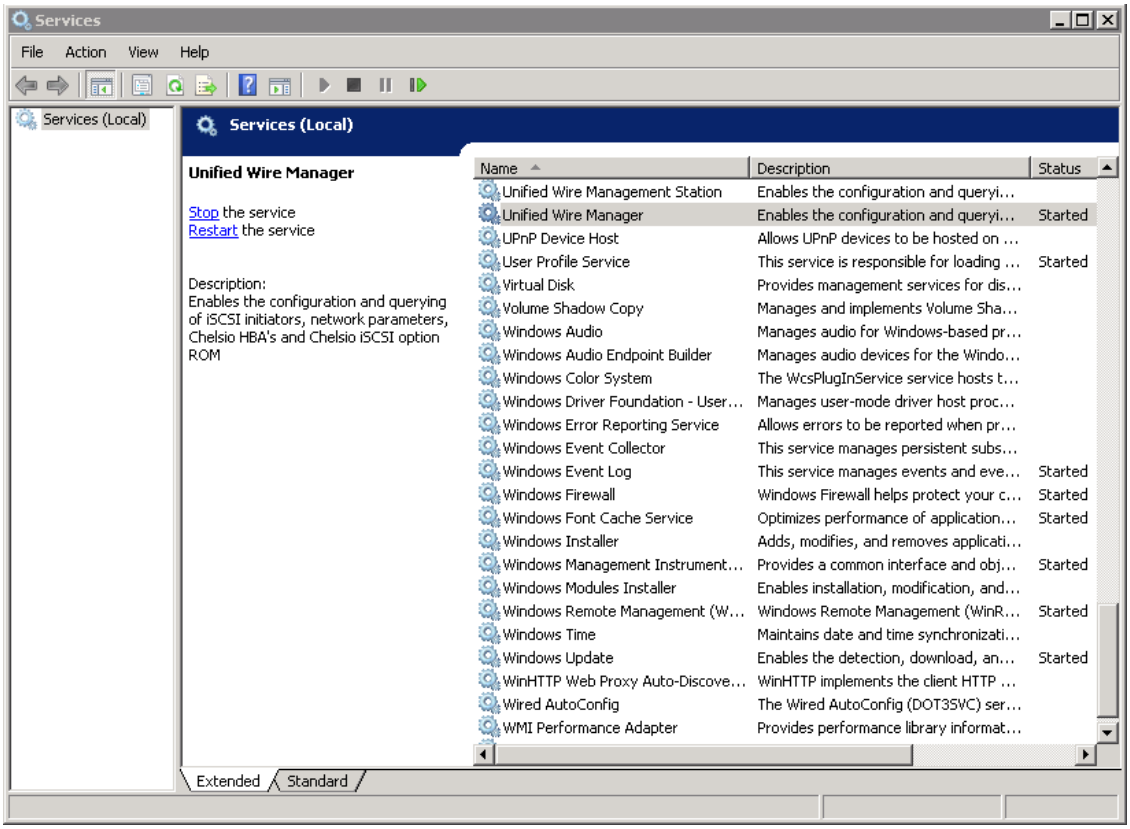

Figure 87 - Windows Services

2. 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.

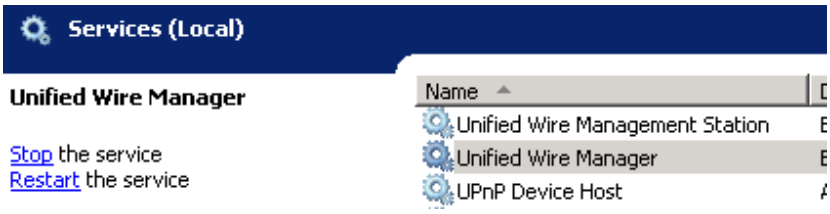

Figure 88 - Actions available for Unified Wire Manager service

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

1. Launch **Services** from the Control Panel. You can also type s*ervices.msc* in the **Run** command**.** This will open the **Services** window.
2. 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.

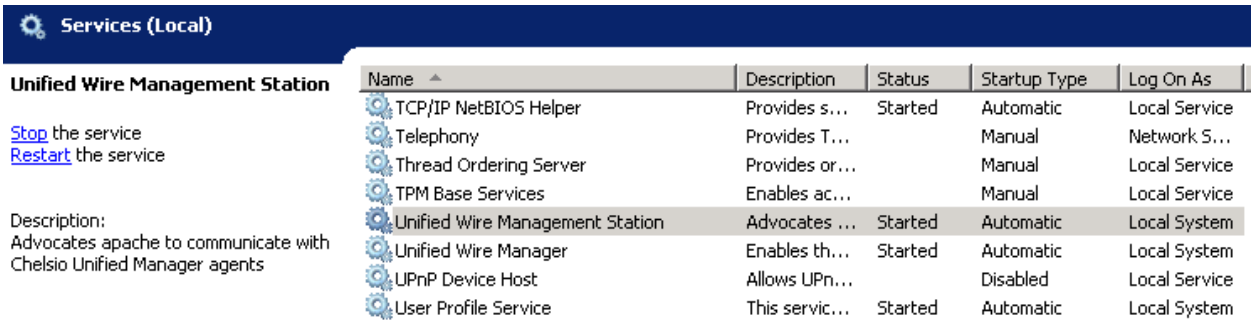

Figure 89 - Verifying Management Station service status

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:

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

6.2.2. Service start/stop/restart

1. Launch **Services** from the Control Panel. You can also type s*ervices.msc* in the **Run** command**.** This will open the **Services** window.
2. Scroll and highlight the "Unified Wire Manager" service.
3. 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

1. Open command prompt by typing *cmd* in the **Run** command.
2. 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

1. Launch **Services** from the Control Panel. You can also type s*ervices.msc* in the **Run** command. This will open the **Services** window.
2. Search for "*Apachex.x"* in the list of services and click *Start* in the **Action** menu on the left.

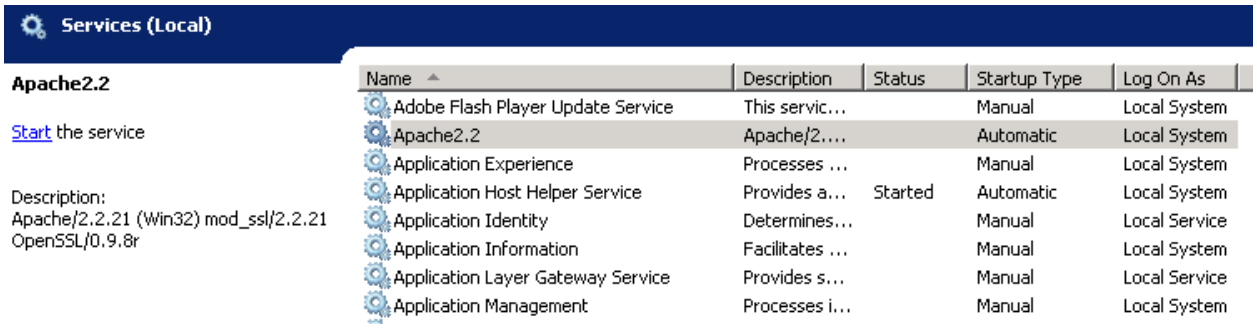

Figure 90 - Starting Apache HTTP server

3. Search for "*Unified Wire Management Station"* and click *Start* in the **Action** menu on the left.

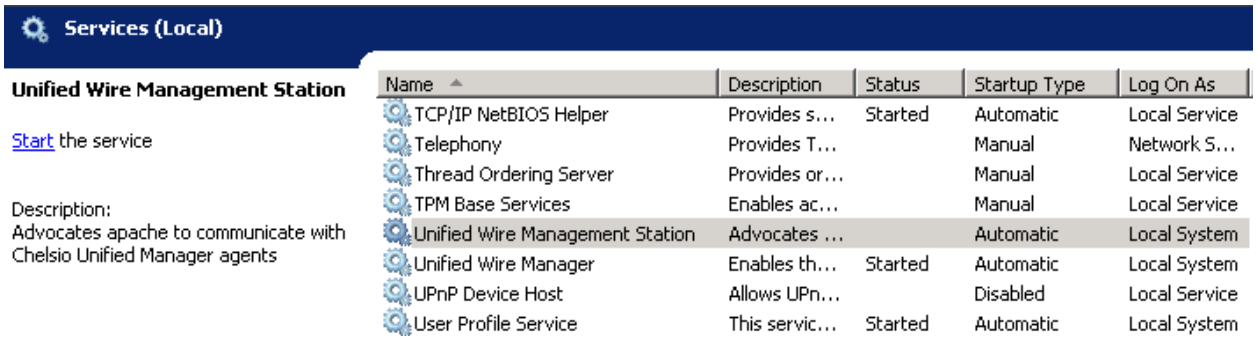

Figure 91 - Starting Management Station service

8.2. Accessing Web Management Interface

1. To access the Web GUI, type in the URL https://<management station IP address> in a web browser.
2. 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.

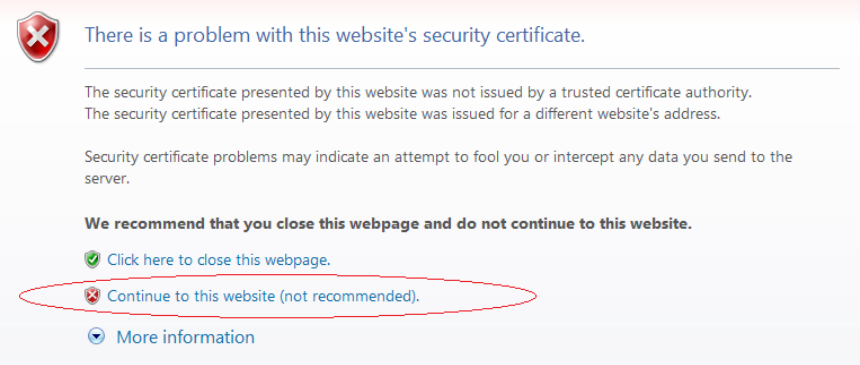

Figure 92 - Security Certificate prompt in Internet Explorer

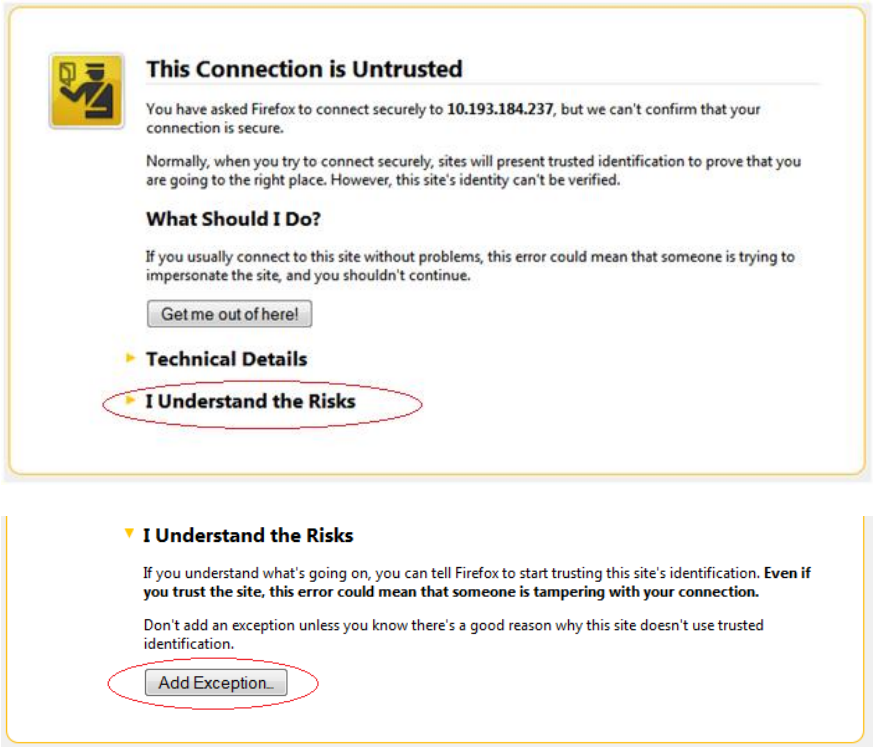

Figure 93 - Security Certificate prompt in Mozilla Firefox

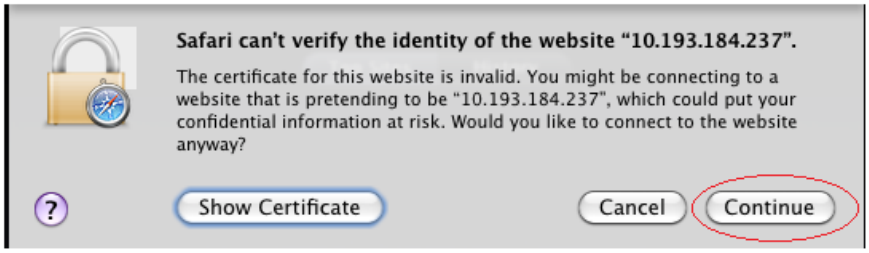

Figure 94 - Security Certificate prompt in Apple Safari

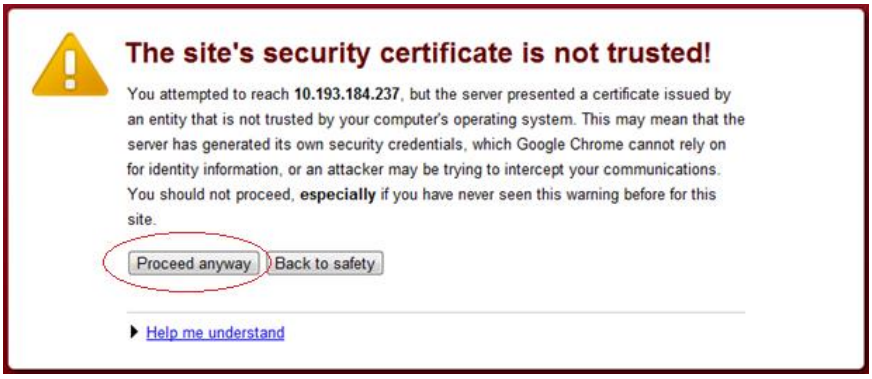

Figure 95 - Security Certificate prompt in Google Chrome

3. 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 96 - 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.

Figure 97 - 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.*

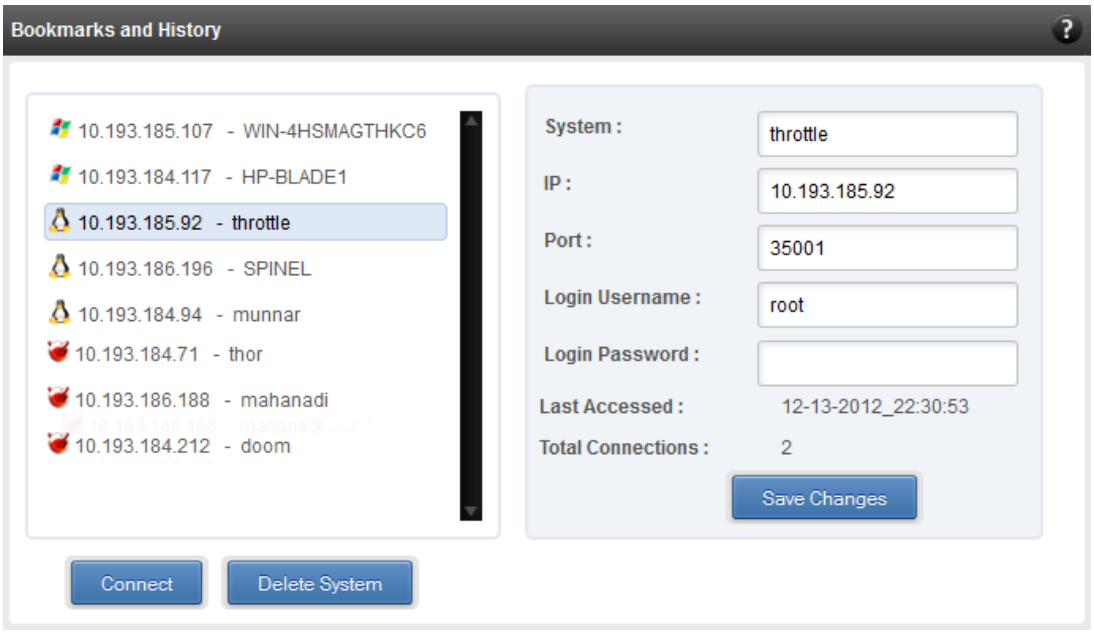

Figure 98 - Bookmarks and history module

- **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.

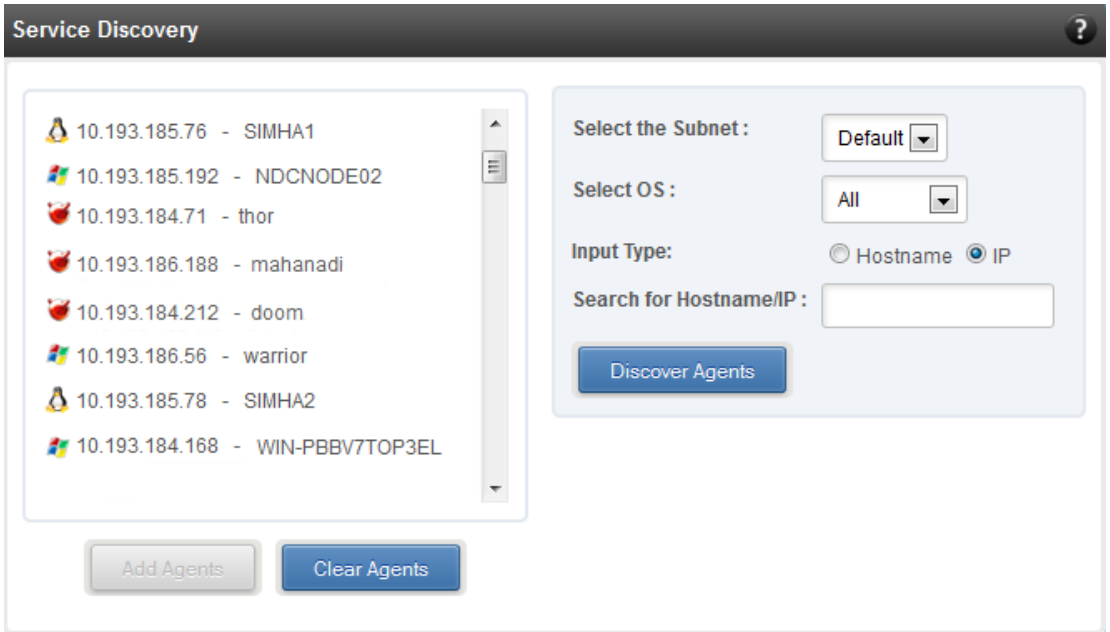

Figure 99 - Services Discovery module

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**

1. In the **Choose the card** fields, select T3 or T4/T5 depending on the chip revision of the network card.
2. 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.
3. Select a system or systems from the list and choose the driver to be installed in the **Driver Installation** section.
4. Download the appropriate driver from Chelsio's Download Center, service.chelsio.com.
5. Locate the driver package.
6. Click **Install** button to install the driver.

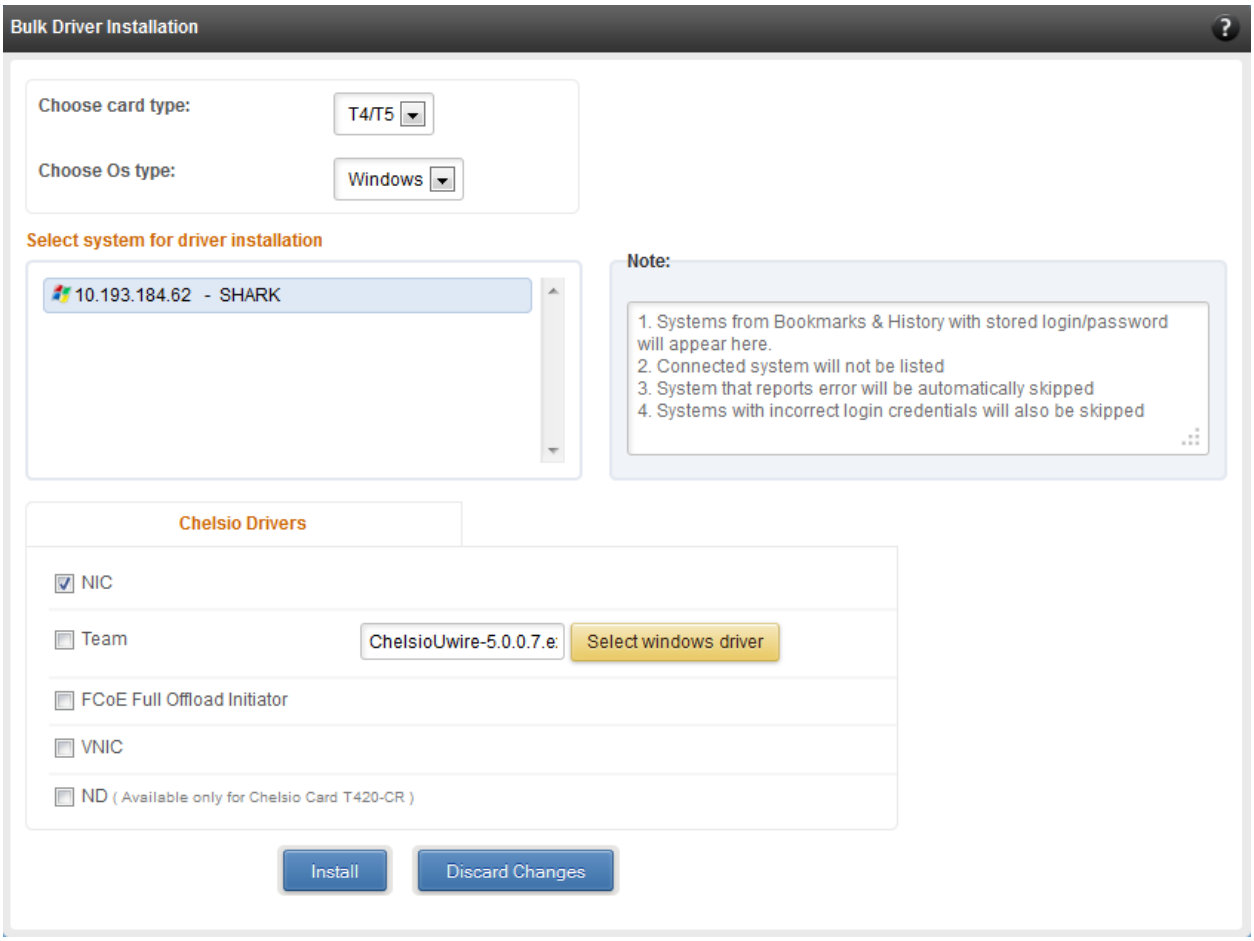

Figure 100 - Bulk Driver Installation module

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.

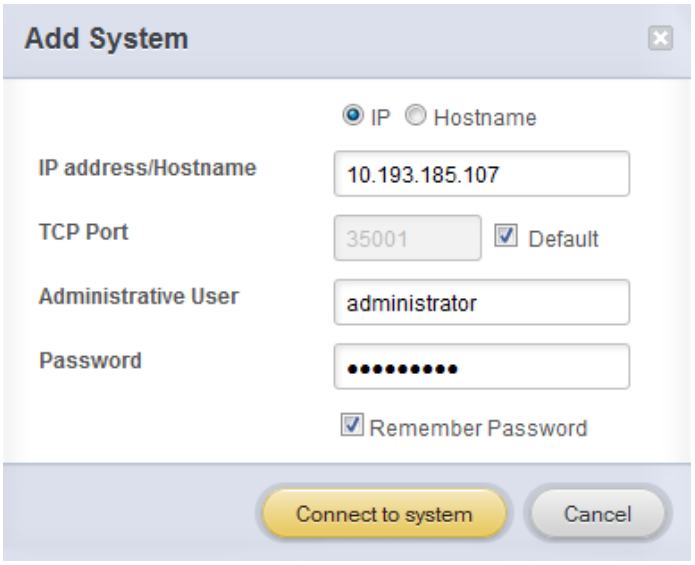

Figure 101 - Adding a UM Agent

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 Syste*m. Then click *Yes* to confirm.

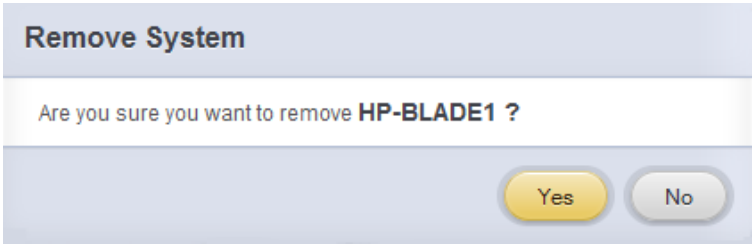

Figure 102 - Removing a UM Agent

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.**

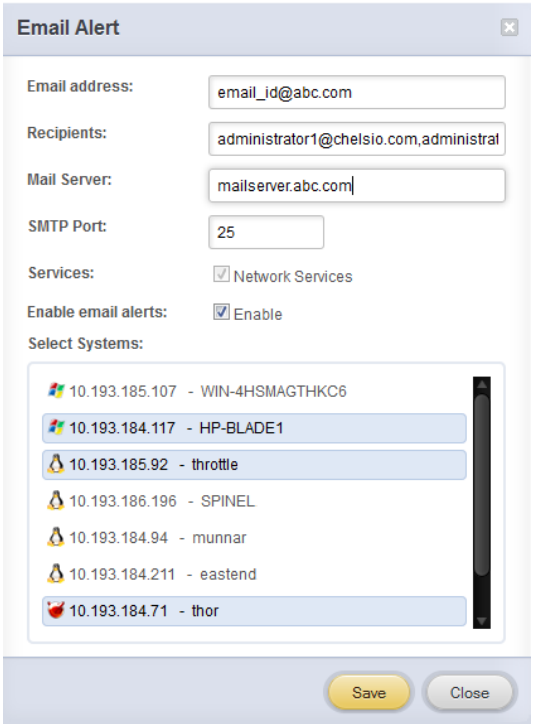

Figure 103 - Subscribing to Email Alerts

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).**

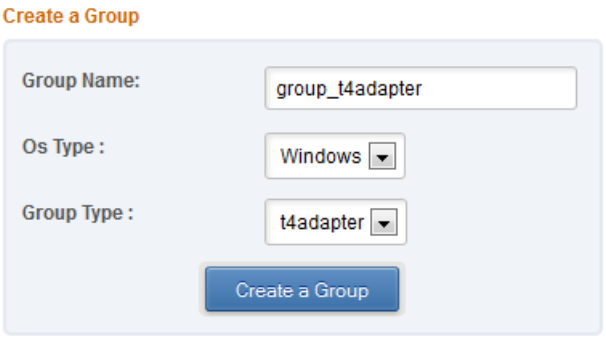

Figure 104 - Creating a group

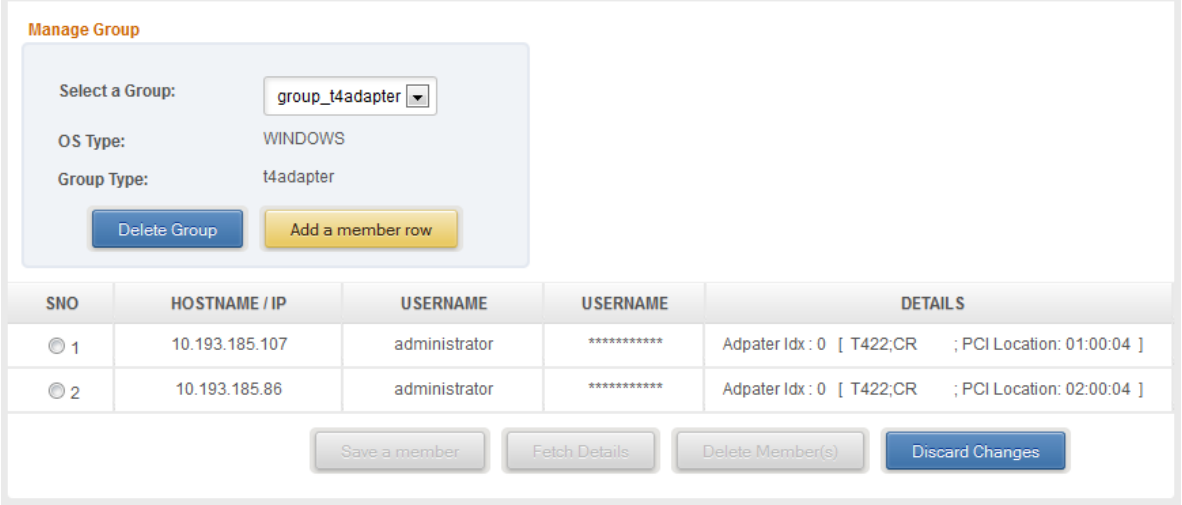

Figure 105 - Managing a group

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.

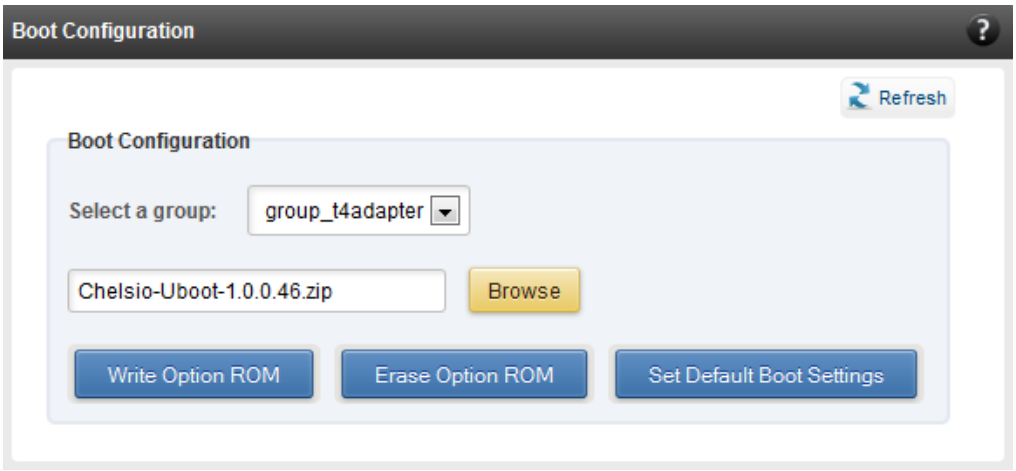

Figure 106 - Boot Configuration module

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.

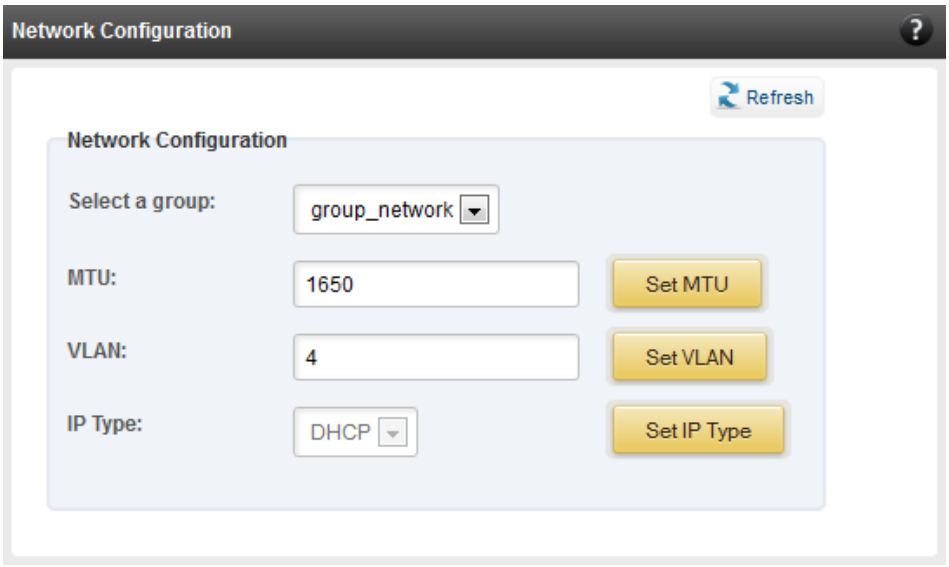

Figure 107 - Network Configuration module

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).

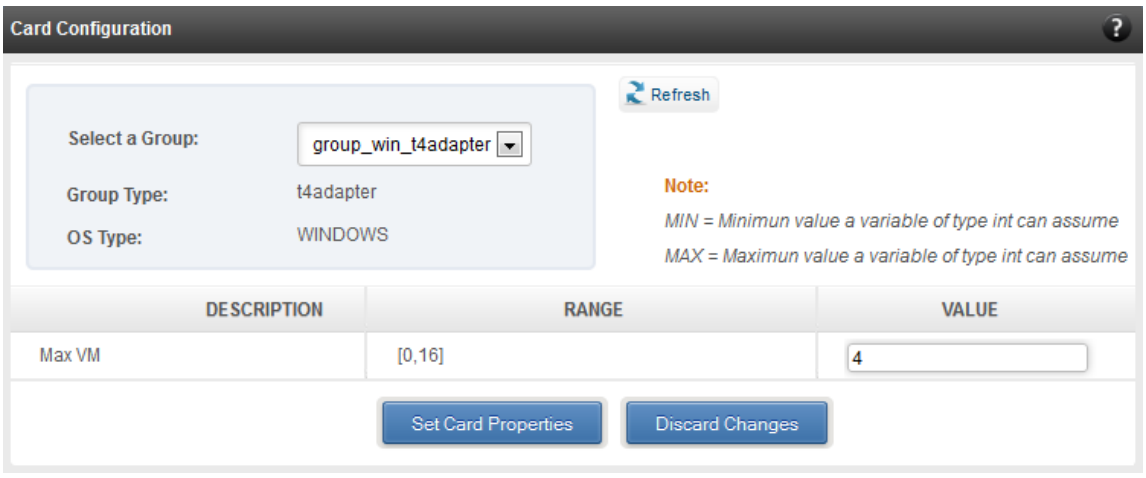

Figure 108 - Card Configuration module

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.

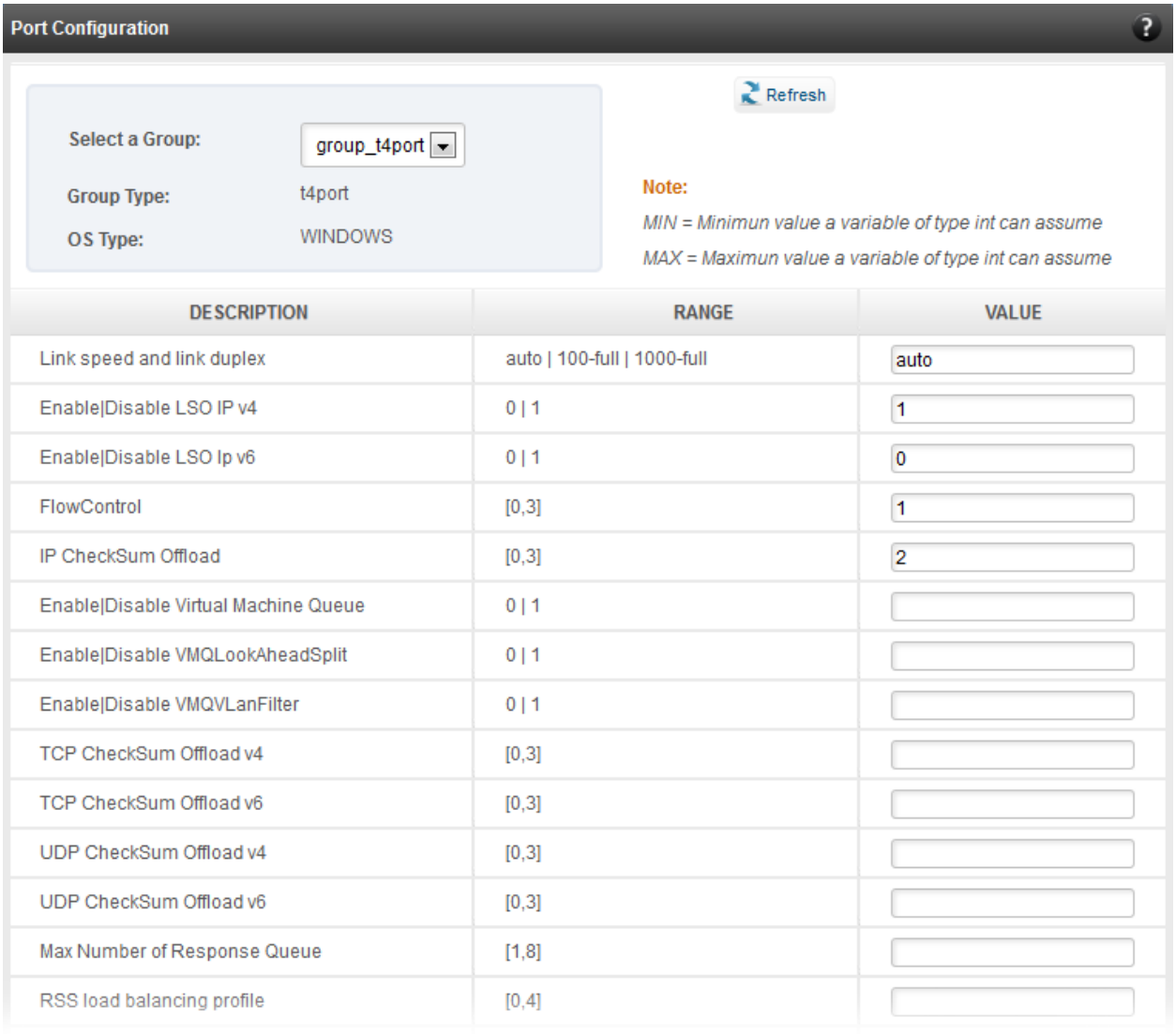

Figure 109 - Port Configuration module

Bypass Configuration

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

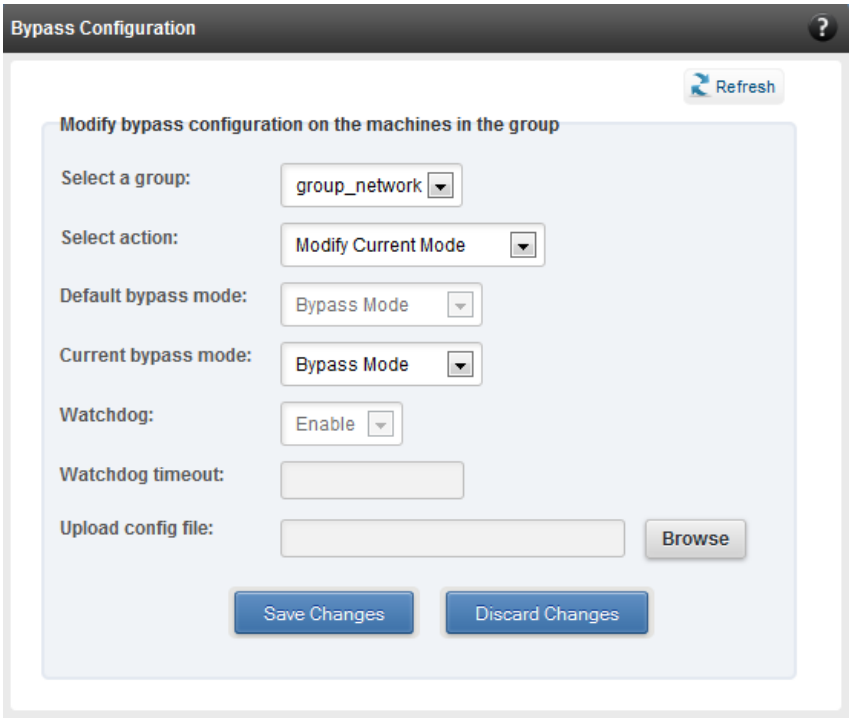

Figure 110 - Bypass Configuration module

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.

Figure 111 - System Summary module

8.5.2. Drivers Installation

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

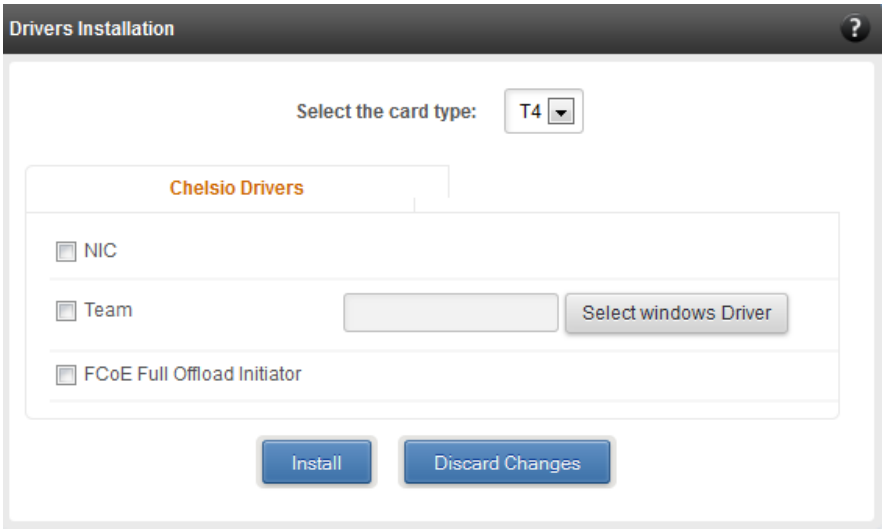

Figure 112 - Drivers Installation module connected to Windows Agent

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**.

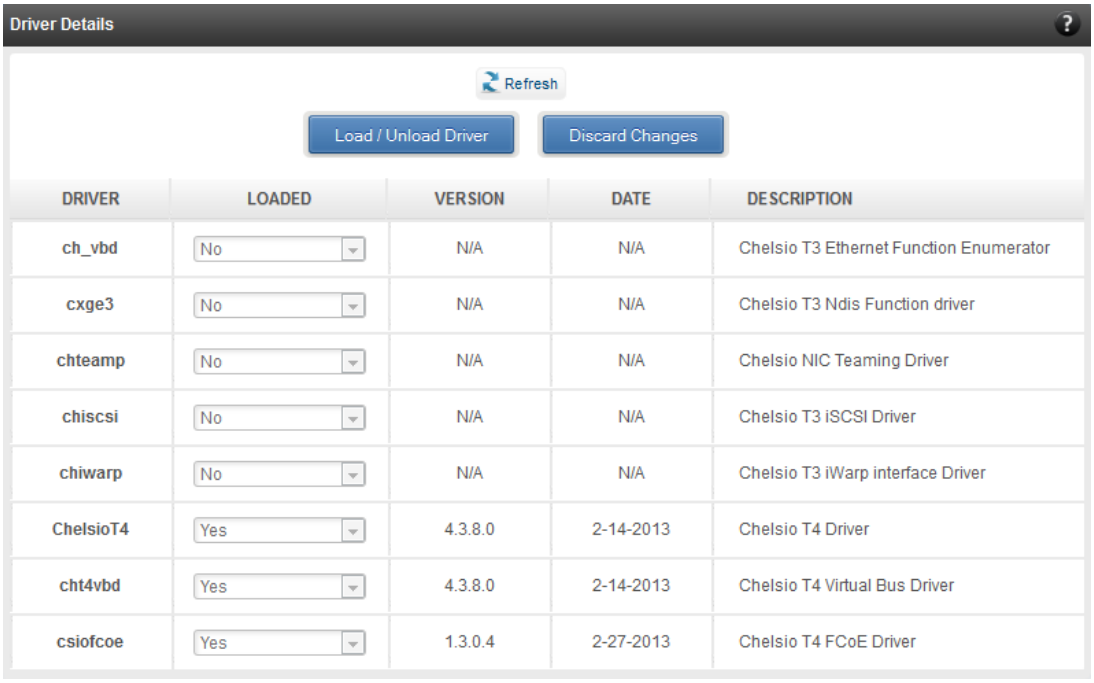

Figure 113 - Driver Details module

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.

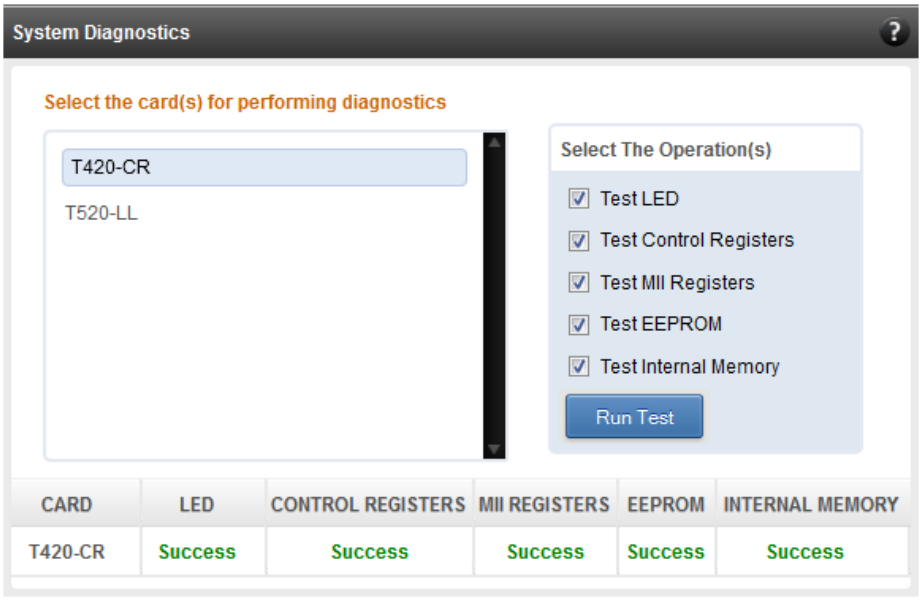

Figure 114 - System Diagnostics module for a T4 CNA

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.

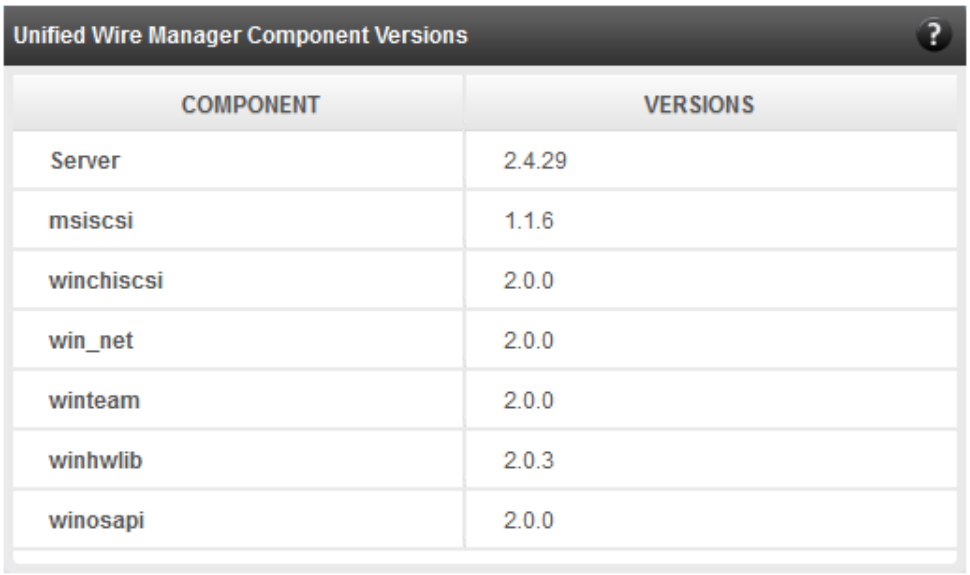

Figure 115 - Unified Wire Manager Component Versions module

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.

Figure 116 - 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.

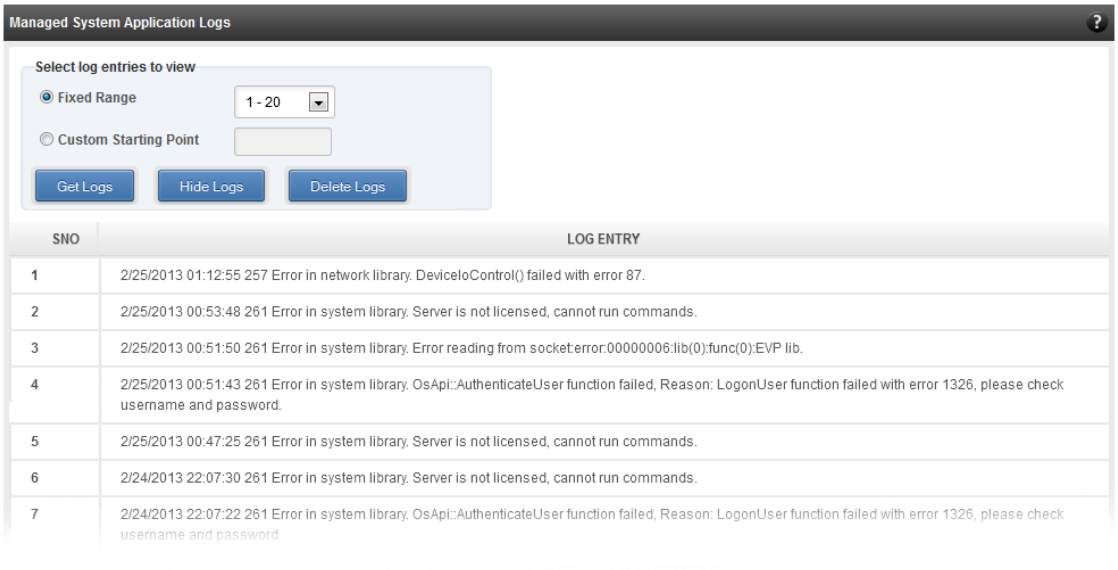

Figure 117 - Managed System Application Logs module for Windows Agent

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 118 - 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.

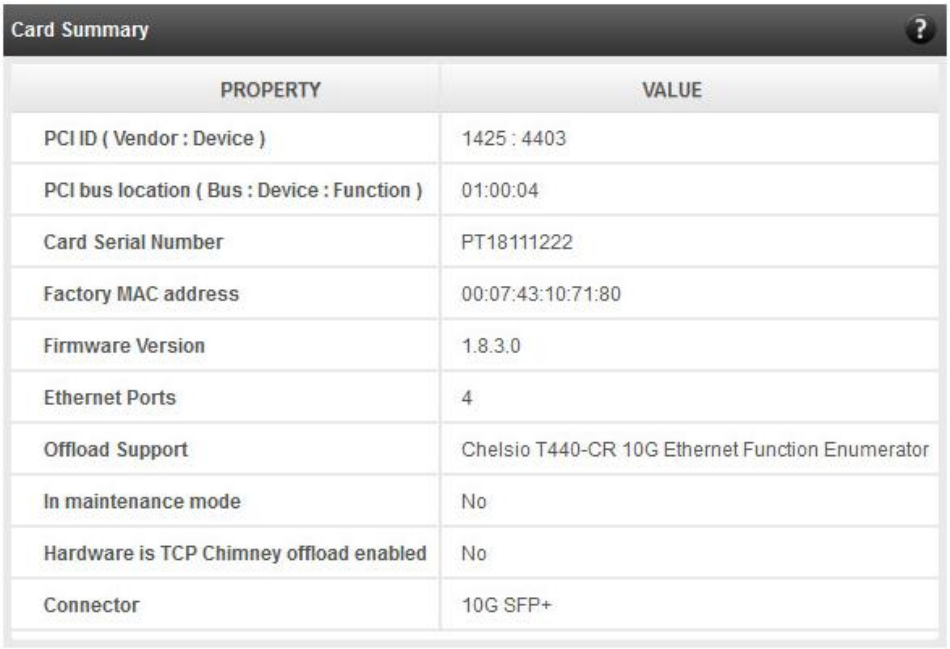

Figure 119 - Card Summary module for a Windows Agent

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.

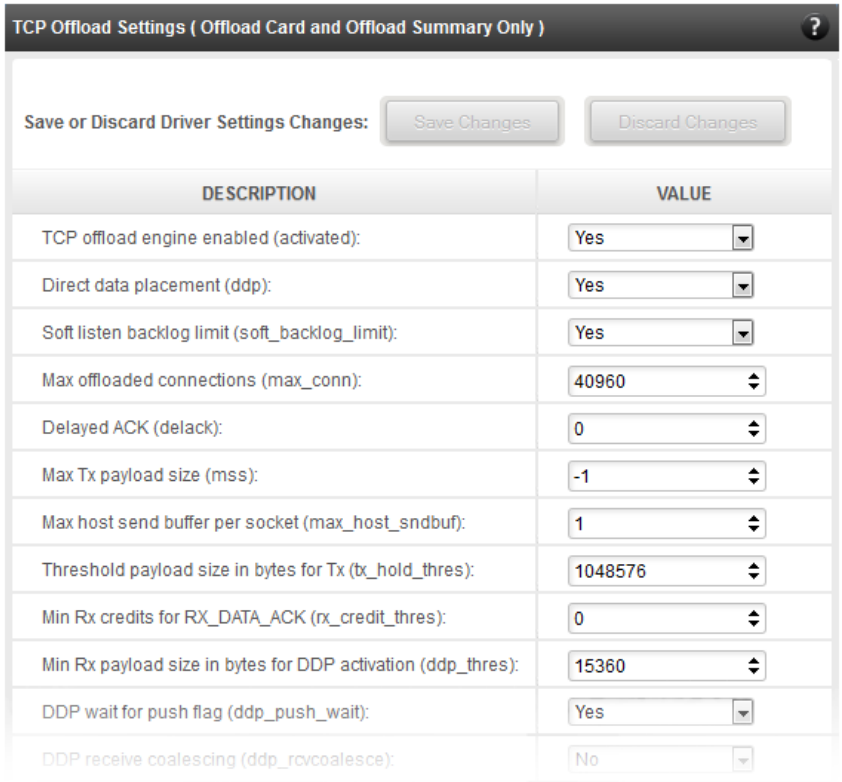

Figure 120 - TCP Offload Settings module for a FreeBSD Agent

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.

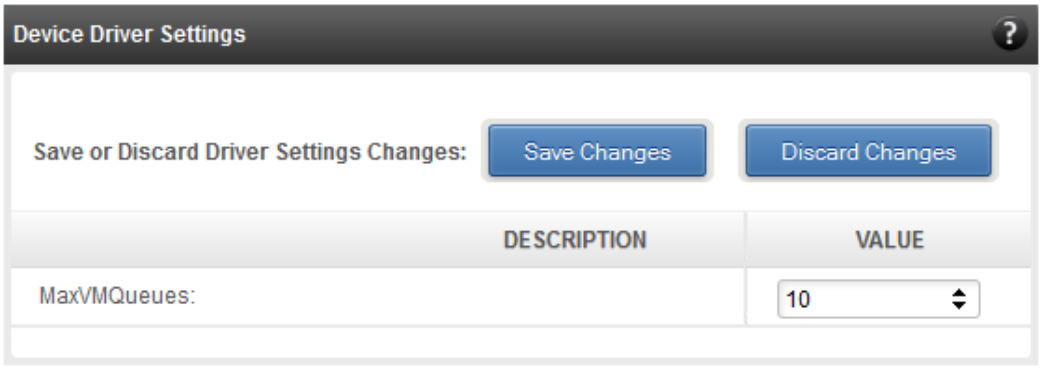

Figure 121 - Device Driver Settings module for a Windows Agent

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.

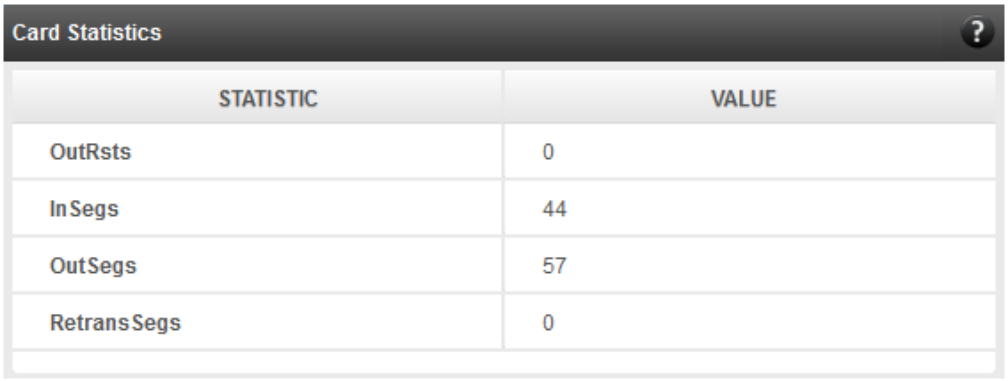

Figure 122 - Card Statistics module

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 123 - 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.

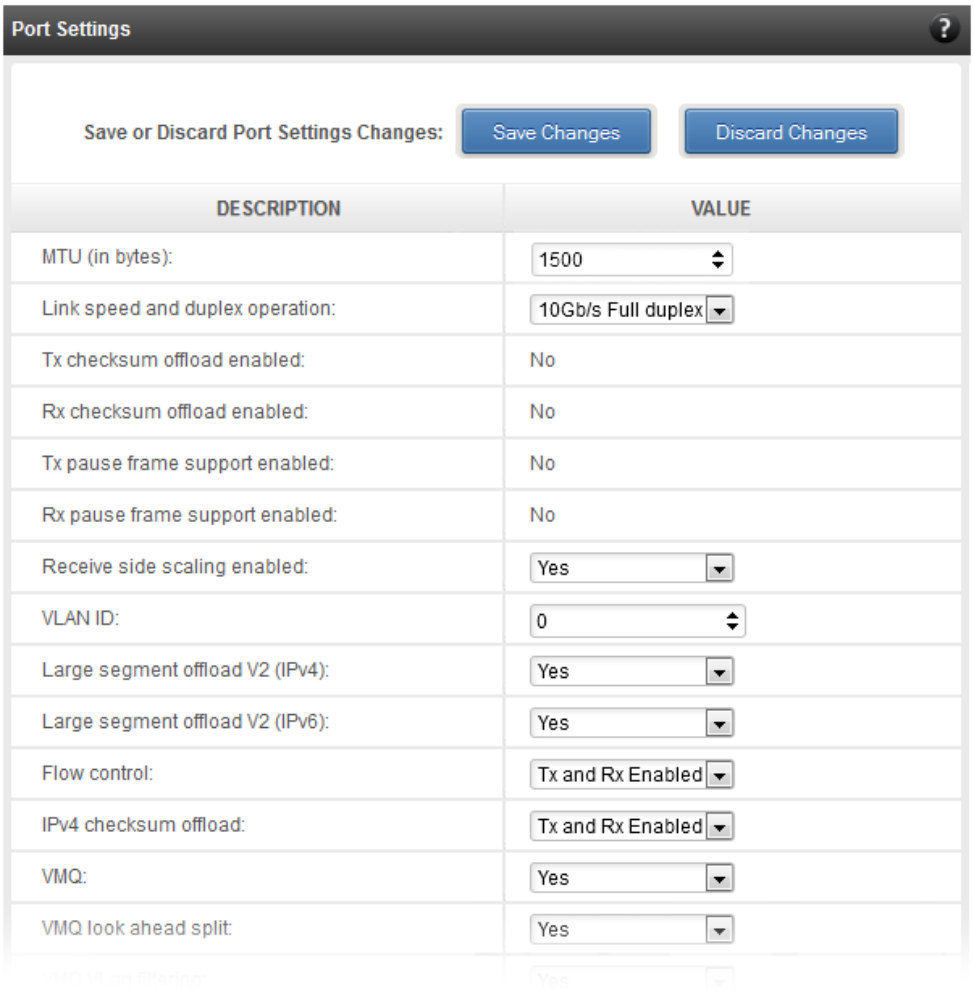

Figure 124 - Port Settings of T4 CNA

- **Port statistics**

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

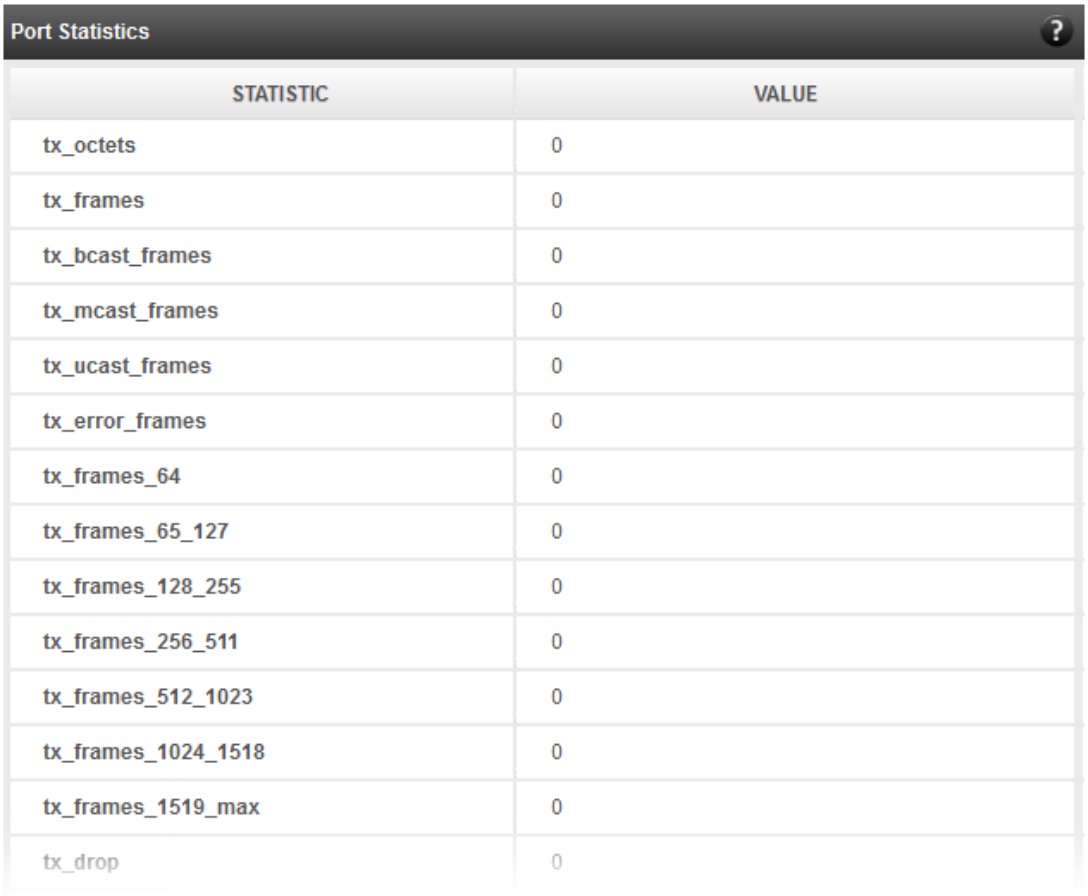

Figure 125 - Port Statistics of T4 CNA on Windows Agent

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.

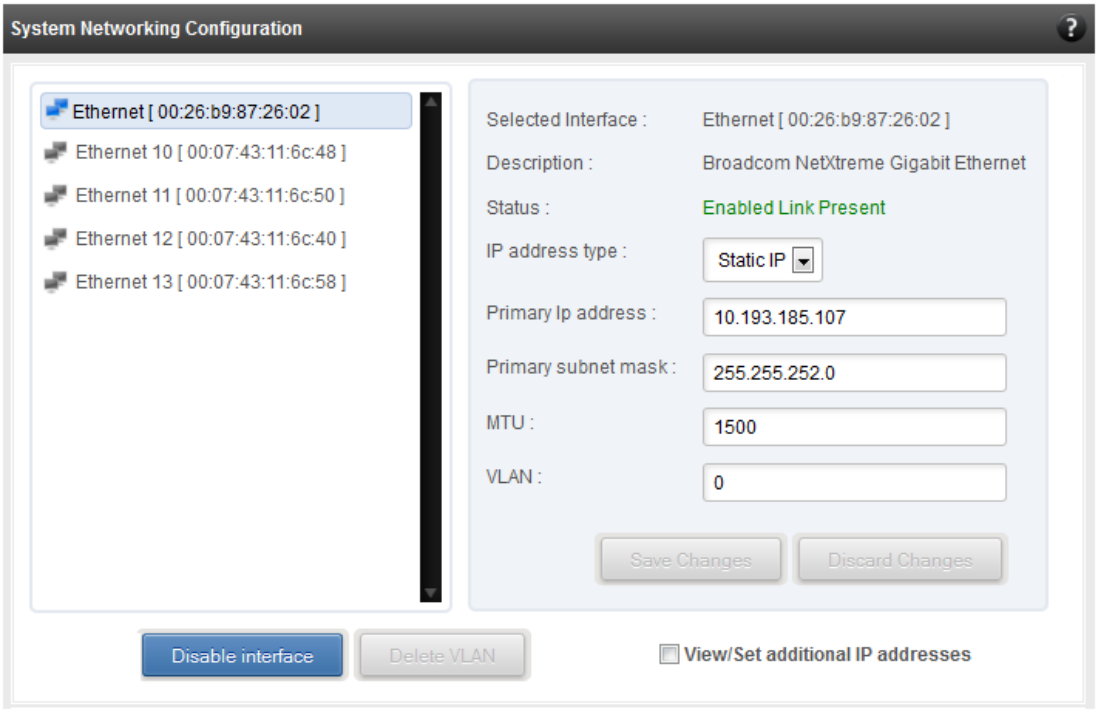

Figure 126 - System network configuration module

Figure 127 - 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.

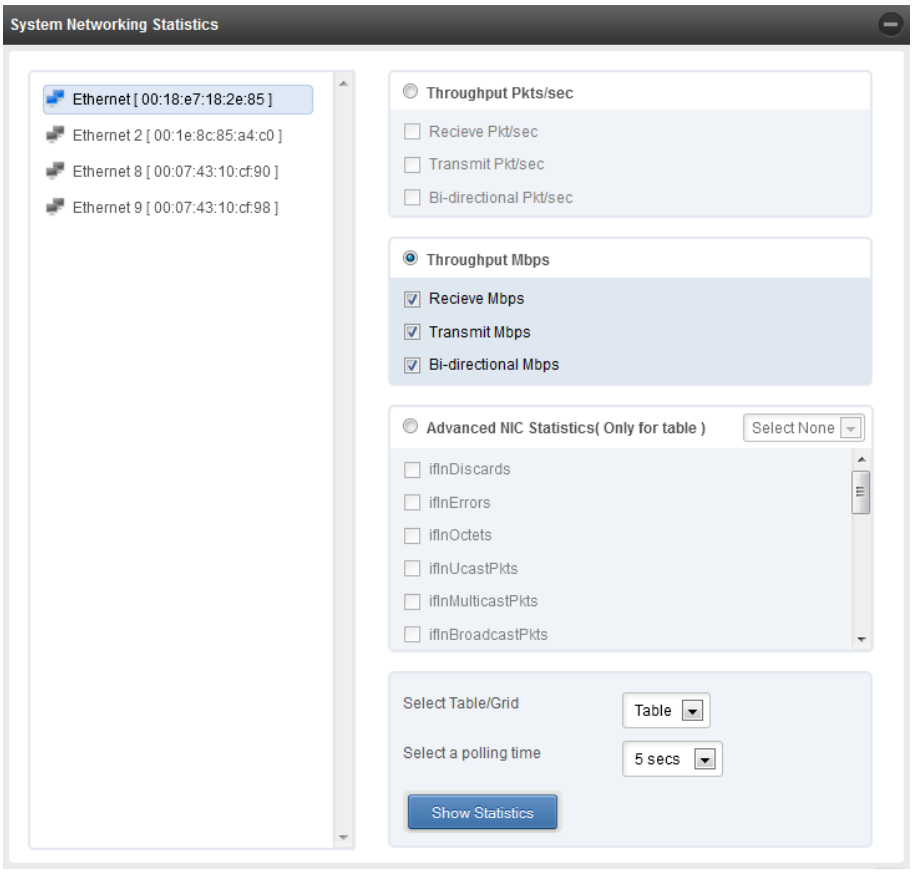

Figure 128 - System network statistics module

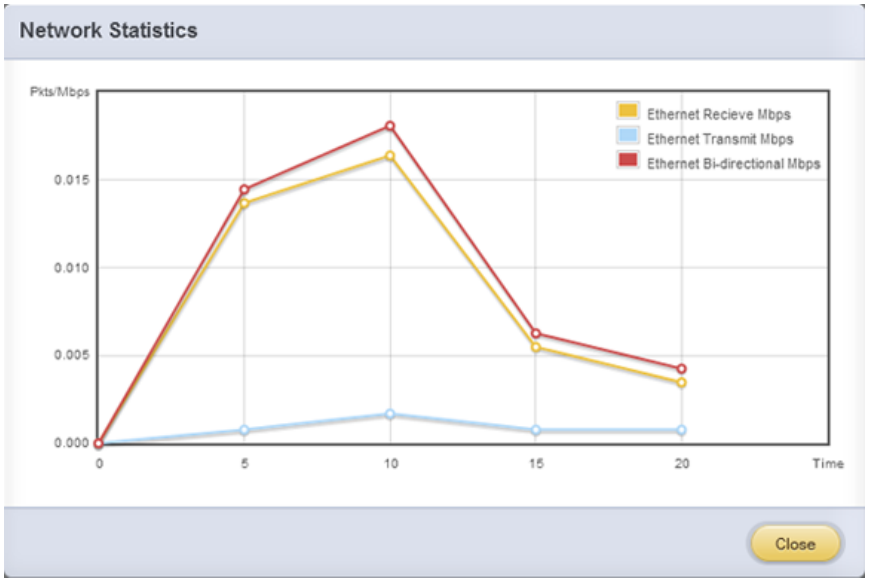

Figure 129 - Network Throughput Vs Time instant Graph

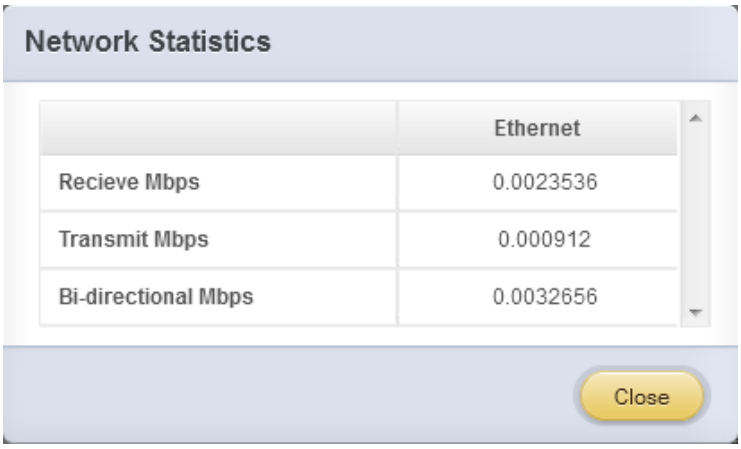

Figure 130 - Network Throughput Vs Time instant Table

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.

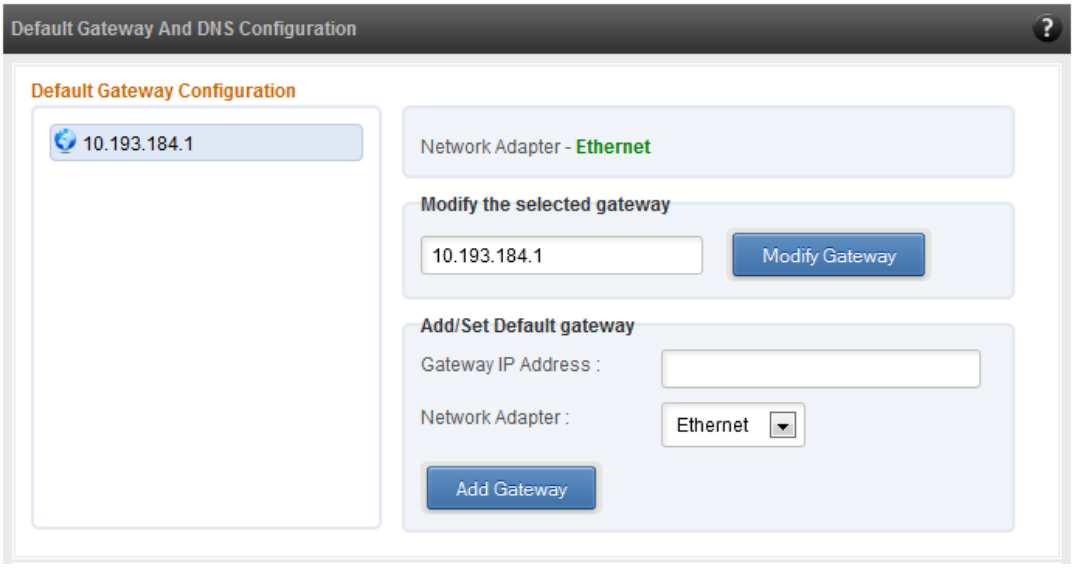

Figure 131 - Default gateway and DNS configuration module for a Windows Agent

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.

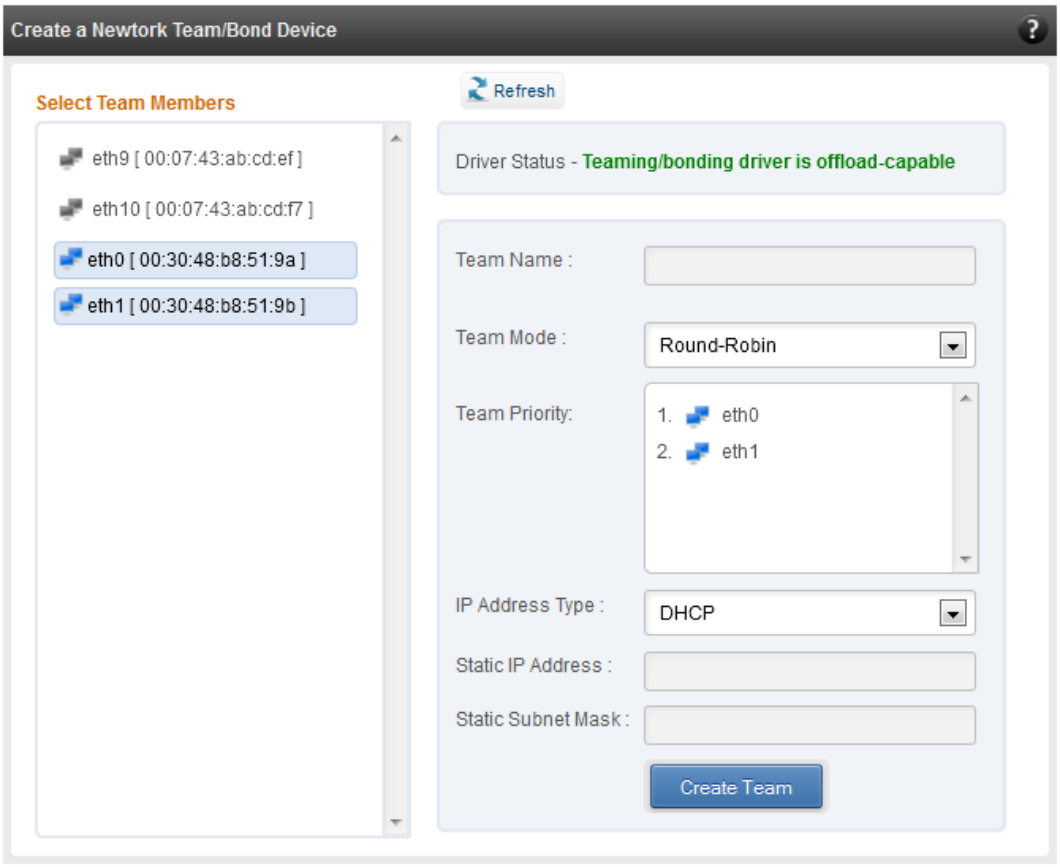

Figure 132 - Create a network team/bond device module for Linux Agent

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.

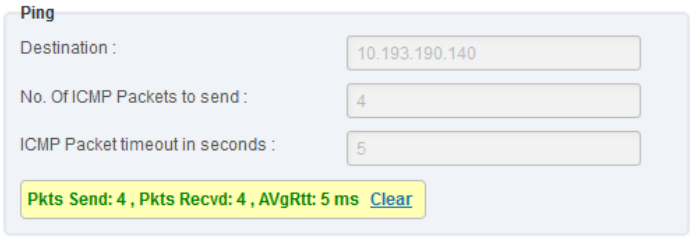

Figure 133 - Ping Utility

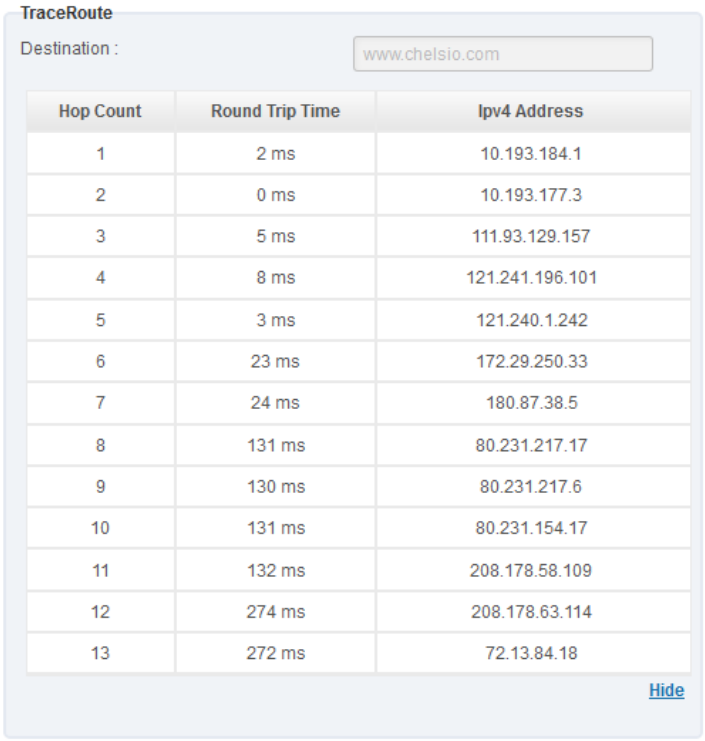

Figure 134 - TraceRoute Utility

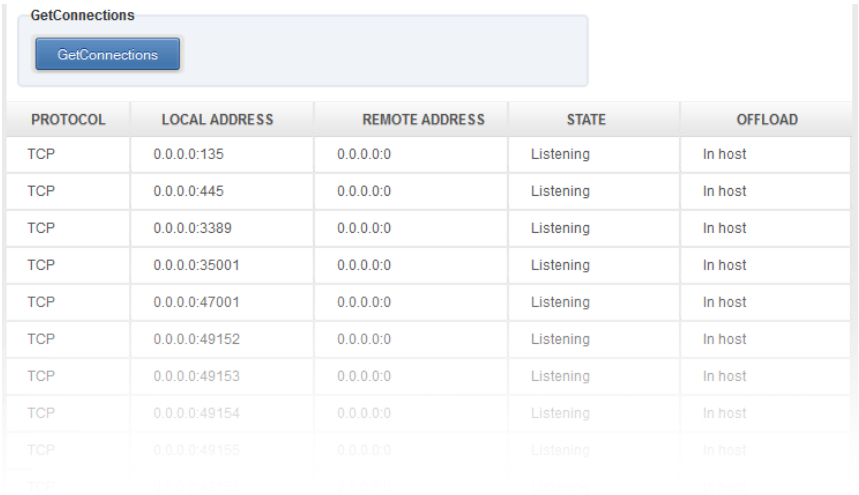

Figure 135 - GetConnections Utility

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.

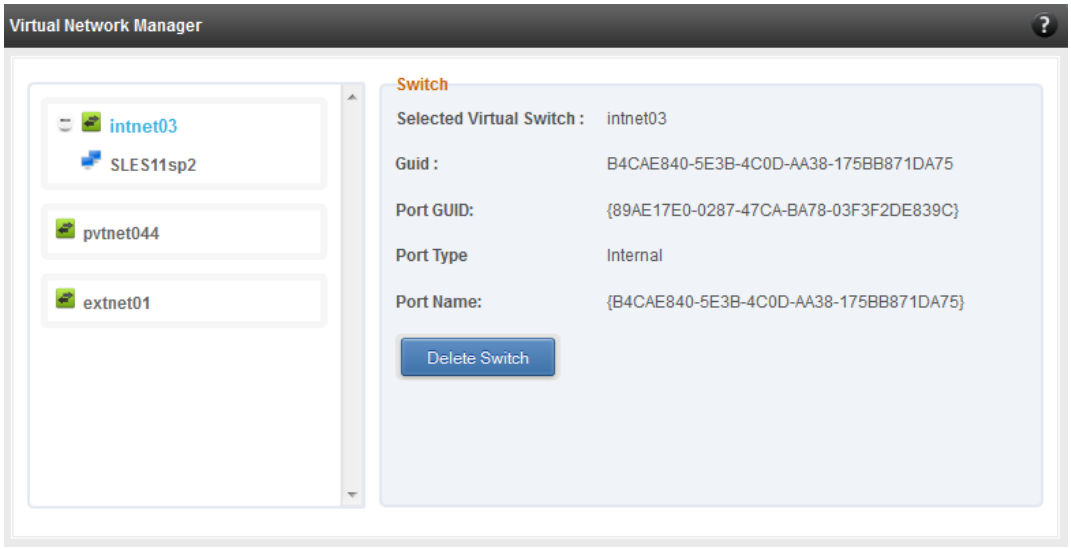

Figure 136 - Virtual Network Manager module

- **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.

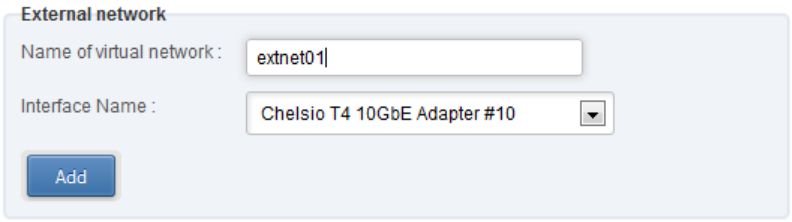

Figure 137 - Creating external virtual network

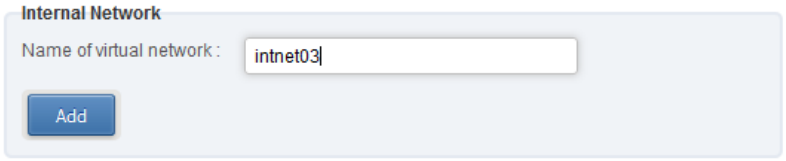

Figure 138 - Creating internal virtual network

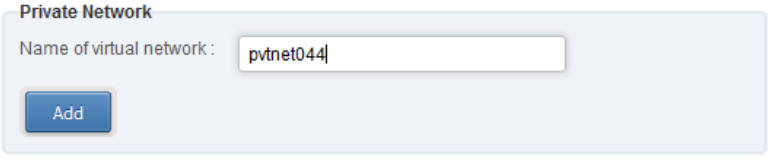

Figure 139 - Creating private virtual network

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*.

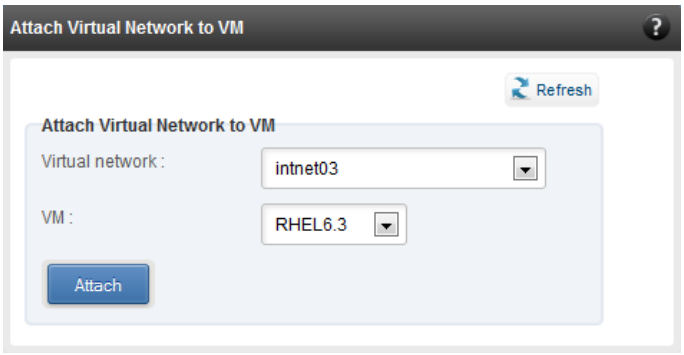

Figure 140 - Attaching Virtual Network to VM

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.

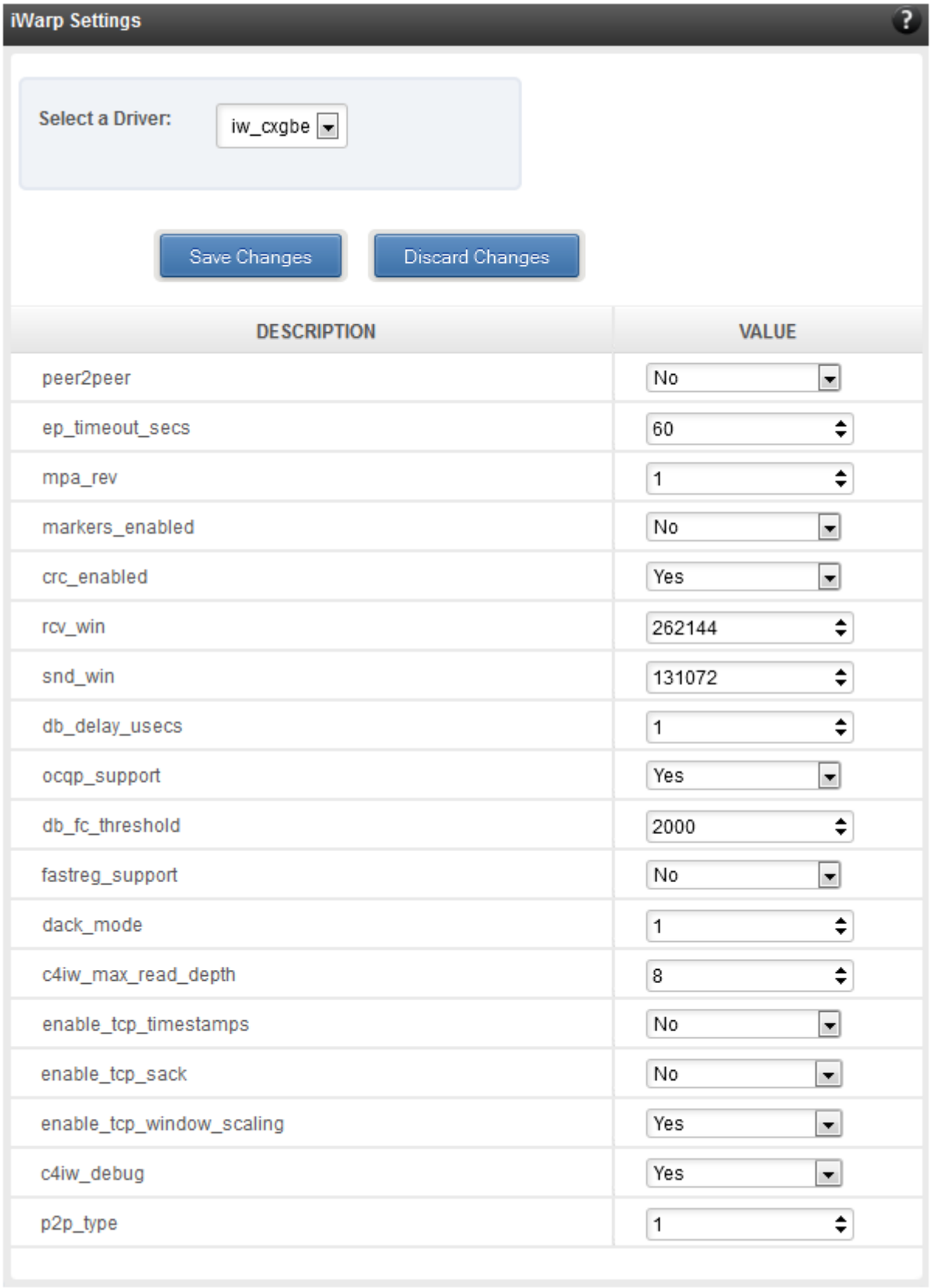

Figure 141 - iWARP settings for T4 CNA for FreeBSD Agent

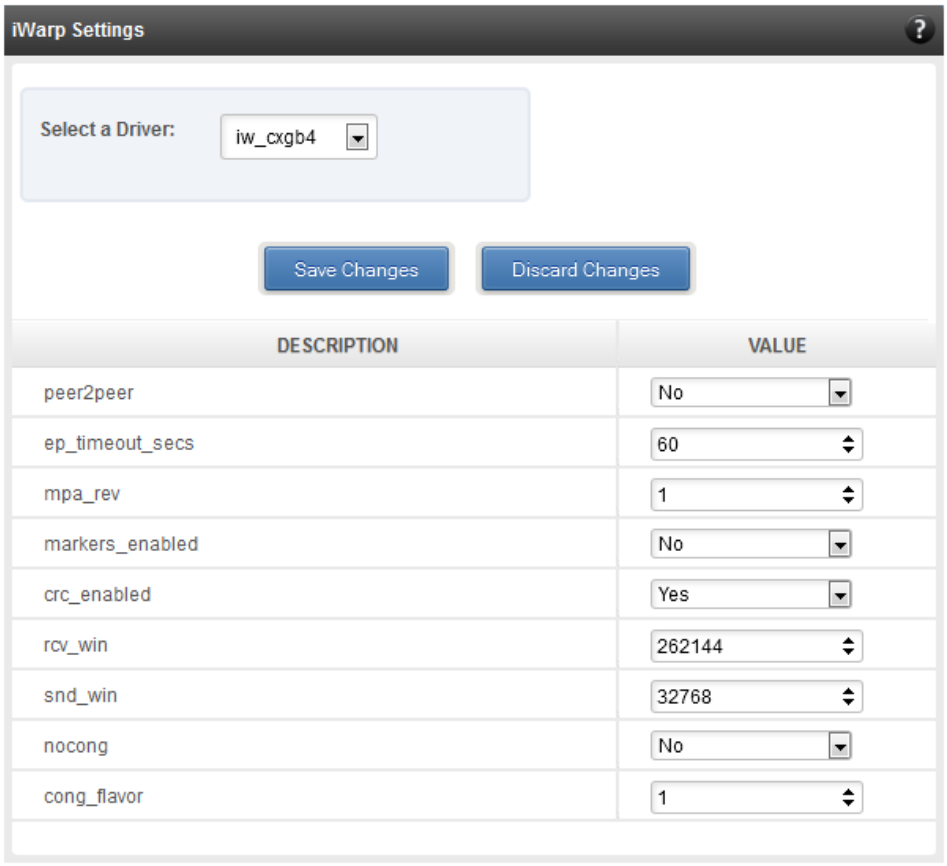

Figure 142 - iWARP settings for T4 CNA for Linux Agent

8.7. Storage

Storage Summary

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

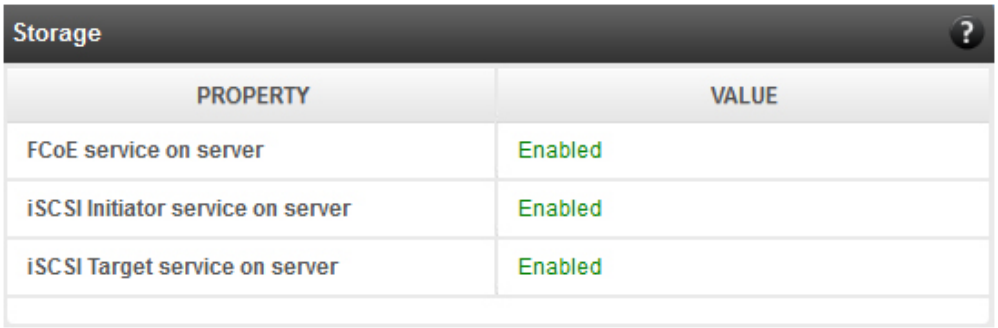

Figure 143 - Storage Summary Module

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 144 - 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 145 - 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.

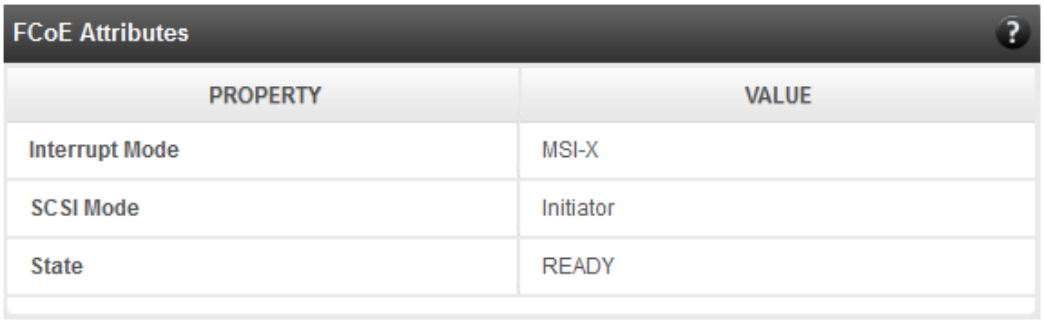

Figure 146 - FCoE Attributes module

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.

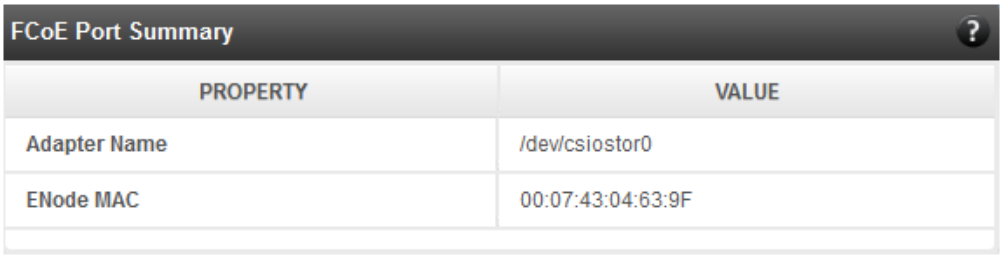

Figure 147 - FCoE Port Summary module for Linux Agent

- **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.

FCoE Port Attributes

PROPERTY	VALUE
State	Operational
NodeWWN	50:00:74:30:46:39:F0:00
PortWWN	50:00:74:30:46:39:F0:80
NPort MAC Address	0E:FC:03:53:00:23
Vlan ID	2
Fabric Name	20:02:00:05:73:D5:7A:C1
NPort ID	53:00:23
Type	NPort
Supported Class of Service	3
OS Device Name	/sys/class/fc_host/host119
Speed	10 GBPS
Maximum Frame Size	2128
No. of SCSI Targets	0

Figure 148 - 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**.

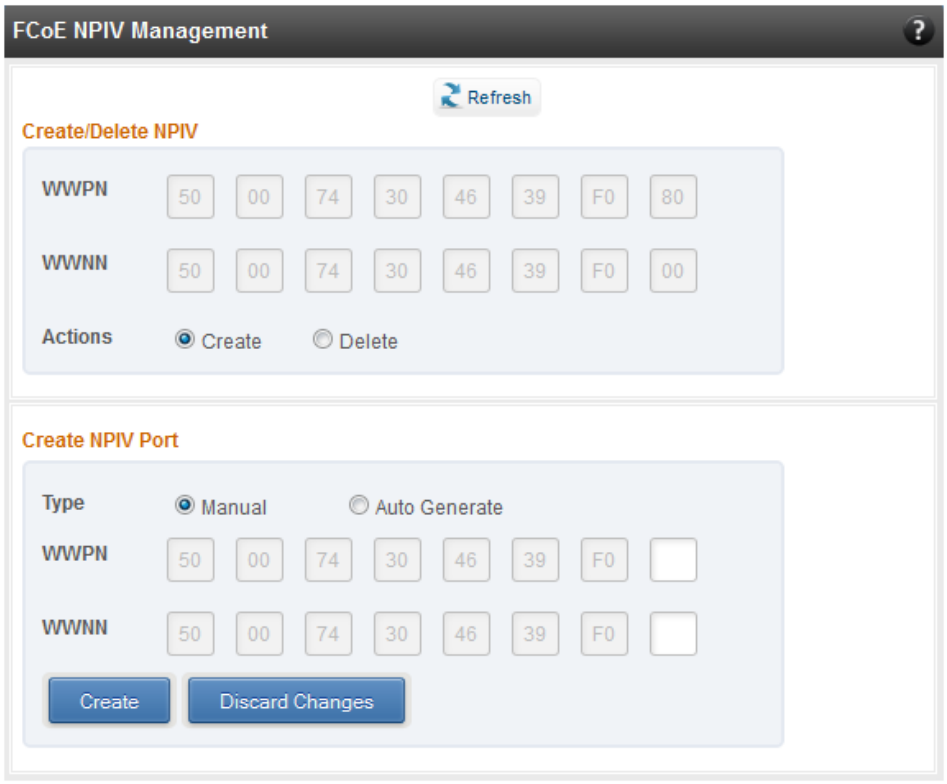

Figure 149 - FCoE NPIV management module

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.

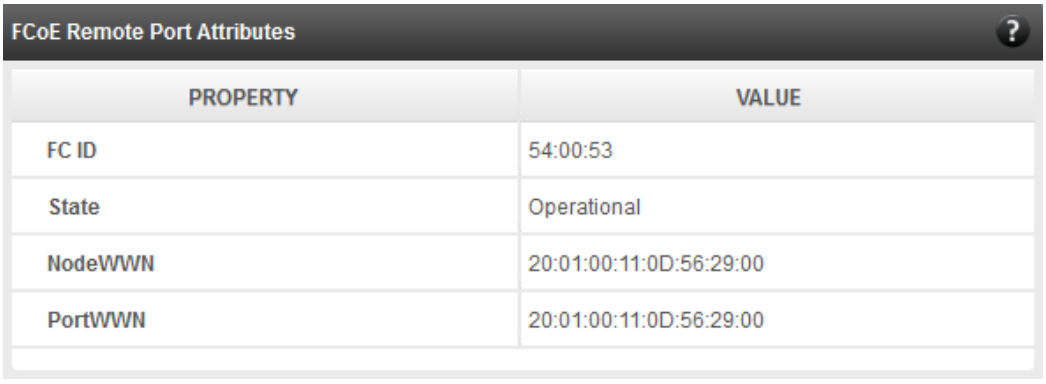

Figure 150 - FCoE Remort Port Attributes module

- **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.

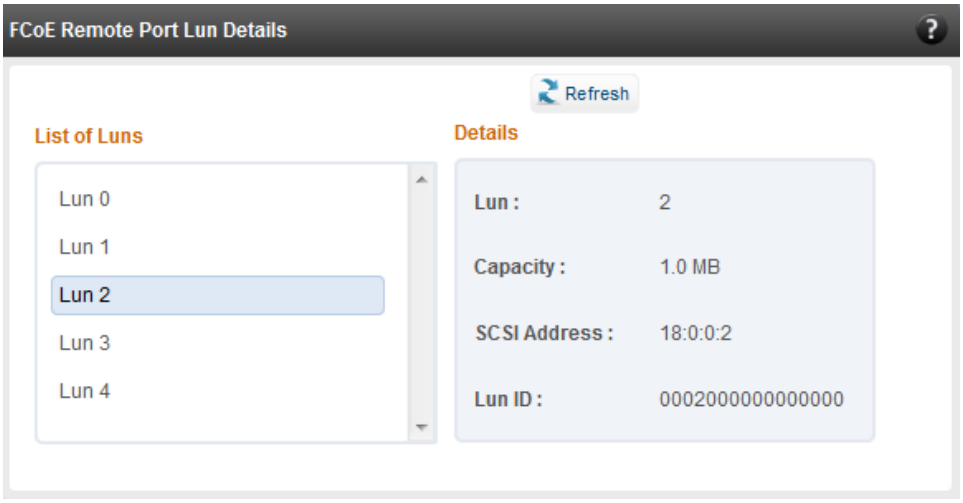

Figure 151 - FCoE Remote Port Lun Details module

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.

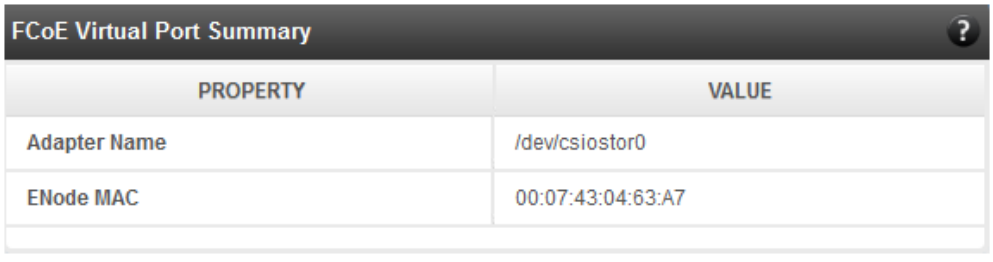

Figure 152 - FCoE Virtual Port Summary module for Linux Agent

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.

FCoE Virtual Port Attributes

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 153 - 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.

FCoE Remote Port Attributes

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 154 - 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.

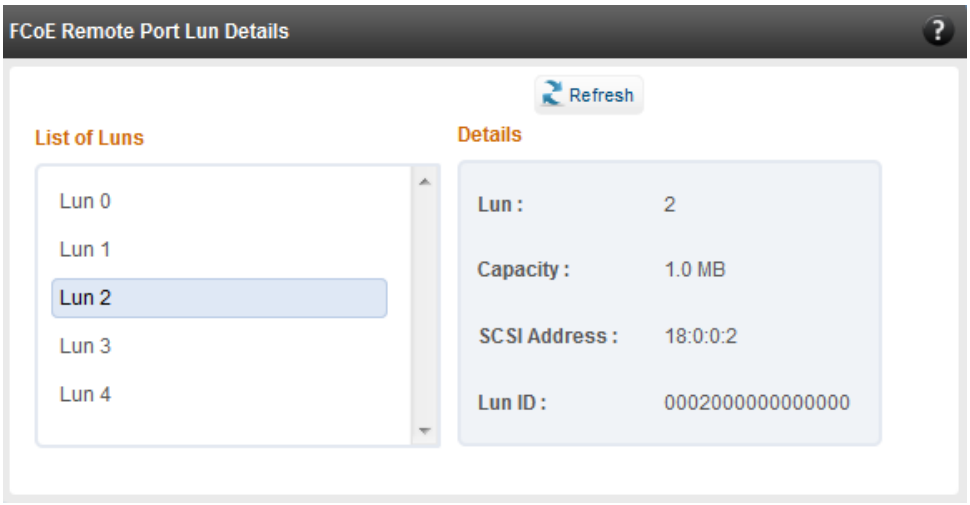

Figure 155 - FCoE Remote Port Lun Details module

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.".

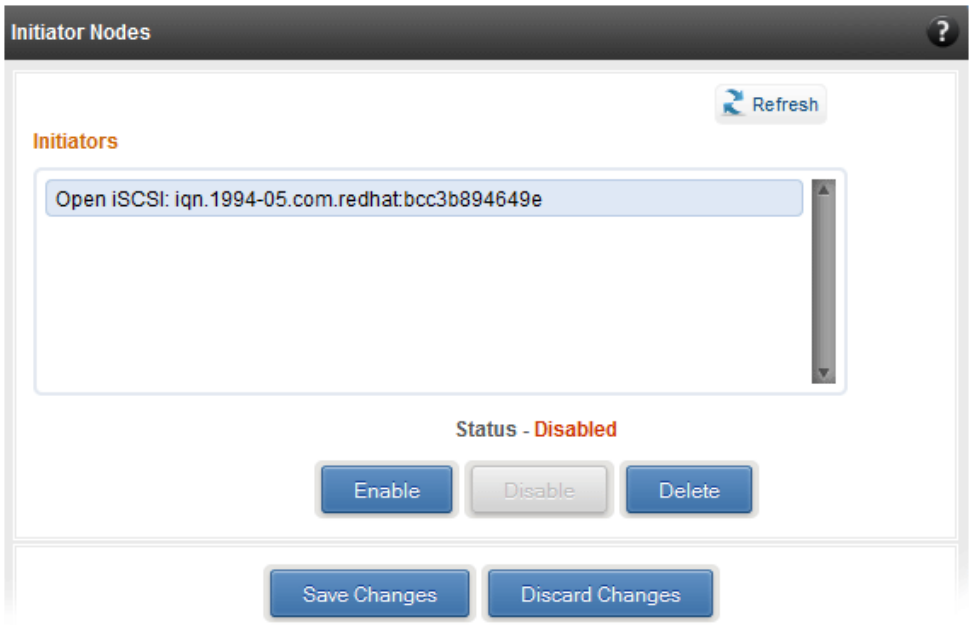

Figure 156 - Open iSCSI initiator

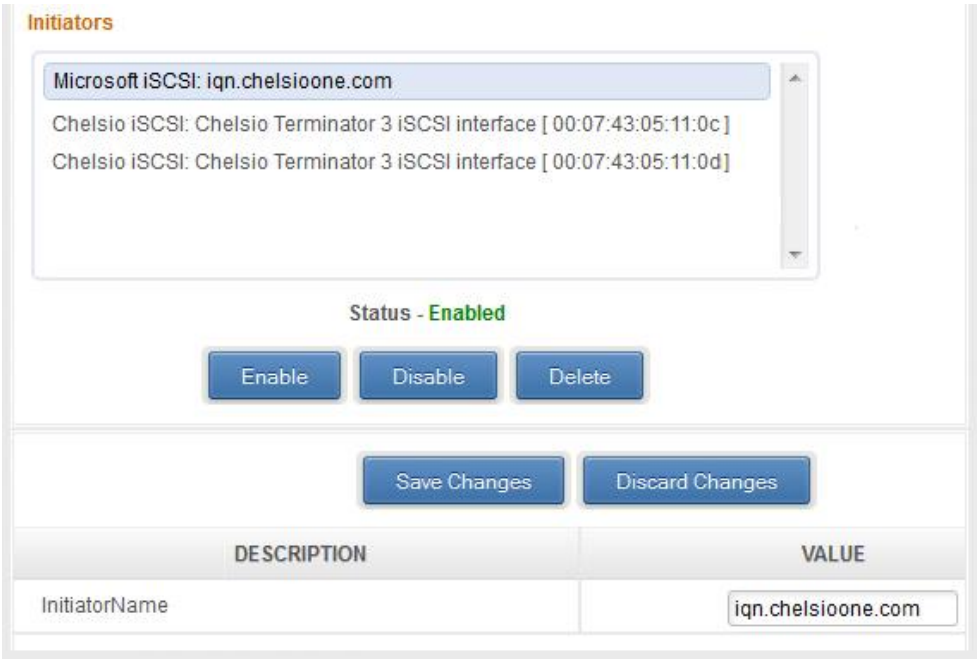

Figure 157 - Microsoft iSCSI initiator

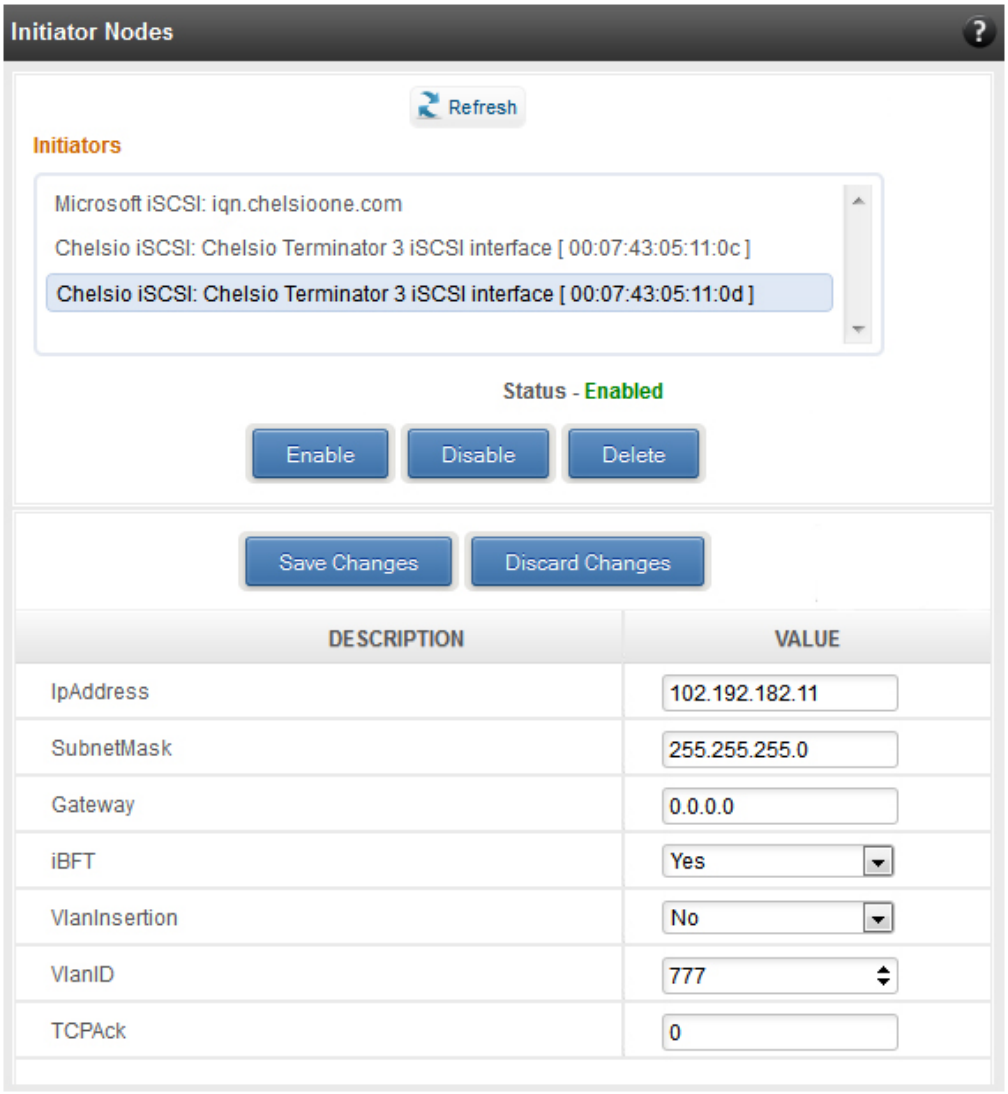

Figure 158 - Chelsio iSCSI initiator

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.*

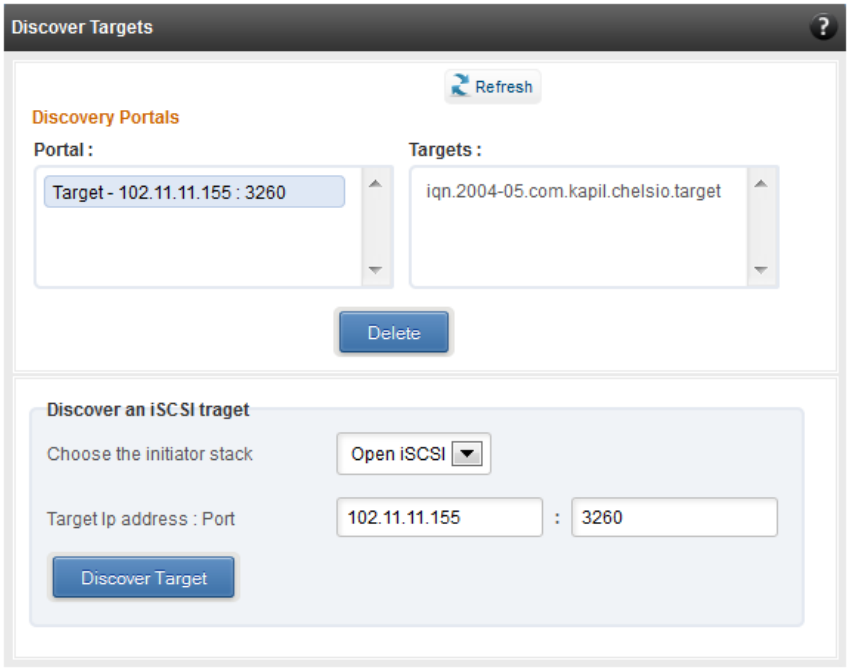

Figure 159 - Discover targets module

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.

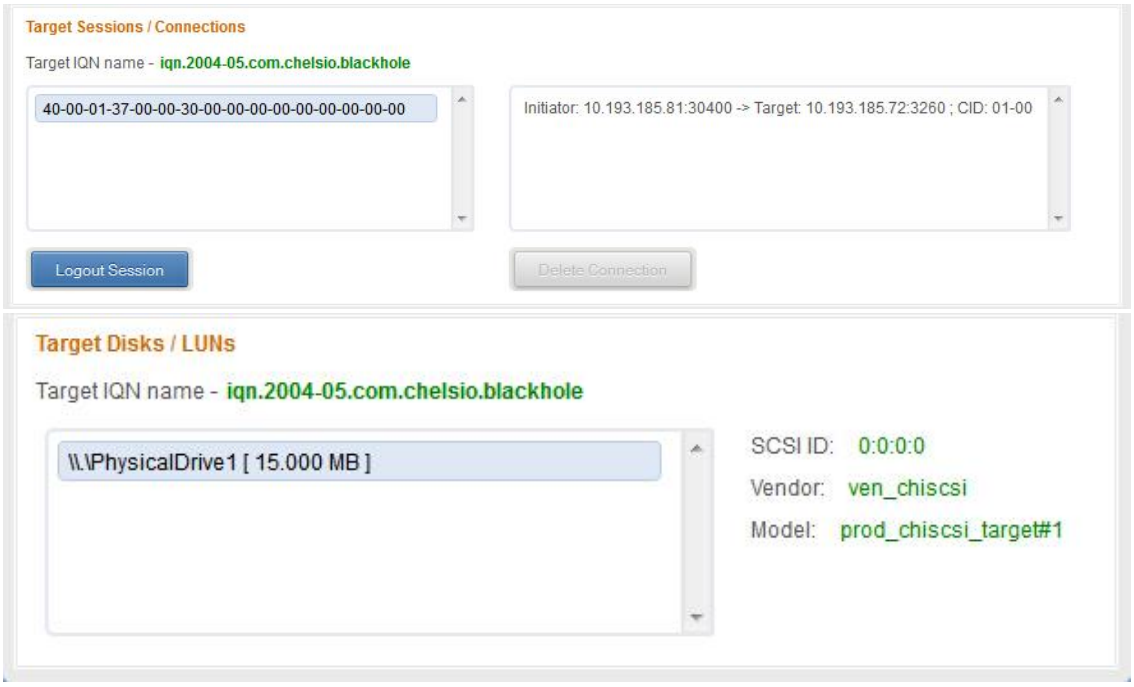

Figure 160 - Targets module after logging in

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.

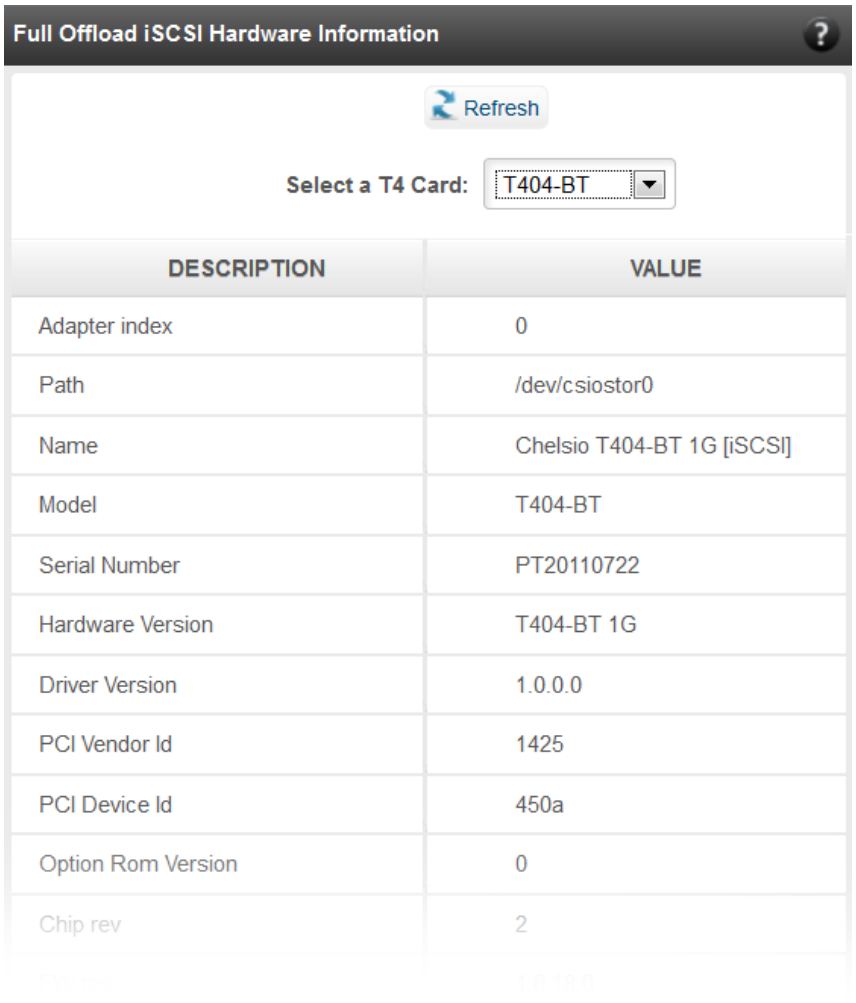

Figure 161 - Full Offload iSCSI Hardware Information module

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.

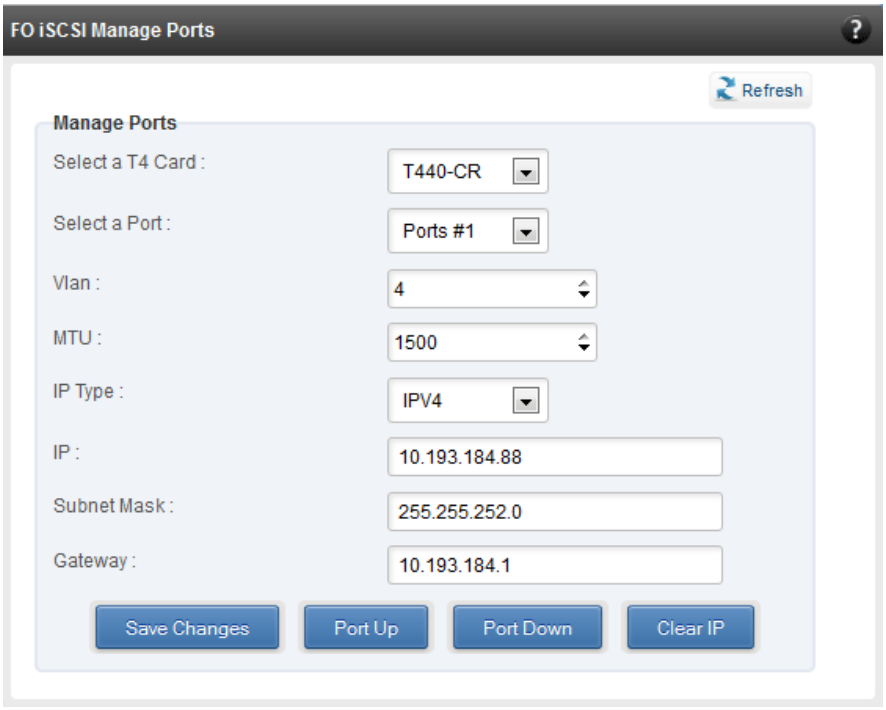

Figure 162 - FO iSCSI Manage Ports module

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.

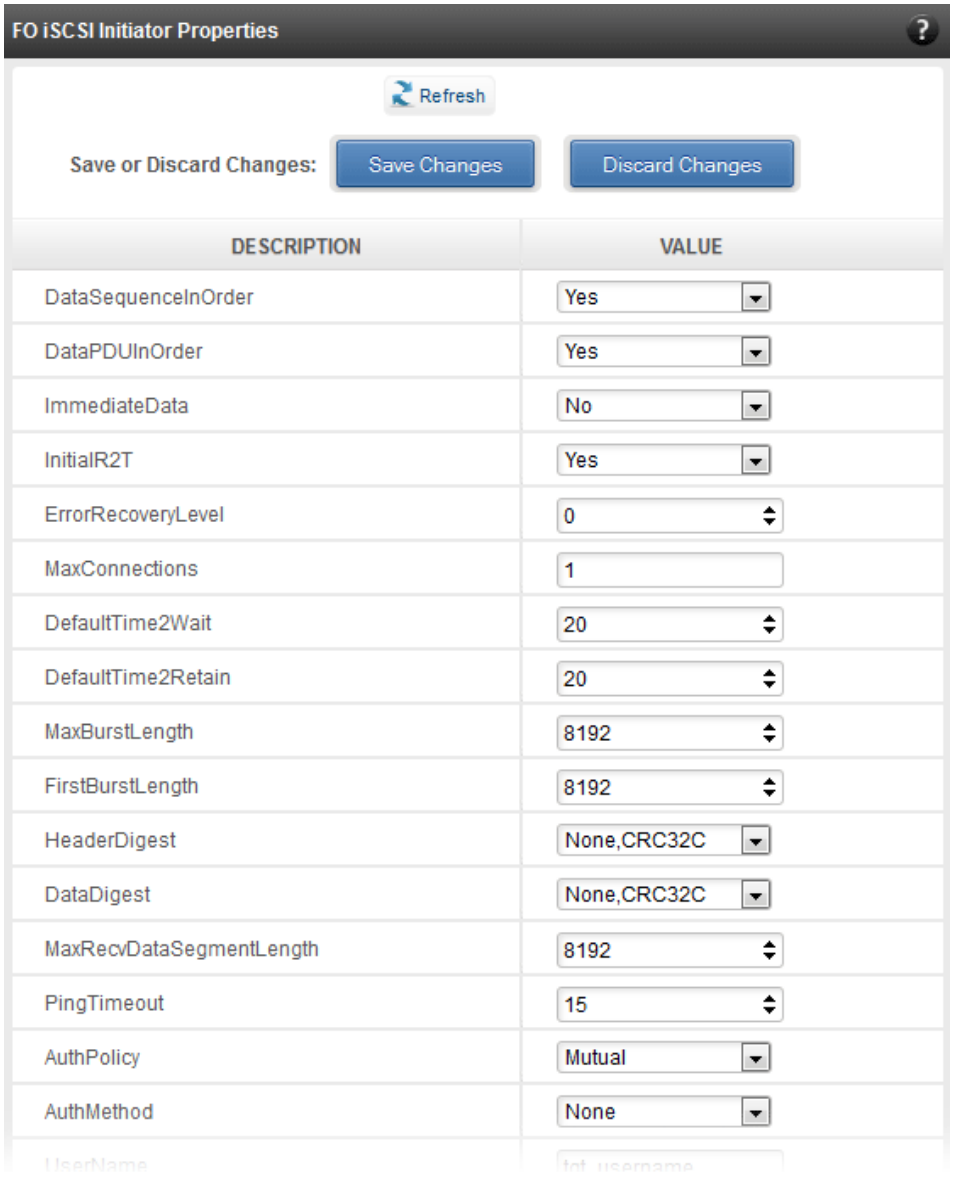

Figure 163 - FO iSCSI Initiator Properties

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.

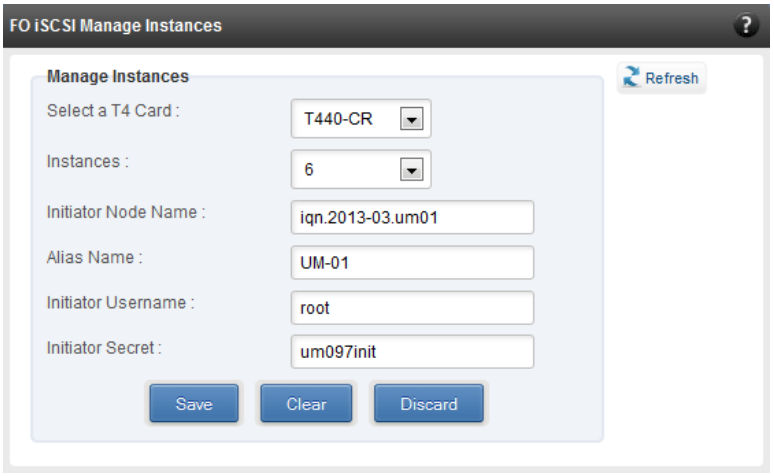

Figure 164 - FO iSCSI Manage Instances module

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.

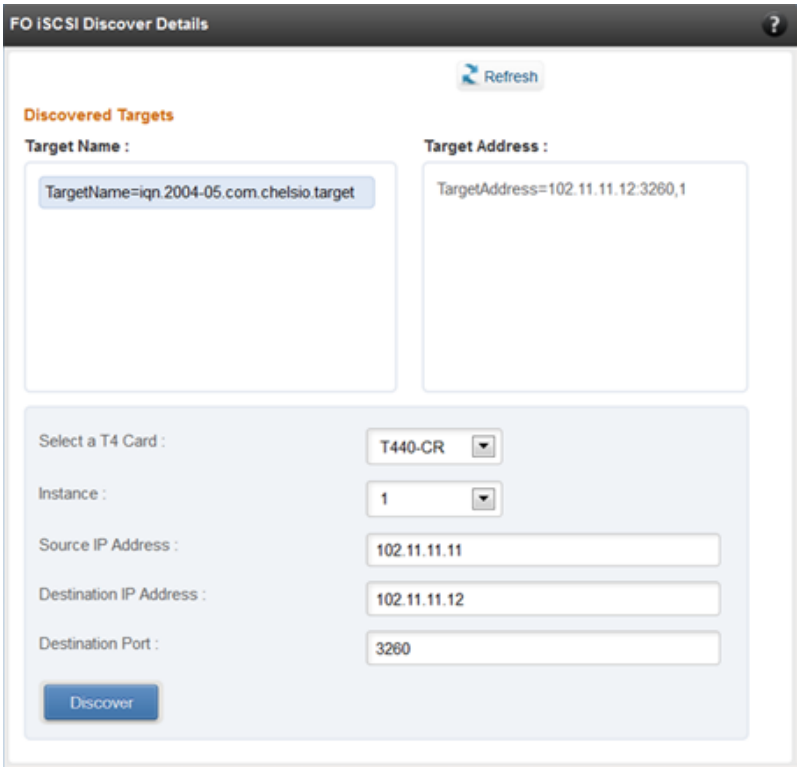

Figure 165 - FO iSCSI Discover Details module

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.

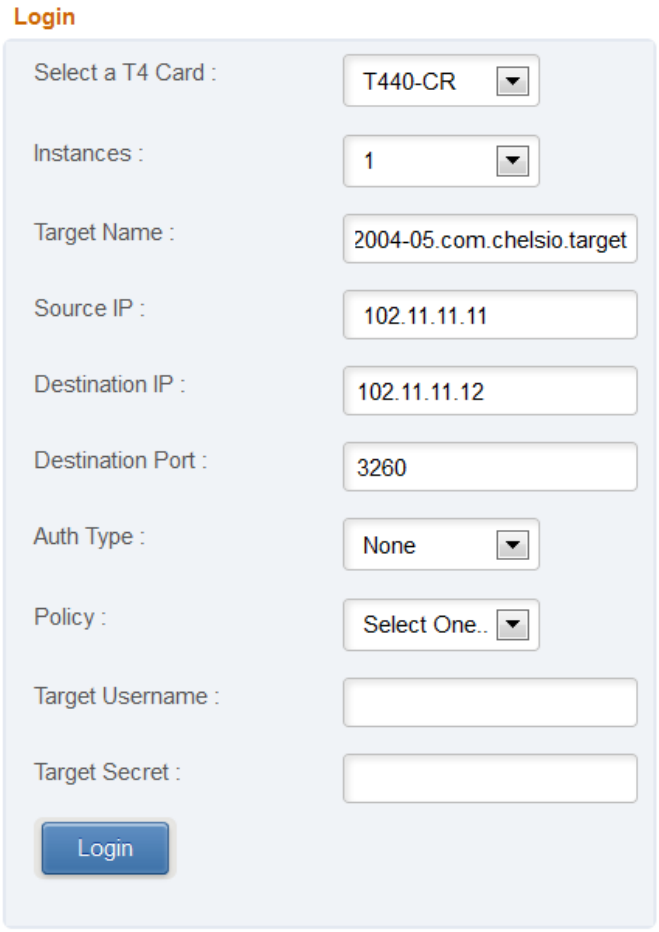

Figure 166 - FO iSCSI Session Details module: Login

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**.

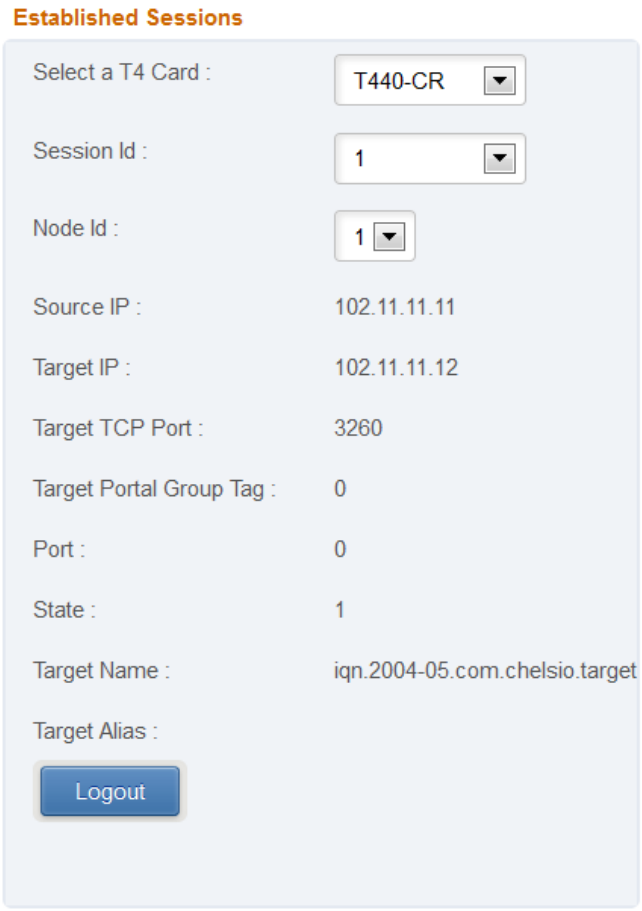

Figure 167 - FO iSCSI Session Details module: Established Sessions

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.

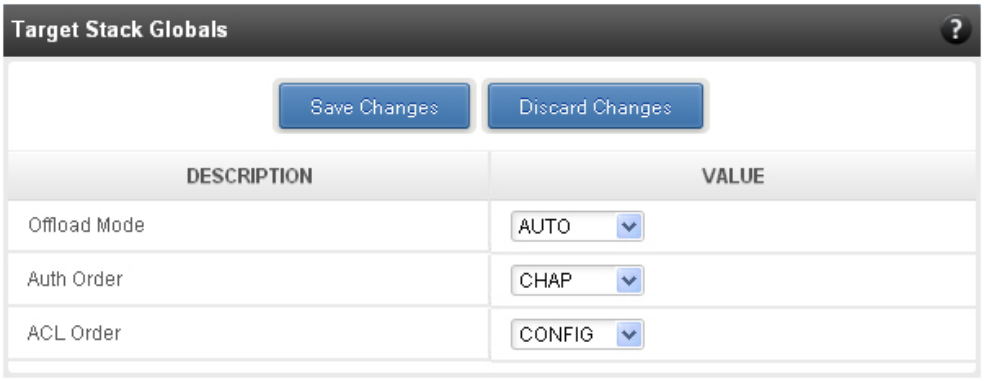

Figure 168 - Target Stack Globals module

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.

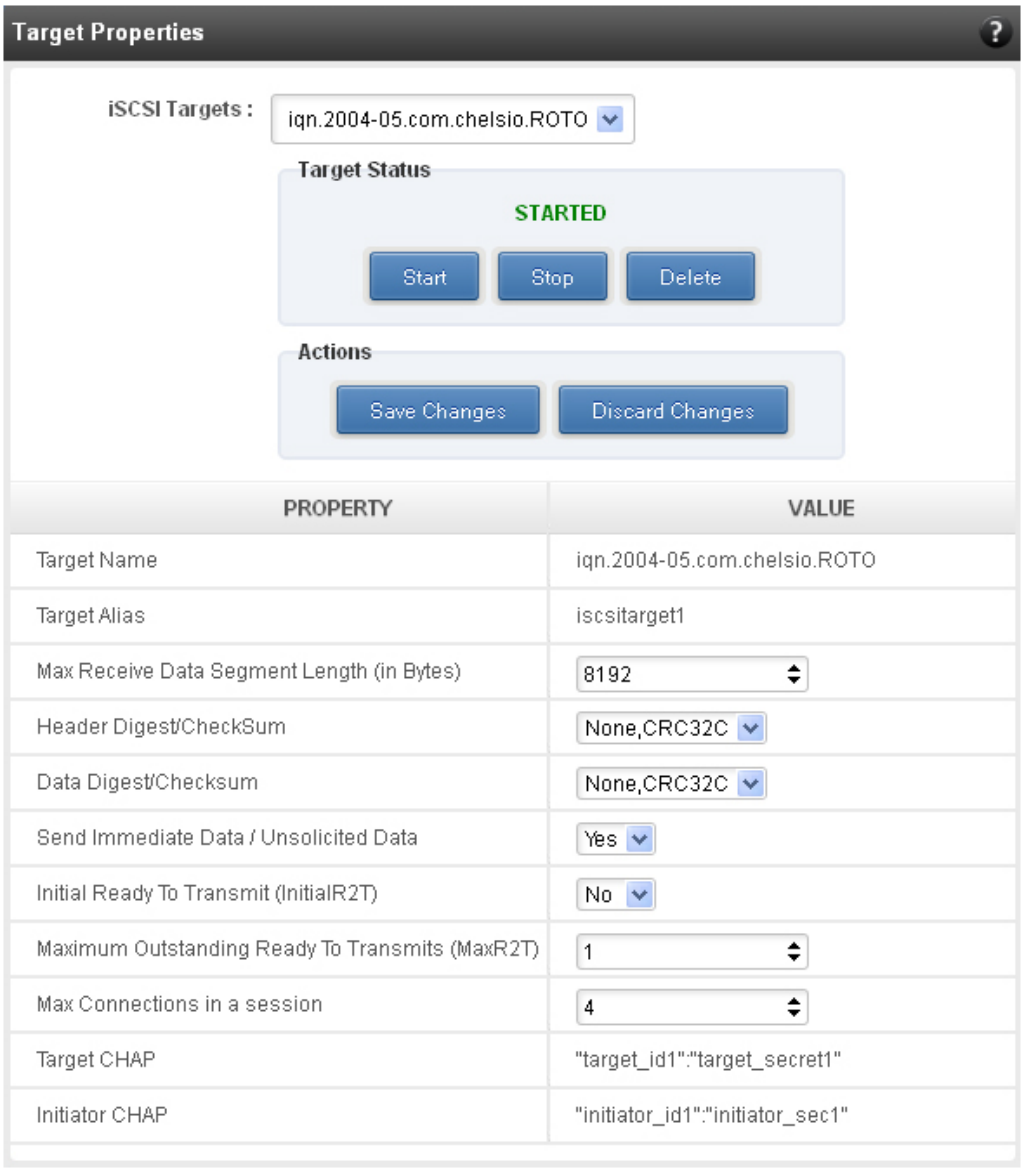

Figure 169 - Target properties module

Session details

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

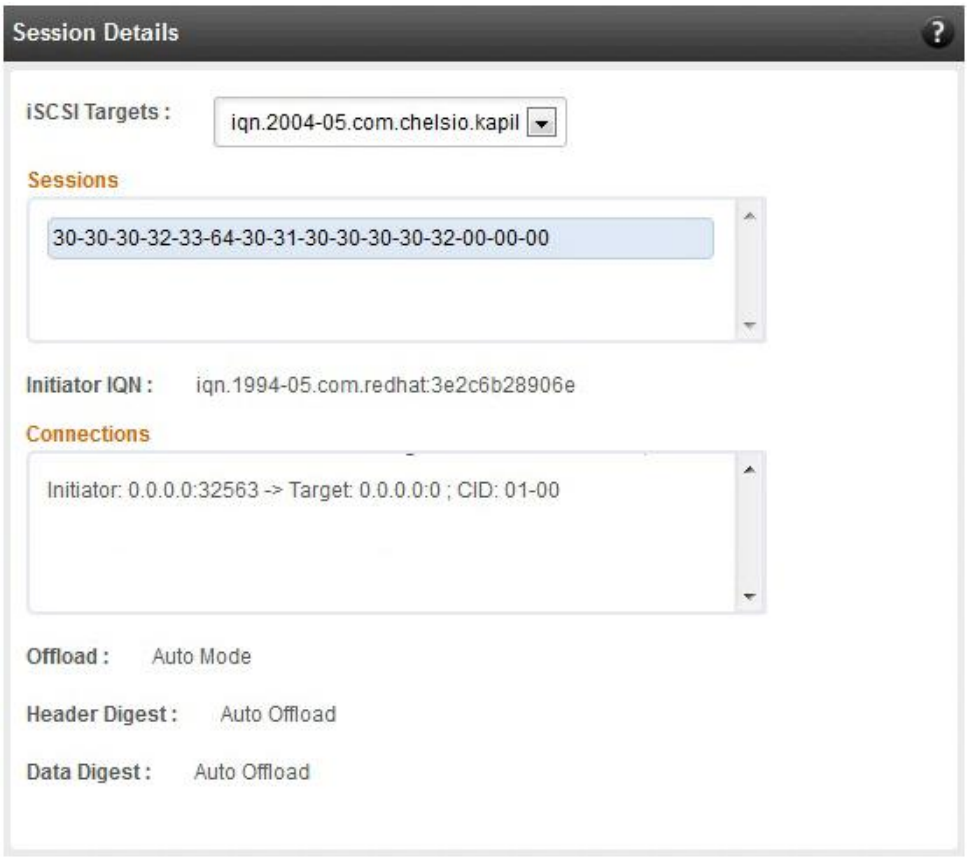

Figure 170 - Session Details module

New Target Creation

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

Figure 171 - 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.

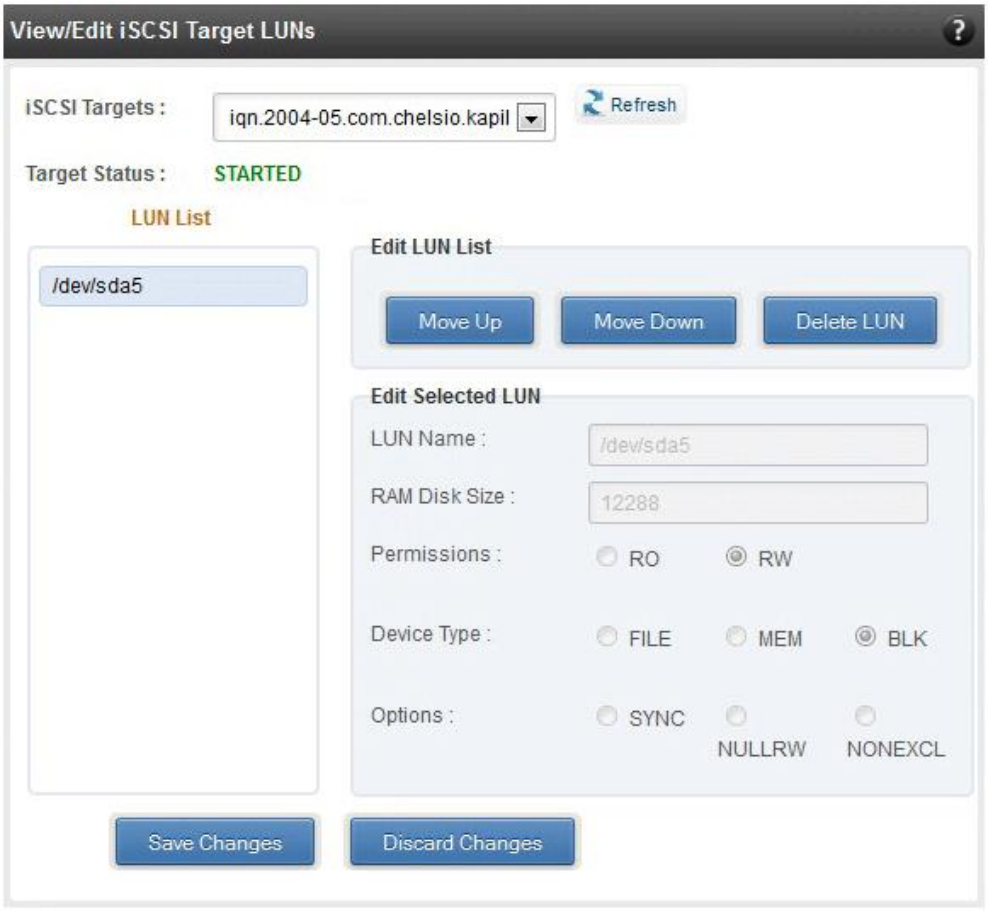

Figure 172 - View/Edit iSCSI Target LUNs module

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.

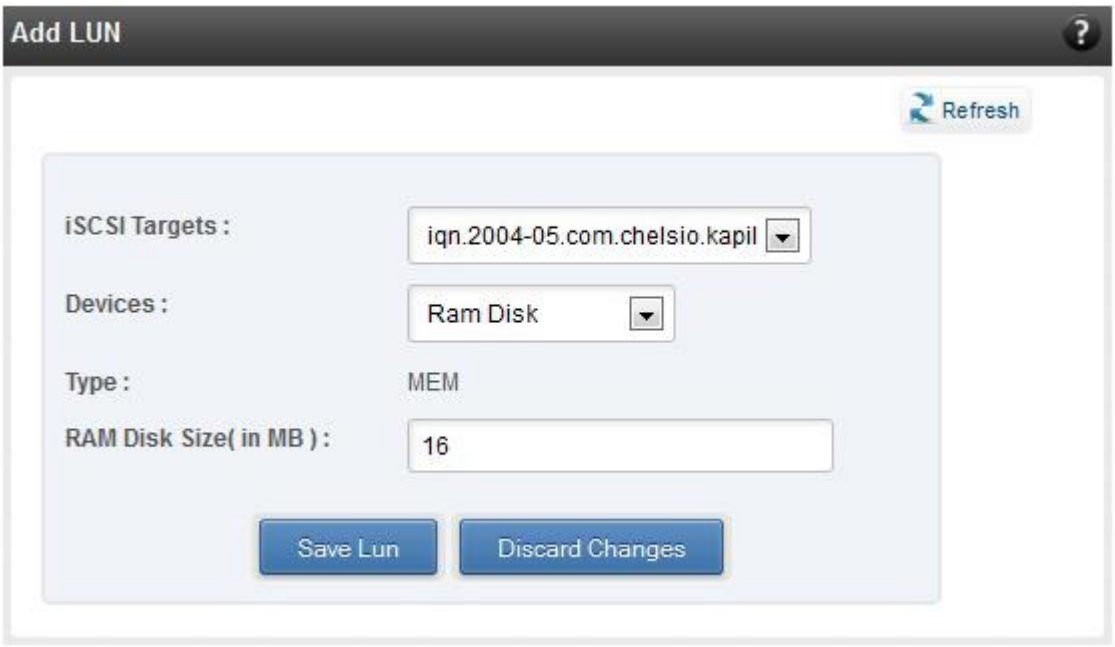

Figure 173 - Adding a new LUN

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.

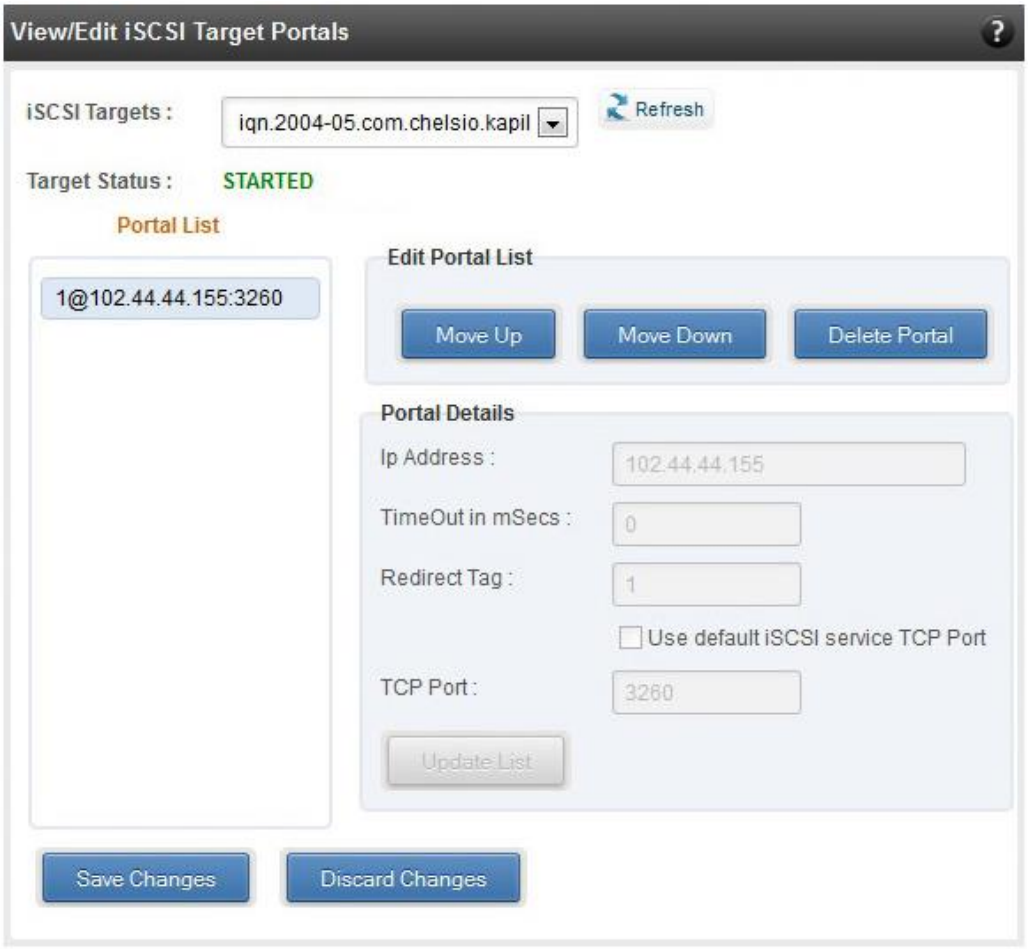

Figure 174 - View/Edit iSCSI Target Portals module

- **Add Portal**

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

Figure 175 - 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.

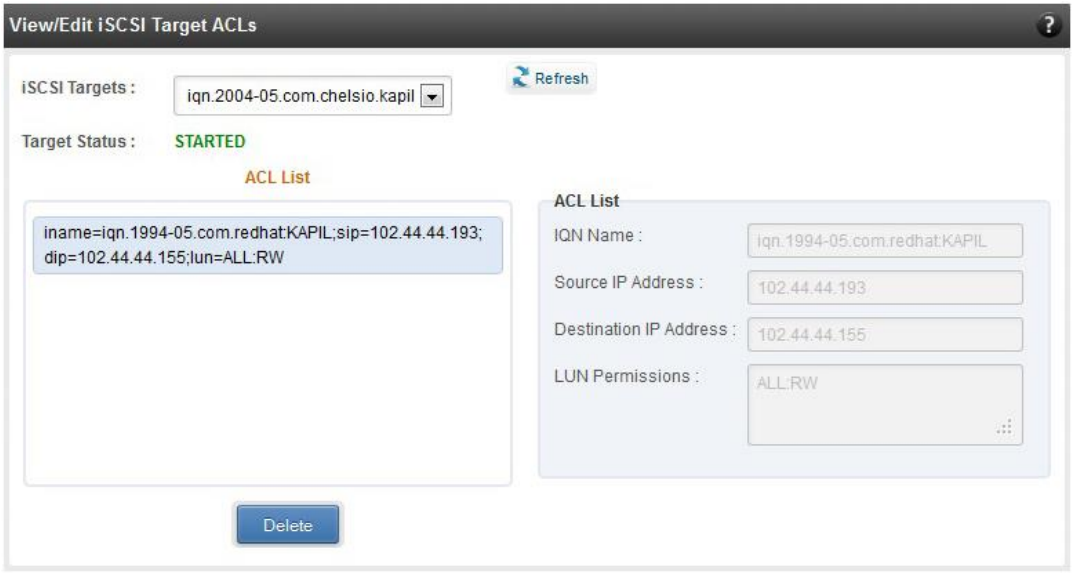

Figure 176 - Target ACL operations module

Add ACL

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

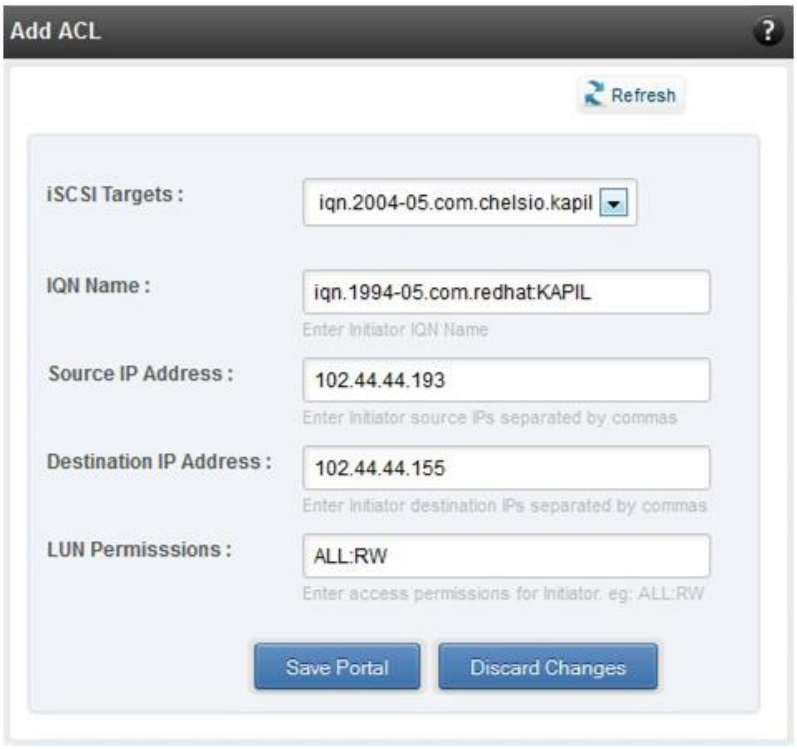

Figure 177 - Adding new ACL

8.8. Hardware Features

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

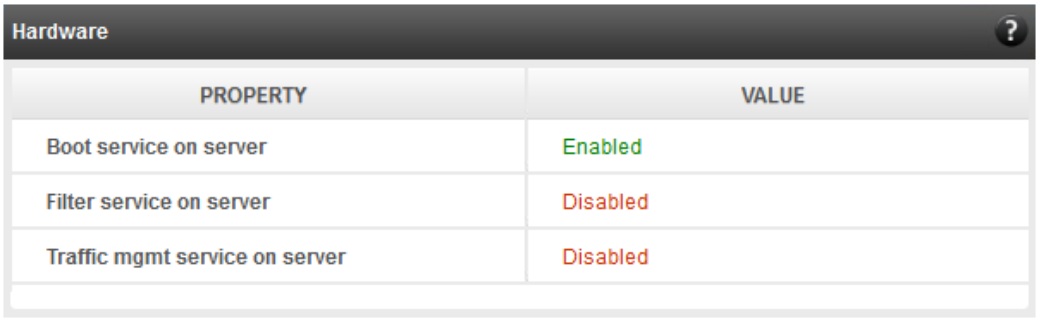

Figure 178 - Hardware module

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.

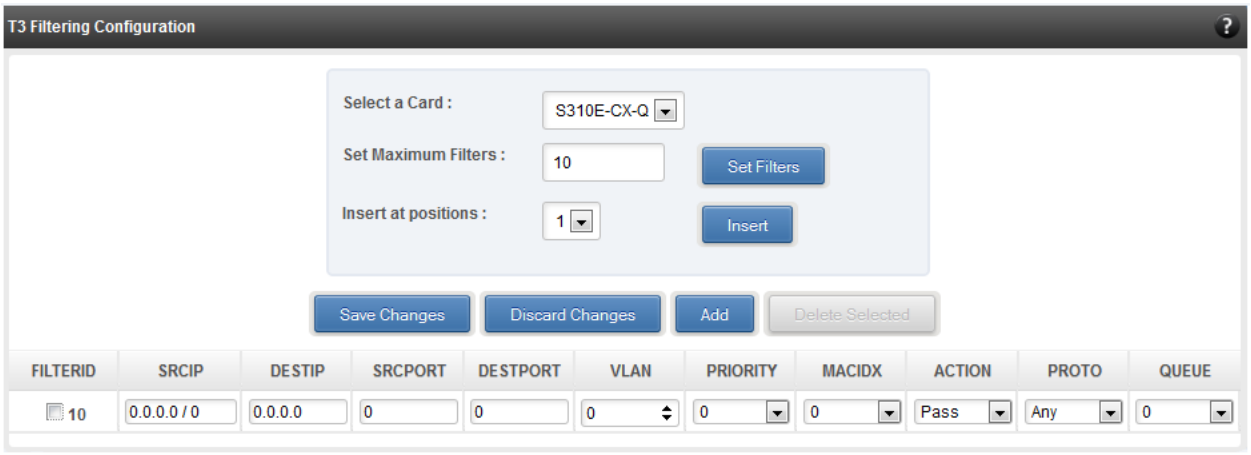

Figure 179 - T3 Filtering Configuration module

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”.

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

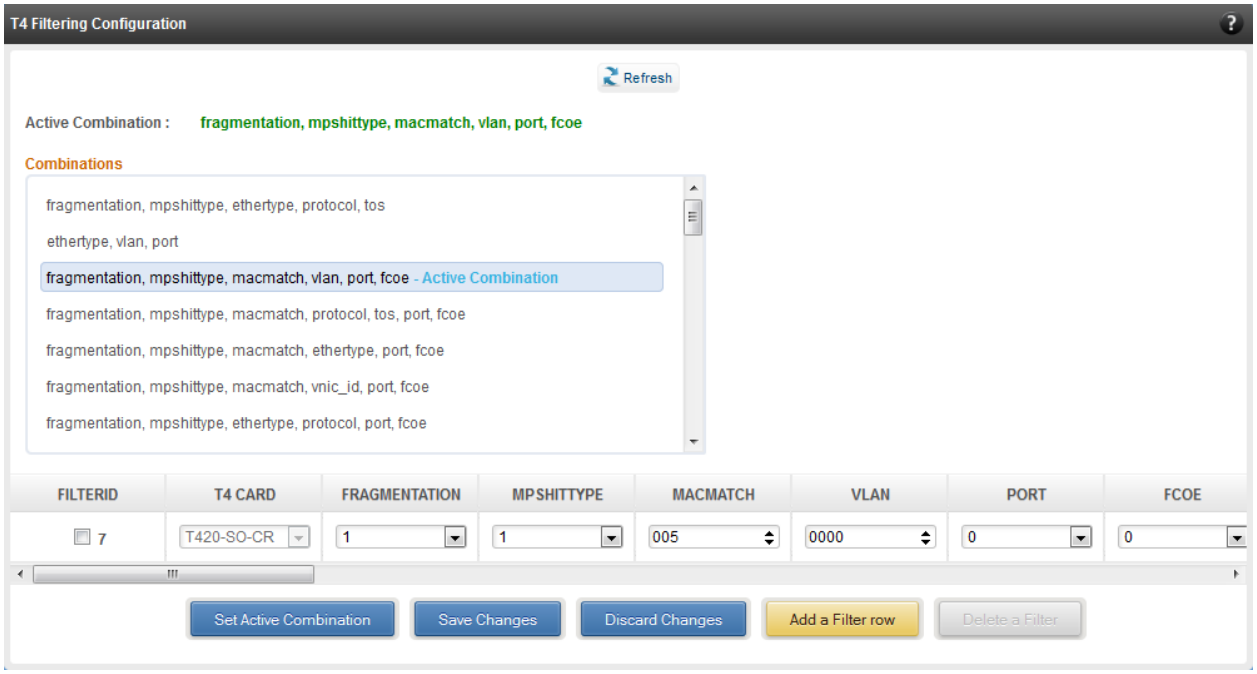

Figure 180 - T4 Filtering Configuration module

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.

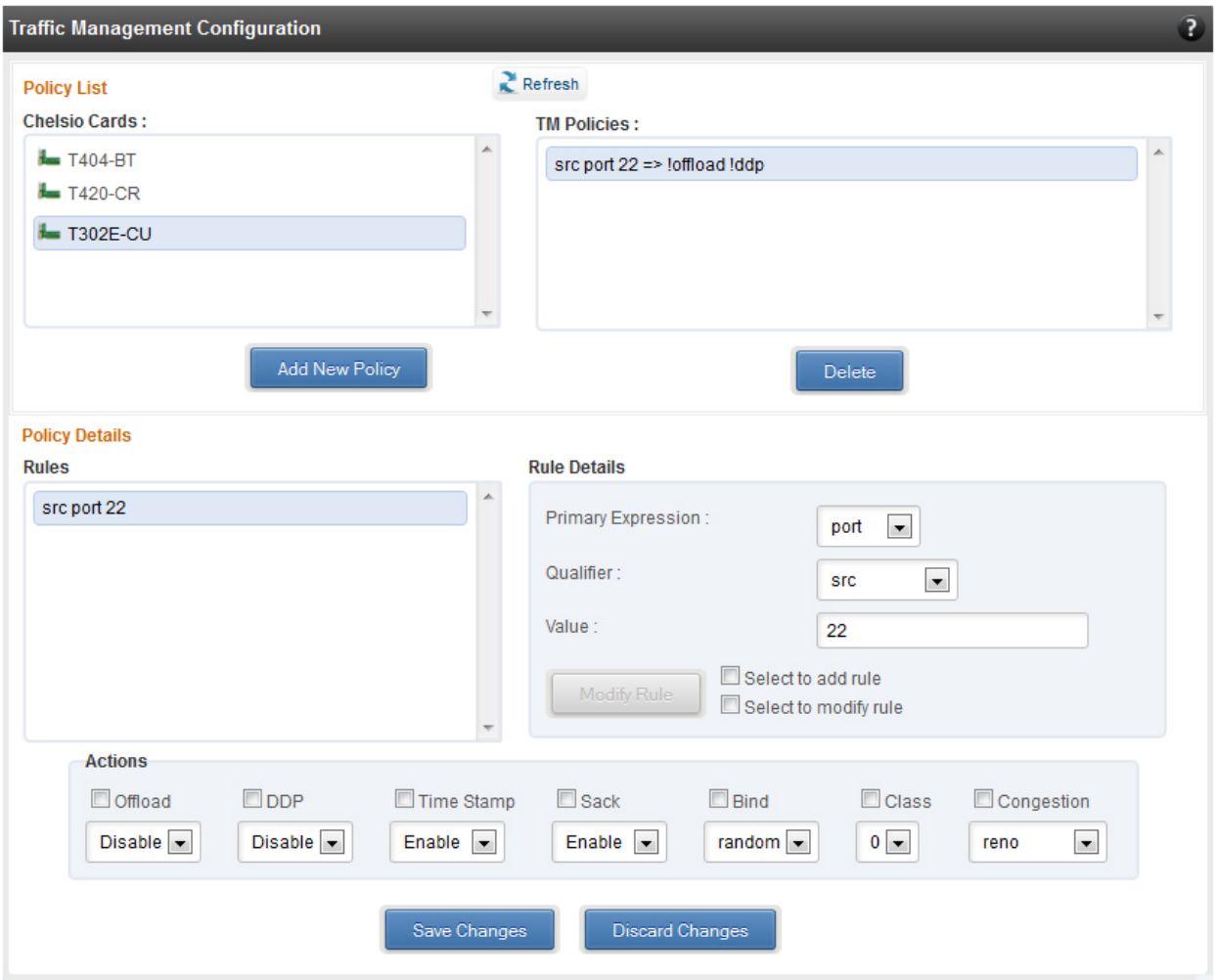

Figure 181 - Traffic Management Configuration module

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.

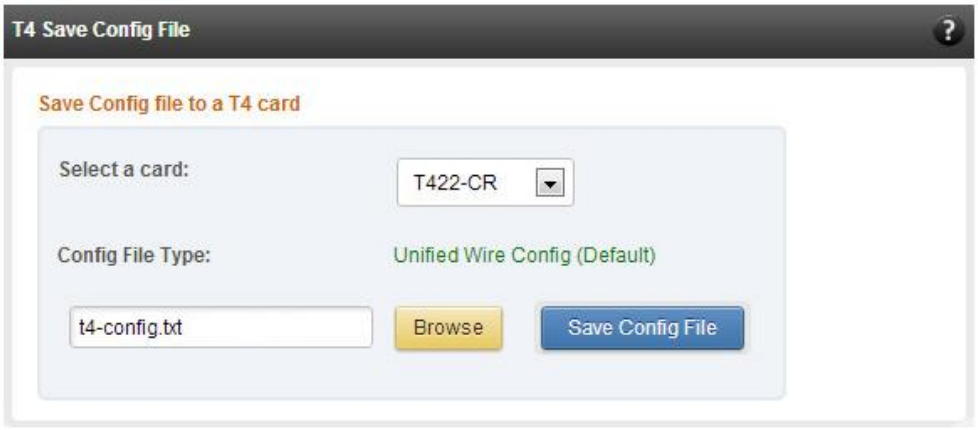

Figure 182 - T4 Save Config File module

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 183 - 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.

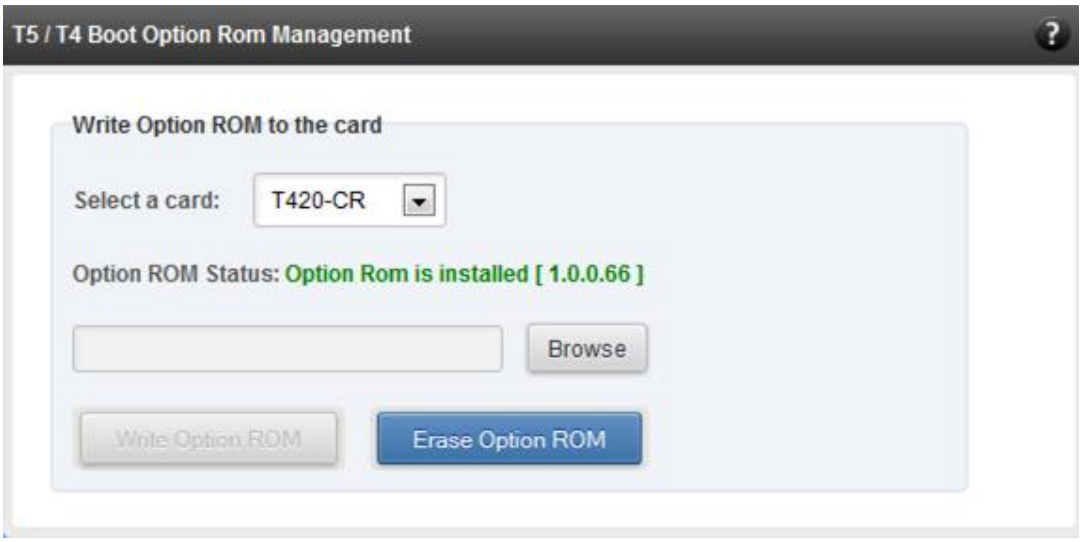

Figure 184 - T5/T4 Option ROM Management module

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.

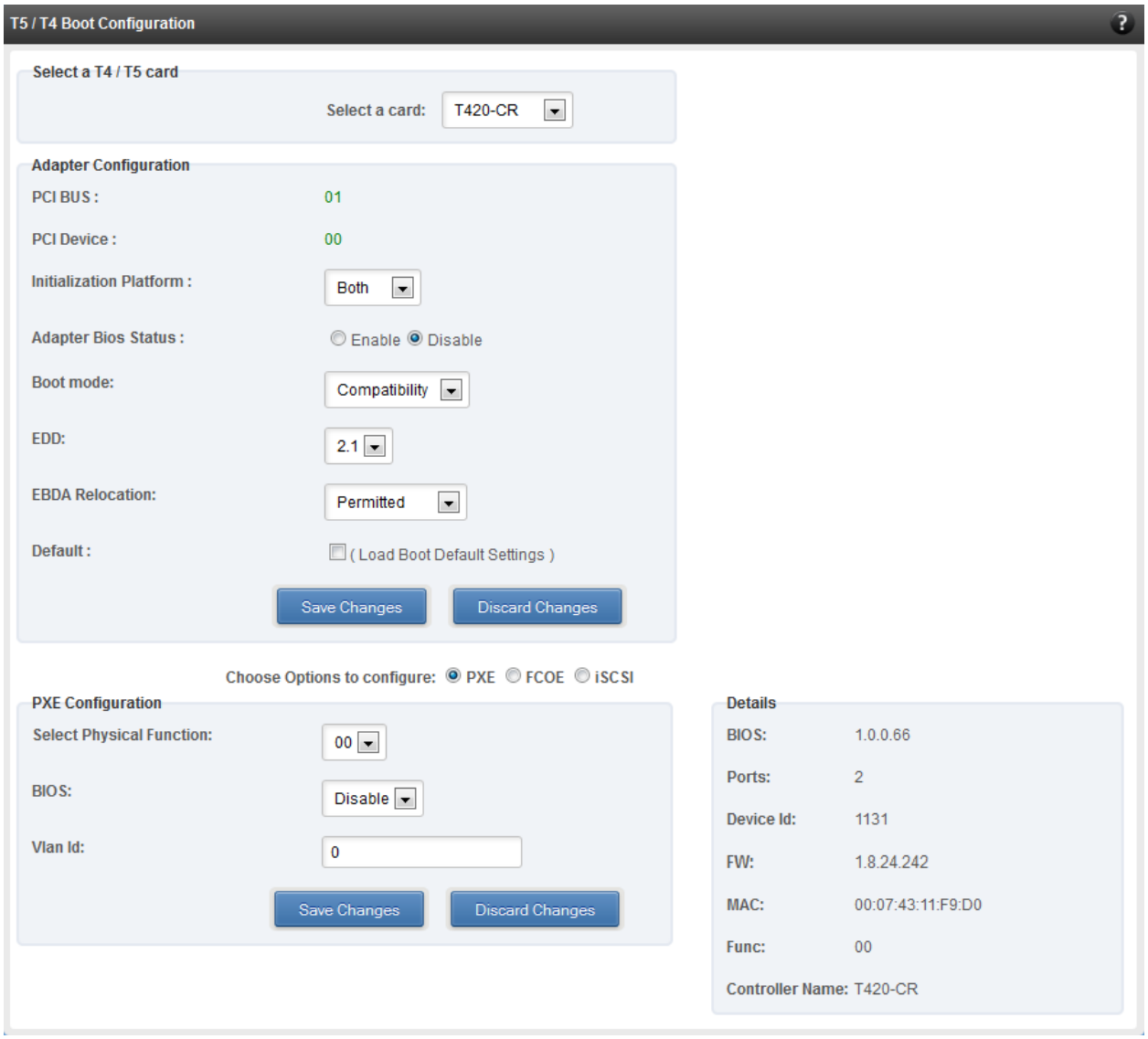

Figure 185 - PXE Boot configuration for T4 CNAs

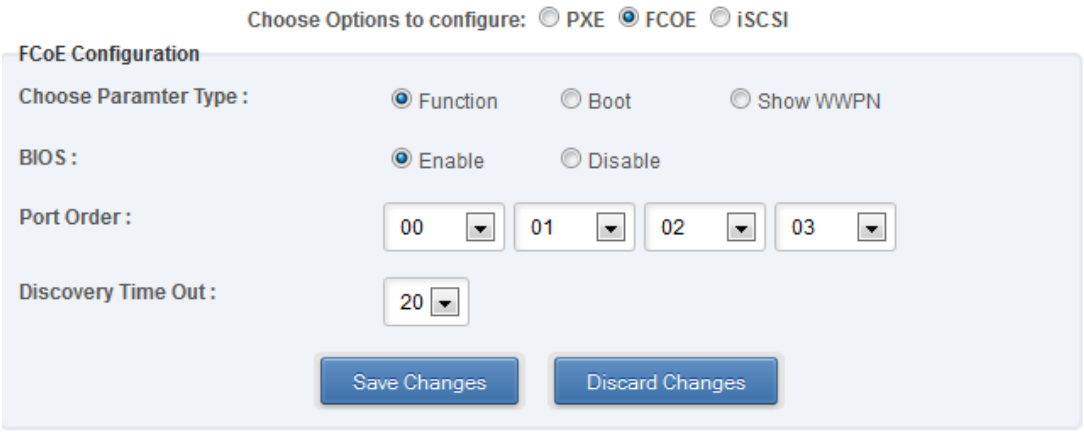

Figure 186 - FCoE Boot configuration for T4 CNAs

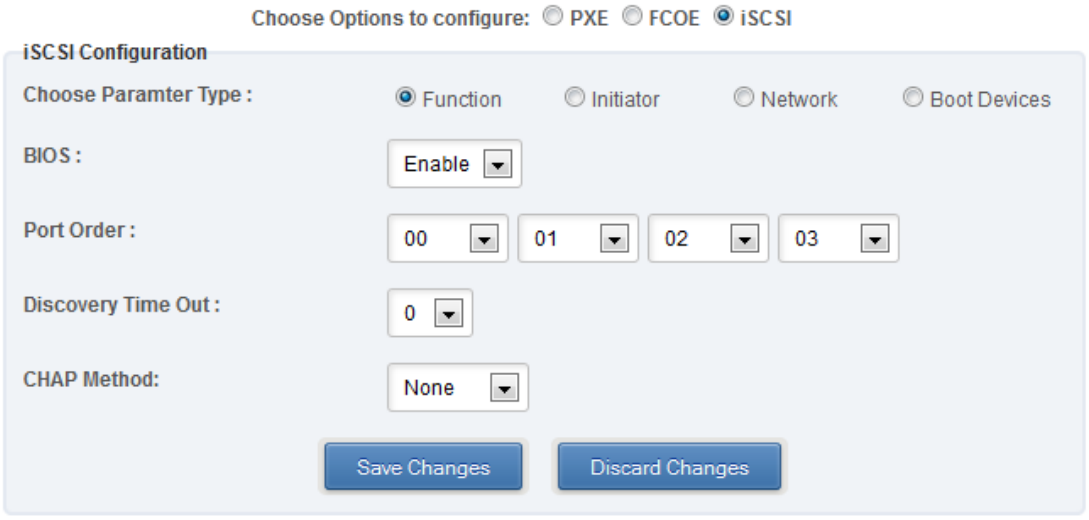

Figure 187 - iSCSI Boot configuration for T4 CNAs

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.

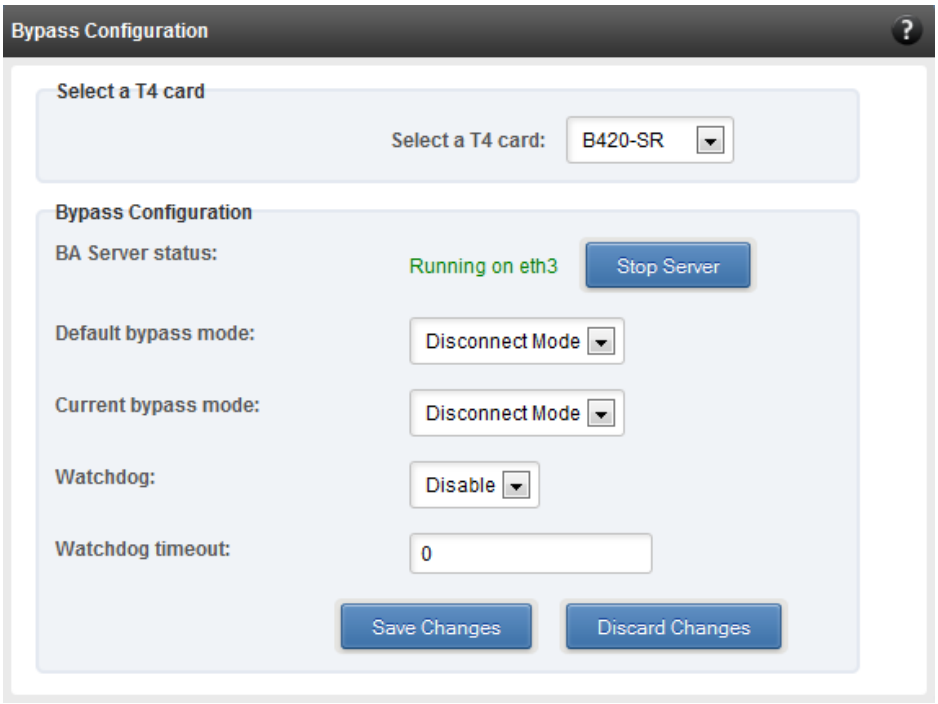

Figure 188 - Bypass Configuration module

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.

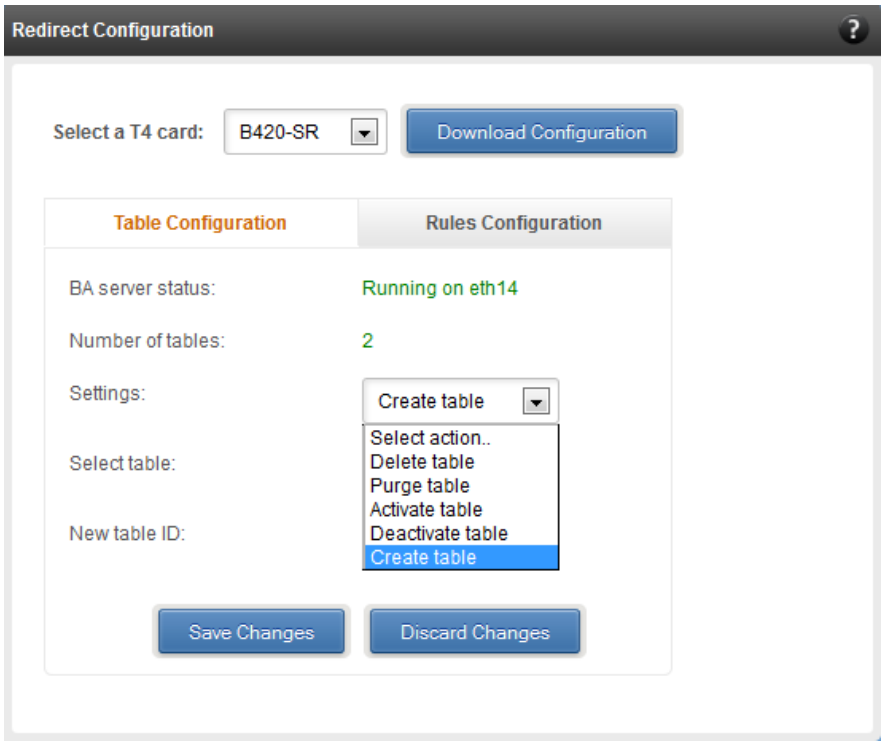

Figure 189 - Redirect Configuration module: Table configuration tab

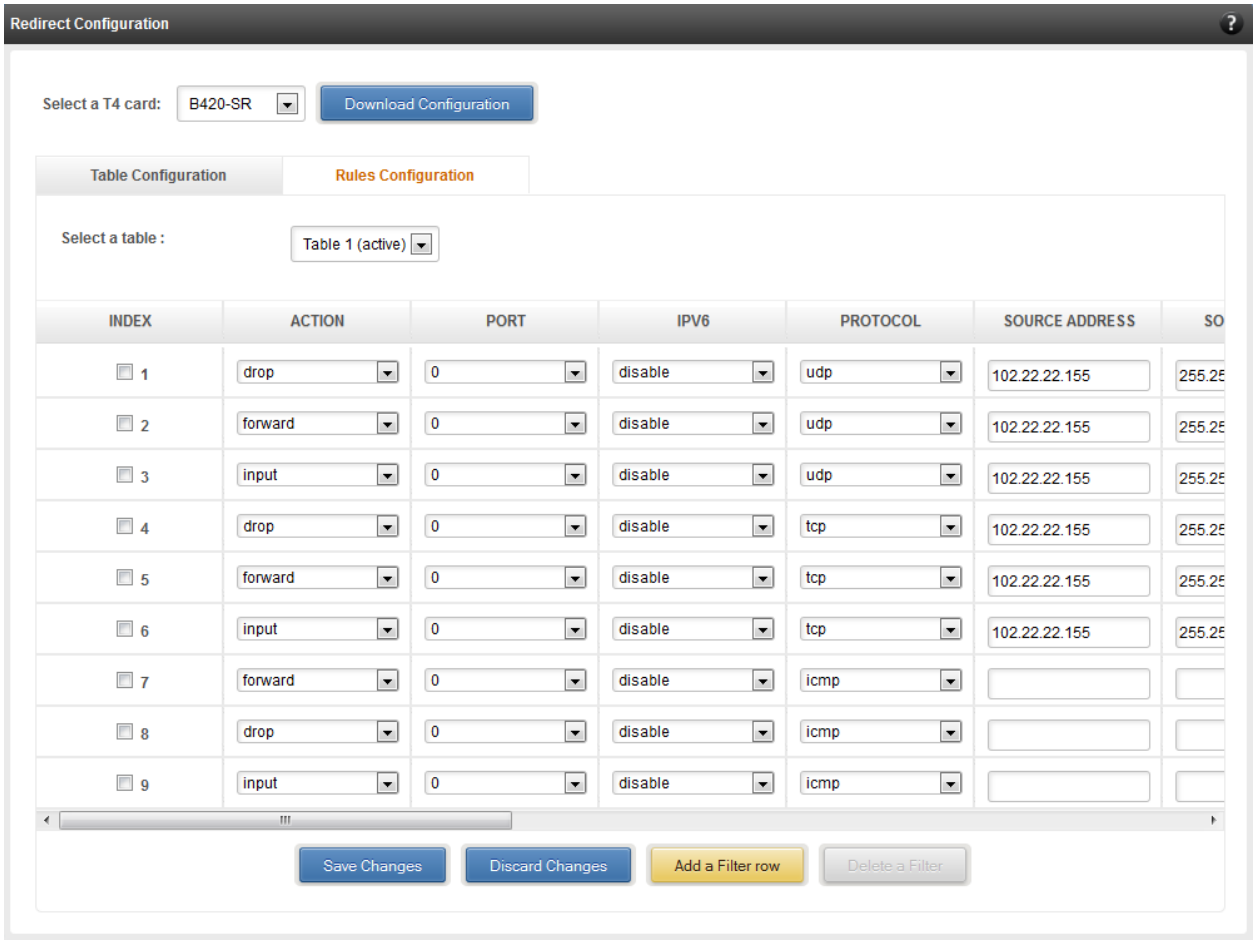

Figure 190 - Redirect Configuration module: Rules configuration tab

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.

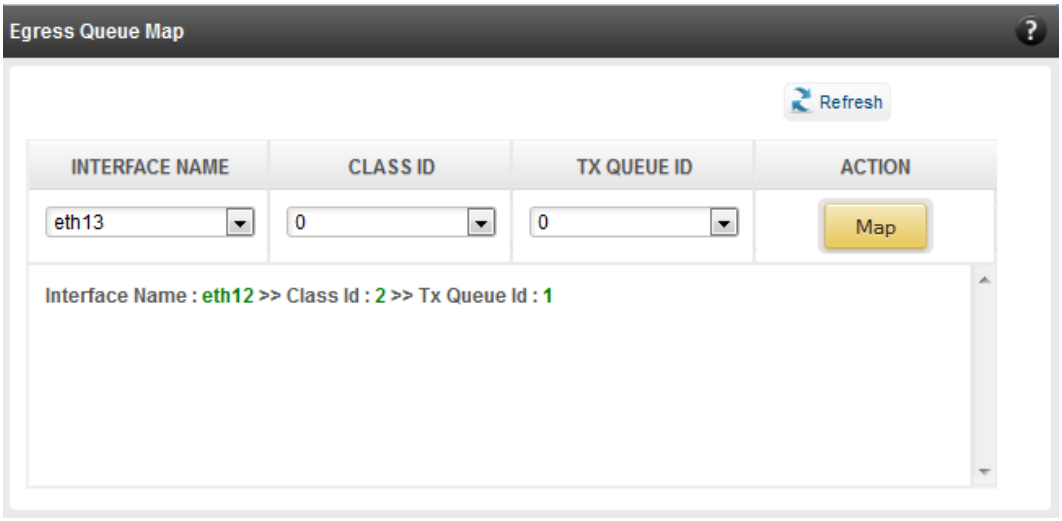

Figure 191 - Egress Queue Map module

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.

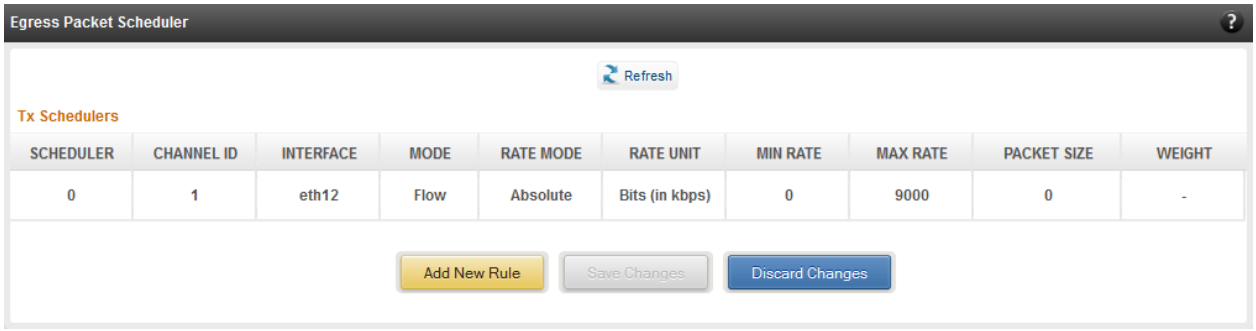

Figure 192 - Egress Packet Scheduler module

VIII. 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>\ChelsioT4\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:

- T520-CR
- T580-CR
- T580-LP-CR
- T580-SO-CR
- T520-LL-CR
- T520-SO-CR
- T540-CR
- T420-CR
- T440-CR
- T422-CR
- T420-LL-CR
- T440-LP-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 193 – '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 194 - '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
```

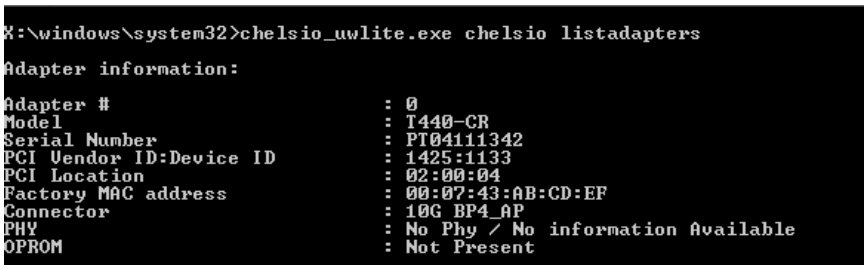

Figure 195 - Listing Chelsio Adapters

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 196 - 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
```

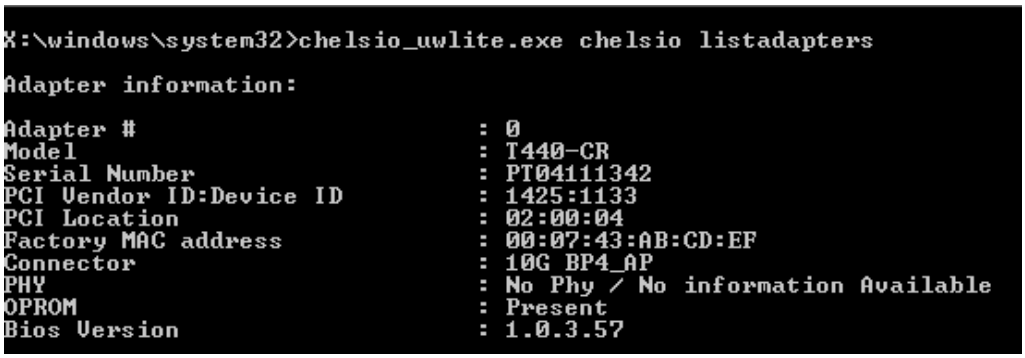

Figure 197 - Listing Chelsio Adapters

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 198 - Erasing option ROM

IX. Appendix

Chelsio End-User License Agreement (EULA)

Installation and use of the 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