

Command Line Installation

Overview	1
Driver Installation.....	2
Network Configuration	3
Creating a Team	4
Creating a VLAN.....	6
Configuring Advanced Driver Features.....	8
Configuring teamed interface advanced properties using slicutil.....	8

Overview

With the advent of the Release 10 driver, Alacritech Network Accelerators can now be installed and configured entirely via the CMD shell. This functionality is most useful for Server Core 2008 and scripted installs. Command line installation and configuration uses the following tools:

- **msiexec.exe:** This is the Microsoft installer.
- **netsh:** Microsoft network shell for manipulating network configuration.
- **reg.exe:** Microsoft tool for registry editing.
- **slicutil.exe:** Alacritech tool for configuring teaming and VLANs.

Documentation for the Microsoft tools involved can be found elsewhere on the net, so we shall not repeat it here. Documentation for the slicutil program can be found on the [Diagnostic Tools page](#).

All of the following examples below assume that an uncompressed release 10 driver is available on the system in question. Whether access is via CDROM, USB device, local drive, or network share is not relevant.

In all of the examples below, the following is true:

Host:	Dell PowerEdge 2950 III
Operating System:	Windows Server 2008 x64
Network interfaces:	onboard Broadcom NetXtreme II (1gb, 2 ports) Alacritech SEN2102ET (1gb, 2 ports) Alacritech SEN3001EF-SR (10gb, 1 port)

The Alacritech driver has been extracted from a zip file and placed in C:\Temp\10GB. The slicutil program has been copied from C:\Temp\10GB\wlh\tools to C:\Temp for convenience.

Driver Installation

In order to install via the command line, you need to know the path to the '.msi' file that is appropriate for your OS. Use the following table to determine this.

Operating System	OS Bits	MSI PATH
Server 2008 Vista (NDIS 6.1)	x64	\wlh\x64\setupx64.msi
	x86	\wlh\x86\setupx86.msi
Server 2003 XP x64 (NDIS 5.2)	x64	\snp\x64\setupx64.msi
	x86	\snp\x86\setupx86.msi

In order to allow full control over the install, all of the release 10 setupxxx.msi files support the following properties:

- **SLIC:** Installs the 1 gigabit miniport driver (slic.sys, slicx86.sys, or slicx64.sys).
- **SXG:** Installs the 10 gigabit miniport driver (sxxg.sys, sxxgx86.sys, or sxxgx64.sys).
- **SLICTEAM:** Installs the intermediate driver for teaming and VLANs (slicteam.sys).
- **ALL:** same as SLIC,SXG,SLICTEAM

SLIC and SXG are the default. If you want to install teaming and VLAN support, you need to add ADDDEFAULT=SLICTEAM or ADDDEFAULT=ALL to the msiexec command line.

EXAMPLE: Installing everything on Server 2008 x64 with no user interaction.

```
C:\Temp> msiexec /i 10GB\wlh\x64\setupx64.msi ADDDEFAULT=ALL /quiet
```

Actually, since some of the release 10 drivers are self-signed, the above command will still require user interaction to accept the self-signed components.

You can verify that the drivers installed successfully with netsh:

```
C:\Temp> netsh int show int
```

Admin State	State	Type	Interface Name
Enabled	Connected	Dedicated	ob1_10net
Enabled	Connected	Dedicated	ob2_cs1
Enabled	Connected	Dedicated	Local Area Connection 2
Enabled	Connected	Dedicated	Local Area Connection 3
Enabled	Connected	Dedicated	Local Area Connection 1

Or with slicutil:

```
C:\Temp> slicutil interface list
```

ID	Type	Connection Name	Member	VLANs
7	SEN2102ET	Local Area Connection 2	No	No
12	SEN2102ET	Local Area Connection 3	No	No

```
9      SEN3001EF      Local Area Connection 1      No      No
```

Network Configuration

Network configuration is standard netsh stuff. The first thing is to change the interface names to something that makes them easier to identify, and that doesn't contain spaces.

```
C:\Temp> netsh int set int name = "Local Area Connection 1" newname = SEN3001
C:\Temp> netsh int set int name = "Local Area Connection 2" newname = SENA
C:\Temp> netsh int set int name = "Local Area Connection 3" newname = SENB
```

```
C:\Temp> netsh int show int
```

Admin State	State	Type	Interface Name
Enabled	Connected	Dedicated	ob1_10net
Enabled	Connected	Dedicated	ob2_csl
Enabled	Connected	Dedicated	SENA
Enabled	Connected	Dedicated	SENB
Enabled	Connected	Dedicated	SEN3001

```
C:\Temp> slicutil interface list
```

ID	Type	Connection Name	Member	VLANs
7	SEN2102ET	SENA	No	No
12	SEN2102ET	SENB	No	No
9	SEN3001EF	SEN3001	No	No

Now assign IP addresses. For this example, we assign an IP address 192.168.52.118, netmask 255.255.255.0 to the SEN3001 interface without a default gateway. If you aren't comfortable with slash '/' notation for netmask, refer to netsh documentation for alternatives.

```
C:\Temp> netsh int ip set address SEN3001 static 192.168.52.118/24
```

```
C:\Temp> ipconfig
```

```
Windows IP Configuration
```

```
...
```

```
Ethernet adapter SEN3001:
```

```

Connection-specific DNS Suffix . . :
Link-local IPv6 Address . . . . . : fe80::c8e:8391:2a2a:7343%19
IPv4 Address. . . . . : 192.168.52.118
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
...
```

Creating a Team

It is recommended that the user familiarize themselves with the teaming section of [the release 10 user guide](#), and the [teaming section of the Server Accelerator FAQ](#) before attempting this procedure.

Now use the slicutil command to create a Hot Standby Failover team from the two ports of the SEN2102ET (Interfaces names: SENA, SENB). To do this, we need to know the interface ID for each interface. This is given in the output of:

```
C:\Temp> slicutil interface list
  ID  Type           Connection Name           Member  VLANs
=====
  7   SEN2102ET      SENA                      No      No
  12  SEN2102ET      SENB                      No      No
  9   SEN3001EF      SEN3001                   No      No
```

Note that this ID is **not** the same as the ldx identifier that is returned by some netsh commands.

Forming the team is pretty straightforward.

```
C:\Temp> slicutil team create memid=7,12 type=hsb
Creating team ... done.

C:\Temp> slicutil interface list
  ID  Type           Connection Name           Member  VLANs
=====
  7   SEN2102ET      SENA                      Yes     No
  12  SEN2102ET      SENB                      Yes     No
  9   SEN3001EF      SEN3001                   No      No
  20  Team           Local Area Connection 2   No      No
```

Note that there is now a 'Yes' in the 'Member' column for interfaces SENA and SENB, and that there is now a new interface of type 'Team' with ID '20'. Had we wanted to for an 802.3ad link aggregation team, we would have used `type=FEC`, and of course we would have had to configure the switch as well.

We rename the new interface and assign it an IP address of 172.17.2.118 with a netmask of 255.255.255.0.

```
C:\Temp> netsh int set int name="Local Area Connection 2"
newname=team_test

C:\Temp> netsh int ip set address team_test static 172.17.2.118/24

C:\Temp> slicutil interface list
  ID  Type           Connection Name           Member  VLANs
=====
  7   SEN2102ET      SENA                      Yes     No
  12  SEN2102ET      SENB                      Yes     No
  9   SEN3001EF      SEN3001                   No      No
  22  Team           team_test                  No      No

C:\Temp> netsh int show int
```

```
Admin State    State          Type           Interface Name
```

```
-----
--
Enabled          Connected      Dedicated      ob1_10net
Enabled          Connected      Dedicated      ob2_csl
Enabled          Connected      Dedicated      SENA
Enabled          Connected      Dedicated      SENB
Enabled          Connected      Dedicated      SEN3001
Enabled          Connected      Dedicated      team_test

C:\Temp> netsh int ipv4 show addresses

Configuration for interface "team_test"
    DHCP enabled:                No
    IP Address:                   172.17.2.118
    Subnet Prefix:                172.17.2.0/24 (mask
255.255.255.0)
    InterfaceMetric:              10

Configuration for interface "SEN3001"
    DHCP enabled:                No
    IP Address:                   192.168.52.218
    Subnet Prefix:                192.168.52.0/24 (mask
255.255.255.0)
    InterfaceMetric:              5

Configuration for interface "ob2_csl"
    DHCP enabled:                No
    IP Address:                   172.16.50.218
    Subnet Prefix:                172.16.48.0/22 (mask
255.255.252.0)
    InterfaceMetric:              10

Configuration for interface "ob1_10net"
    DHCP enabled:                No
    IP Address:                   10.1.50.218
    Subnet Prefix:                10.1.0.0/16 (mask
255.255.0.0)
    InterfaceMetric:              20

Configuration for interface "Loopback Pseudo-Interface 1"
    DHCP enabled:                No
    IP Address:                   127.0.0.1
    Subnet Prefix:                127.0.0.0/8 (mask 255.0.0.0)
    InterfaceMetric:              50
```

You can now examine the team to determine which interfaces it is created from.

```
C:\Temp> slicutil interface show 22
Interface ID:      22
Name:              team_test
Device Name:      Alacritech Accelerator Team #Team1
Type:              Hot standby failover Team
Members:          Alacritech SEN2102ET Accelerator Slot #3 Port A
Team1 [7]
```

```

Alacritech SEN2102ET Accelerator Slot #3 Port B
Team1 [12]
  Properties:  Jumbo Packet:  0 = Disabled
                IPv4 Checksum Offload:  3 = RX & TX Enabled
                TCPv4 Checksum Offload:  3 = RX & TX Enabled
                Receive Side Scaling:  1 = Enabled
                TCP Large Send Offload:  1 = Enabled
  
```

To remove the team, you would do:

```
C:\Temp> slicutil team dissolve teamid=22
```

Creating a VLAN

Tagged VLAN virtual interfaces are created using slicutil as well. It is important to note that once you create a VLAN, the physical interface that the new VLAN is associated with will become unavailable to untagged traffic. The Windows API does not support having a single physical interface pass both tagged and untagged frames. This is unlike switch behavior, where it is quite common for a single port to be a member of both tagged and untagged VLANs. You can create multiple tagged VLAN interfaces on top of a single physical interface,

Starting in the same configuration as the end of the previous section we add a tagged VLAN to SEN3001. The VLAN name is VLAN1, and the VLAN ID is 1. Note that the VLAN name is not the same as the interface name, so after creating the interface we will rename it to vlan_test.

```
C:\Temp> slicutil vlan add parentid=9 vlanid=1 vlanname=VLAN1
Adding test ... done.
```

```
C:\Temp> netsh int show int
```

Admin State	State	Type	Interface Name
Enabled	Connected	Dedicated	ob1_10net
Enabled	Connected	Dedicated	ob2_cs1
Enabled	Connected	Dedicated	SENA
Enabled	Connected	Dedicated	SENB
Enabled	Connected	Dedicated	SEN3001
Enabled	Connected	Dedicated	team_test
Enabled	Connected	Dedicated	Local Area Connection 2

```
C:\Temp> slicutil interface list
```

ID	Type	Connection Name	Member	VLANs
7	SEN2102ET	SENA	Yes	No
12	SEN2102ET	SENB	Yes	No
9	SEN3001EF	SEN3001	No	Yes
19	VLAN	Local Area Connection 2	No	No
22	Team	team_test	No	No

You can see from the "Yes" in the VLANs column that interface ID 9 is a parent physical interface for at least on VLAN virtual interface.

Now to rename the VLAN interface and assign it an IP address of 192.168.60.118/24.

```
C:\Temp> netsh int set int name="Local Area Connection 2"  
newname=vlan_test  
  
C:\Temp> netsh int ip set address vlan_test static 192.168.60.118/24  
  
C:\Temp> netsh int ipv4 show addresses  
  
Configuration for interface "ob1_10net"  
    DHCP enabled:                No  
    IP Address:                   10.1.50.218  
    Subnet Prefix:                10.1.0.0/16 (mask  
255.255.0.0)  
    InterfaceMetric:             20  
  
Configuration for interface "ob2_csl"  
    DHCP enabled:                No  
    IP Address:                   172.16.50.218  
    Subnet Prefix:                172.16.48.0/22 (mask  
255.255.252.0)  
    InterfaceMetric:             10  
  
Configuration for interface "team_test"  
    DHCP enabled:                No  
    IP Address:                   172.16.2.118  
    Subnet Prefix:                172.16.2.0/24 (mask  
255.255.255.0)  
    InterfaceMetric:             10  
  
Configuration for interface "vlan_test"  
    DHCP enabled:                No  
    IP Address:                   192.168.60.118  
    Subnet Prefix:                192.168.60.0/24 (mask  
255.255.255.0)  
    InterfaceMetric:             5  
  
Configuration for interface "Loopback Pseudo-Interface 1"  
    DHCP enabled:                No  
    IP Address:                   127.0.0.1  
    Subnet Prefix:                127.0.0.0/8 (mask 255.0.0.0)  
    InterfaceMetric:             50
```

We can examine vlan_test to see who its parent is, as well as other properties.

```
C:\Temp>slicutil interface show 19  
    Interface ID:    19  
    Name:           vlan_test  
    Device Name:    VLAN1  
    Type:           VLAN  
    VLAN ID:       1  
    VLAN Name:     VLAN1  
    VLAN Parent:   Alacritech SEN3001EF Accelerator Slot #2 Port A  
[9]
```

To remove the VLAN, you would do:

```
C:\Temp> slicutil vlan remove devid=19
```

Configuring Advanced Driver Features

The command line interface for configuring advanced interface properties was not complete at the time this page is was written. The idea is to be able to use slicutil to do all of it, but for now, slicutil will only manipulate some, but not all, advanced interface properties for teamed interfaces. Normal interfaces have to be handled with the reg command, as does changing team properties that are not supported by slicutil. VLAN interfaces have no advanced properties, and instead reflect the advanced settings of the underlying physical interface.

Configuring teamed interface advanced properties using slicutil

Teamed interface advanced properties are manipulated using the syntax:

```
slicutil team modify teamid=<Id> property=value [...property=value...]
```

The following properties are supported:

Property	Description	Possible values	Default Value
Type	Team Type: Either GigabitEtherChannel/802.3ad or Hot Standby Failover	FEC = 802.3ad/GEC hsb = Failover only	FEC = 802.3ad/GEC
JumboPacket	Jumbo Packet	0 = Disabled (1500 MTU) 1 = Enabled (9000 MTU)	0 = Disabled
RSS	Receive Side Scaling	0 = Disabled 1 = Enabled	1 = Enabled
IPCsumIPv4	IPv4 Checksum Offload	0 = Disabled 1 = TX Enabled 2 = RX Enabled 3 = TX&RX Enabled	3 = TX&RX Enabled
TCPCsumIPv4	TCP Checksum Offload (IPv4)	0 = Disabled 1 = TX Enabled 2 = RX Enabled 3 = TX&RX Enabled	3 = TX&RX Enabled
TCPCsumIPv6	TCP Checksum Offload (IPv6)	0 = Disabled 1 = TX Enabled 2 = RX Enabled 3 = TX&RX Enabled	3 = TX&RX Enabled
Lsov1IPv4	Large Send Offload Version 1 (IPv4)	0 = Disabled 1 = Enabled	1 = Enabled

Lsov2IPv4	Large Send Offload Version 2 (IPv4)	0 = Disabled 1 = Enabled	1 = Enabled
Lsov2IPv6	Large Send Offload Version 2 (IPv6)	0 = Disabled 1 = Enabled	1 = Enabled
Property	All properties except teamtype	Default	None

In one of the examples above, we created a Hot Standby team. Starting with that team, we can use slicutil to change the team type to FEC and enable jumbo frames.

```
C:\Temp> slicutil team modify teamid=22 type=FEC jumbopacket=1
Modifying Alacritech Accelerator Team #Team1 ... done.

C:\Temp> slicutil interface show 22
Interface ID:      22
Name:             team_test
Device Name:      Alacritech Accelerator Team #Team1
Type:             802.3AD Team
Members:         Alacritech SEN2102ET Accelerator Slot #3 Port A
Team1 [7]
                 Alacritech SEN2102ET Accelerator Slot #3 Port B
Team1 [12]
                 Jumbo Packet:  1 = Enabled
                 IPv4 Checksum Offload:  3 = RX & TX Enabled
                 TCPv4 Checksum Offload:  3 = RX & TX Enabled
                 Receive Side Scaling:  1 = Enabled
                 TCP Large Send Offload:  1 = Enabled
```

Configuring normal interface advanced properties using reg

Warning: The following procedure requires careful attention to detail. Typographic errors can result in unpredictable behavior.

Configuring driver properties using the reg command is a workaround until a safer simpler method is available through an updated version of slicutil. The basic idea is that you use reg to modify registry values, and then reset the interface (via netsh) to get it to read the new values. For a description of what the registry setting do, see [the driver options page](#).

Because it is easy to make mistakes when messing with the registry in this fashion, we recommend that the first thing you do is set a variable to contain the base registry path.

```
set REGNET=HKLM\SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}
```

The ID value returned by **slicutil interface list** is the same as the registry key under HKLM\SYSTEM\CurrentControlSet\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318} when zero padded to four digits.

So to disable TCP offload (IPv4) on the interface ID 9 (SEN3001), you would first query its current state:

```
C:\Temp> reg query %regnet%\0009 /v *TCPConnectionOffloadIPv4
```

```
HKEY_LOCAL_MACHINE\...\0009
  *TCPConnectionOffloadIPv4    REG_SZ    1
```

End of search: 1 match(es) found.

Next, change the value to 0 (note that the argument to /t below must match the type returned by the reg query above).

```
C:\Temp> reg add %regnet%\0009 /v *TCPConnectionOffloadIPv4 /t REG_SZ
/d 0 /f
The operation completed successfully.
```

Verify that the change was made as expected.

```
C:\Temp> reg query %regnet%\0009 /v *TCPConnectionOffloadIPv4

HKEY_LOCAL_MACHINE\...\0009
  *TCPConnectionOffloadIPv4    REG_SZ    0
```

End of search: 1 match(es) found.

Finally, use netsh to reset the interface. This will force the interface to read the updated registry value.

```
C:\Temp> netsh int set int SEN3001 disable
C:\Temp> netsh int set int SEN3001 enable
```