Intel® Ethernet Adapters and Devices User Guide

Rev. 29.1.2

Contents

Welcome to Intel® Ethernet Adapters and Devices User Guide!

Welcome to the software user guide for Intel® Ethernet adapters and devices. This guide covers software installation, setup procedures, and troubleshooting tips for Intel network adapters, connections, and other devices.

What's New in This Release

- View the Intel® Ethernet Controller Products Release Notes on intel.com for the high-level list of what has changed in this release.
- See the Release Notes page in this user guide for information how to view the release notes for past releases.

Supported Operating Systems and Devices

Refer to the following pages for detailed information:

- Supported Operating Systems
- Supported Hardware

Intended Audience

This document is intended for information technology professionals with a high level of knowledge, experience, and competency in Ethernet networking technology.

Overview

This section covers basic information about the operating systems and hardware supported in this release. It also provides links to additional documentation.

- Supported Operating Systems
- Supported Hardware
- Related Documentation
- Discontinued Support

Supported Operating Systems

The drivers in this release have been tested with the operating systems (OSs) listed below. Additional OSs may function with our drivers but are not tested.

Note:

Not all devices support all operating systems listed. Refer to the Release Notes for detailed OS support information for your device.

Refer to Discontinued Support for information on the last release to support particular versions of an operating system.

Microsoft* Windows Server*, Azure Stack HCI, and Windows*

- Microsoft Windows Server 2022
- Microsoft Windows Server 2019, Version 1903
- Microsoft Windows Server 2016
- Microsoft Azure Stack HCI
- Microsoft Windows 11 version 23H2 (build 22631.2506)
- Microsoft Windows 11 version 22H2 (build 22621)
- Microsoft Windows 10 version 21H2 (build 19044)
- Microsoft Windows 10 RS5, Version 1809 (build 17763)

Note:

- Devices based on the following do not support Microsoft Windows or Windows Server:
  - Intel® Ethernet Connection E822-C
  - Intel® Ethernet Connection E822-L
- Microsoft Windows 32-bit operating systems are only supported on Intel 1GbE Ethernet Adapters.
- Some older Intel Ethernet adapters do not have full software support for the most recent versions of Microsoft Windows. Many older Intel Ethernet adapters have base drivers supplied by Microsoft Windows.

VMware* ESXi*

- VMWare ESXi 8.0
- VMware ESXi 7.0

Please refer to VMware's download site for the latest ESXi drivers for Intel Ethernet devices.

Linux*

- Linux Real Time Kernel 5.x and 4.x (only on Intel Ethernet E810 Series)
- Linux, v2.4 kernel or higher
- Red Hat* Enterprise Linux* (RHEL) 9.4

- Red Hat Enterprise Linux 8.10
- SUSE* Linux Enterprise Server (SLES) 15 SP5
- SUSE Linux Enterprise Server 12 SP5
- Canonical* Ubuntu* 22.04 LTS
- Canonical Ubuntu 20.04 LTS
- Debian* 11

FreeBSD*

- FreeBSD 14.0
- FreeBSD 13.3

Supported Hardware

This page lists the Intel Ethernet devices that are supported in the current release.

For help identifying your network device and finding supported devices, visit https://www.intel.com/support.

Note:

Available features and settings are dependent on your device and operating system. **Not all settings described in this user guide are available on every device/OS combination.**

Refer to the Release Notes and Feature Support Matrix documents for details on supported features and OS combinations.

Intel® Ethernet 800 Series

- Intel® Ethernet Controller E810-C
- Intel® Ethernet Controller E810-XXV
- Intel® Ethernet Connection E822-C
- Intel® Ethernet Connection E822-L
- Intel® Ethernet Connection E823-C
- Intel® Ethernet Connection E823-L

Intel® Ethernet 700 Series

- Intel® Ethernet Controller I710
- Intel® Ethernet Controller X710
- Intel® Ethernet Controller XL710
- Intel® Ethernet Network Connection X722

- Intel® Ethernet Controller XXV710
- Intel® Ethernet Controller V710

Intel® Ethernet 500 Series

- Intel® Ethernet Controller 82599
- Intel® Ethernet Controller X520
- Intel® Ethernet Controller X550
- Intel® Ethernet Controller X552
- Intel® Ethernet Controller X553

Intel® Ethernet 300 Series and Other

- Intel® I210 Gigabit Ethernet Controller
- Intel® I350 Gigabit Ethernet Controller
- Intel® Ethernet Controller I225
- Intel® Ethernet Controller I226

Compatibility Notes

In addition, note the following limitations for Intel Ethernet devices and connections.

Intel Ethernet 800 Series:

- Devices based on the Intel® Ethernet Controller E810-C have an expected total throughput for the entire device of 100 Gb/s in each direction if one 100G cable is connected or if two 100G cables are connected.
- Devices based on the Intel® Ethernet 800 Series do not support RDMA when operating in multiport mode with more than 4 ports.

Intel Ethernet 700 Series:

- Devices based on the Intel® Ethernet Controller XL710 (4x10 GbE, 1x40 GbE, and 2x40 GbE) have an expected total throughput for the entire device of 40 Gb/s in each direction.
- The first port of Intel® Ethernet 700 Series adapters will display the correct branding string. All other ports on the same device will display a generic branding string.
- For an Intel® Ethernet 700 Series adapter to reach its full potential, you must install it in a PCIe Gen3 x8 slot. Installing it in a shorter slot, or a Gen2 or Gen1 slot, will limit the throughput of the adapter.
- Devices based on the Intel® Ethernet Controller X722 do not support the following features:
  - Intel® PROSet for Windows* Device Manager
  - Intel® Advanced Network Services (Intel® ANS) teams or VLANs (LBFO is supported)

Intel Ethernet 500 Series:

- Devices based on the Intel® Ethernet Connection X552 and Intel® Ethernet Connection X553 do not support the following features:
  - Energy Efficient Ethernet (EEE)
  - Intel PROSet for Windows Device Manager
  - Intel ANS teams or VLANs (LBFO is supported)
  - Data Center Bridging (DCB)
  - IPSec Offloading
  - MACSec Offloading
- In addition, SFP+ devices based on the Intel Ethernet Connection X552 and Intel Ethernet Connection X553 do not support the following features:
  - Speed and duplex auto-negotiation
  - Wake on LAN
  - 1000BASE-T SFP Modules

Related Documentation

Release Notes

Please see the Intel® Ethernet Controller Products Release Notes for details about new or removed features, supported operating systems per device family, and known issues and limitations. See the Release Notes page in this user guide for information how to view the release notes for the current or past releases.

Refer to the Feature Support Matrix for information on supported features, cables, media types, operating systems, and more.

Intel Resource and Documentation Center

Visit the Intel Resource and Documentation Center for additional resources, configuration guides, and technical documentation on Intel Ethernet software and devices:

- https://www.intel.com/content/www/us/en/resources-documentation/developer.html

You can find content related to Intel Ethernet in the following collection:

- https://www.intel.com/content/www/us/en/resources-documentation/developer.html?collection=for-hardware/ethernet-products

Note: Some content may require a login.

User Guides for Specific Devices

Some adapters and devices have user guides with detailed configuration and setup information. You can access public versions of these documents in the Intel Resource and Documentation Center at the link above.

Refer to the following documentation for advanced configuration information.

Affected Products	Type of Document + Link
Intel® Ethernet Network Adapter E810-XXV-4T	User guide (PDF)
Intel® Ethernet Network Adapter E810-C-Q2T	User guide (PDF)

Discontinued Support

This page lists the features, devices, or operating systems that were discontinued in a particular release. Subsequent releases will not support the discontinued feature.

Release 28.3

End of support for:

- Microsoft Windows Server 2012 and Windows Server 2012 R2
- Microsoft Azure Stack HCI, version 21H2 and 20H2
- Intel® I354 Gigabit Ethernet Controller and the e1s driver
- Autorun.exe and its associated files

Release 28.2.3 is the last release that includes drivers for these OSs and devices.

Release 28.1

End of support for RSS on Microsoft Windows operating systems for the following devices:

- Intel® Ethernet Connection I217-LM
- Intel® Ethernet Connection I218-LM
- Intel® Ethernet Connection (2) I218-LM
- Intel® Ethernet Connection (3) I218-LM

- Intel® Ethernet Connection I218-LM
- Intel® Ethernet Connection I218-V
- Intel® Ethernet Connection (2) I218-V
- Intel® Ethernet Connection (3) I218-V
- Intel® Ethernet Connection I218-V
- Intel® Ethernet Connection I217-V
- Intel® Ethernet Network Adapter I226-T1
- Intel® Ethernet Network Adapter I226-T1
- Intel® Ethernet Network Adapter I225-T1
- Intel® Ethernet Network Adapter I225-T1
- Intel® Ethernet Controller (2) I225-IT
- Intel® Killer™ E3100X 2.5 Gigabit Ethernet Controller
- Intel® Ethernet Controller (2) I225-LM
- Intel® Ethernet Controller (2) I225-LMvP
- Intel® Ethernet Controller (2) I225-V
- Intel® Ethernet Controller (3) I225-IT
- Intel® Killer™ E3100X 2.5 Gigabit Ethernet Controller
- Intel® Ethernet Controller (3) I225-LM
- Intel® Ethernet Controller (3) I225-LMvP
- Intel® Ethernet Controller (3) I225-V
- Intel® Ethernet Controller I226-IT
- Intel® Killer™ E3100X 2.5 Gigabit Ethernet Controller (3)
- Intel® Ethernet Controller I226-LM
- Intel® Ethernet Controller I226-LMvP
- Intel® Ethernet Controller I226-V
- Intel® Ethernet Controller I225-LM
- Intel® Ethernet Controller I225-V
- Intel® Ethernet Connection (2) I219-LM
- Intel® Ethernet Connection (2) I219-V
- Intel® Ethernet Connection (3) I219-LM
- Intel® Ethernet Connection (4) I219-LM
- Intel® Ethernet Connection (4) I219-V
- Intel® Ethernet Connection (5) I219-LM
- Intel® Ethernet Connection (5) I219-V
- Intel® Ethernet Connection I219-LM
- Intel® Ethernet Connection (10) I219-LM

- Intel® Ethernet Connection (11) I219-LM
- Intel® Ethernet Connection (12) I219-LM
- Intel® Ethernet Connection (13) I219-LM
- Intel® Ethernet Connection (14) I219-LM
- Intel® Ethernet Connection (15) I219-LM
- Intel® Ethernet Connection (16) I219-LM
- Intel® Ethernet Connection (17) I219-LM
- Intel® Ethernet Connection (18) I219-LM
- Intel® Ethernet Connection (19) I219-LM
- Intel® Ethernet Connection (20) I219-LM
- Intel® Ethernet Connection (22) I219-LM
- Intel® Ethernet Connection (23) I219-LM
- Intel® Ethernet Connection (24) I219-LM
- Intel® Ethernet Connection (25) I219-LM
- Intel® Ethernet Connection (26) I219-LM
- Intel® Ethernet Connection (27) I219-LM
- Intel® Ethernet Connection (6) I219-LM
- Intel® Ethernet Connection (7) I219-LM
- Intel® Ethernet Connection (8) I219-LM
- Intel® Ethernet Connection (9) I219-LM
- Intel® Ethernet Connection I219-V
- Intel® Ethernet Connection (10) I219-V
- Intel® Ethernet Connection (11) I219-V
- Intel® Ethernet Connection (12) I219-V
- Intel® Ethernet Connection (13) I219-V
- Intel® Ethernet Connection (14) I219-V
- Intel® Ethernet Connection (16) I219-V
- Intel® Ethernet Connection (17) I219-V
- Intel® Ethernet Connection (18) I219-V
- Intel® Ethernet Connection (19) I219-V
- Intel® Ethernet Connection (20) I219-V
- Intel® Ethernet Connection (22) I219-V
- Intel® Ethernet Connection (23) I219-V
- Intel® Ethernet Connection (24) I219-V
- Intel® Ethernet Connection (25) I219-V

- Intel® Ethernet Connection (26) I219-V
- Intel® Ethernet Connection (27) I219-V
- Intel® Ethernet Connection (6) I219-V
- Intel® Ethernet Connection (7) I219-V
- Intel® Ethernet Connection (8) I219-V
- Intel® Ethernet Connection (9) I219-V
- Intel® Ethernet Connection (3) I218-LM
- Intel® Ethernet Connection (3) I218-V

Release 27.9

Release 27.9 is the last release that supports the following:

- Microsoft Windows 8.1
- Microsoft Windows 10 Version 20H2 (build 19042)
- Microsoft Windows 10 Version 21H1 (build 19043)
- VMWare ESXi 6.5
- VMWare ESXi 6.7
- iw_ixl FreeBSD driver

Release 27.5

Release 27.5 is the last release that includes the Microsoft e1q driver in your download package. Release 27.6 removed the e1q driver from the installation package. This affects the following devices:

- Intel® 82575EB Gigabit Network Connection
- Intel® 82575EB Gigabit Backplane Connection
- Intel® Gigabit VT Quad Port Server Adapter
- Intel® 82575EB Multi-Function Network Device
- Intel® 82574L Gigabit Network Connection
- Intel® 82583V Gigabit Network Connection
- Intel® Gigabit CT Desktop Adapter
- Intel® Gigabit CT2 Desktop Adapter
- Intel® 82576 Gigabit Dual Port Network Connection
- Intel® Gigabit ET Dual Port Server Adapter
- Intel® 82576NS Gigabit Ethernet Controller
- Intel® 82576NS Gigabit Network Connection
- Intel® 82576NS SerDes Gigabit Ethernet Controller
- Intel® Gigabit EF Dual Port Server Adapter

- Intel® 82576 Gigabit Dual Port Server Network Connection
- Intel® Gigabit ET Quad Port Server Adapter
- Intel® Gigabit ET2 Quad Port Server Adapter
- Intel® 82576 Gigabit Dual Port Network Connection
- Intel® Gigabit ET Quad Port Mezzanine Card

Release 27.0

Release 27.0 the last release that supports:

- The Microsoft Windows v1q driver

This driver will no longer be tested or updated. The driver may still be provided in your download package or on your install media for your convenience.

Release 26.7

Release 26.7 is the last release that supports:

- Intel® Ethernet Network Adapter E810-XXV-2 for OCP 2.0

Release 26.4

Release 26.4 is the last release that supports the following:

- SUSE Linux Enterprise Server 11
- Canonical Ubuntu 16.04
- The Microsoft Windows e1q driver and devices. These drivers will no longer be tested or updated. This affects devices based on the following:
  - Intel® 82574L Gigabit Network Connection
  - Intel® Gigabit ET2 Quad Port Server Adapter
  - Intel® 82575EB Gigabit Backplane Connection
  - Intel® 82575EB Gigabit Network Connection
  - Intel® 82576 Gigabit Dual Port Network Connection
  - Intel® 82576 Gigabit Dual Port Server Network Connection
  - Intel® 82576NS Gigabit Ethernet Controller
  - Intel® 82576NS SerDes Gigabit Ethernet Controller
  - Intel® 82583V Gigabit Network Connection
  - Intel® Gigabit CT Desktop Adapter
  - Intel® Gigabit CT2 Desktop Adapter
  - Intel® Gigabit EF Dual Port Server Adapter
  - Intel® Gigabit ET Dual Port Network Connection
  - Intel® Gigabit ET Dual Port Server Adapter

    - Intel® Gigabit ET Quad Port Mezzanine Card
    - Intel® Gigabit ET Quad Port Server Adapter
    - Intel® Gigabit VT Quad Port Server Adapter
- Starting with Release 26.4, the following adapters and devices will no longer be tested or updated. The drivers may still be provided in your download package or on your install media for your convenience.
    - Intel® Ethernet Controller X540-AT2
    - Intel® Ethernet Controller X540-AT1
    - Intel® Ethernet Converged Network Adapter X540-T2
    - Intel® Ethernet Converged Network Adapter X540-T1
    - Intel® X540 Virtual Function

Release 26.3

Release 26.3 is the last release that supports the following:

- Microsoft Windows 10, Version 1803
- Microsoft Windows 10, Version 1903
- Red Hat* Enterprise Linux* (RHEL) 6.x
- The FreeBSD em driver. Maintenance for this driver will be continued by the community.

Release 25.2

Release 25.2 is the last release that supports the following:

- DOS tools and diagnostics
- The 32-bit Microsoft Windows 10 e1d driver. 64-bit Microsoft Windows 10 is still supported. This affects devices based on the following controllers:
    - Intel® Ethernet Connection I217-LM
    - Intel® Ethernet Connection I217-V
    - Intel® Ethernet Connection I218-LM
    - Intel® Ethernet Connection I218-V
    - Intel® Ethernet Connection I219-LM
    - Intel® Ethernet Connection I219-V
- The Linux e1000e driver. Maintenance for this driver will be continued by the community.

Release 25.0

Release 25.0 is the last release to support:

- Microsoft* Windows* 7
- Microsoft* Windows Server* 2008 R2
- Intel® QSFP+ Configuration Utility (QCU)
- IOUtil

Release 24.4

Release 24.4 removed the following from the installation package:

- Support for PRO/100 devices
- DOS drivers
- UEFI driver support for PCI/PCI-X devices
- Support for WinCE
- Microsoft* Windows* 10 RS3 (NDIS65) Universal Drivers. Please use the NDIS68 drivers.
- Support for FCoE

Release 23.5

Starting with Release 23.5, the drivers for the following adapters and devices will no longer be tested or updated. The drivers may still be provided in your download package or on your install media for your convenience.

- Intel® 82566MM Gigabit Network Connection
- Intel® PRO/1000 PB Dual Port Server Connection
- Intel® PRO/1000 EB Network Connection with I/O Acceleration
- Intel® PRO/1000 EB1 Network Connection with I/O Acceleration
- Intel® PRO/1000 EB Backplane Connection with I/O Acceleration
- Intel® 82567V-3 Gigabit Network Connection
- Intel® 82567V-4 Gigabit Network Connection
- Intel® 82567LM-4 Gigabit Network Connection
- Intel® 82567LF Gigabit Network Connection
- Intel® 82567V Gigabit Network Connection
- Intel® 82567LM-2 Gigabit Network Connection
- Intel® 82567LF-2 Gigabit Network Connection
- Intel® 82567V-2 Gigabit Network Connection
- Intel® 82567LM-3 Gigabit Network Connection
- Intel® 82567LF-3 Gigabit Network Connection
- Intel® 82577LM Gigabit Network Connection
- Intel® 82577LC Gigabit Network Connection

- Intel® 82578DM Gigabit Network Connection
- Intel® 82578DC Gigabit Network Connection
- Intel® 82567LM Gigabit Network Connection
- Intel® 82566DM Gigabit Network Connection
- Intel® 82566DC Gigabit Network Connection
- Intel® 82566MC Gigabit Network Connection
- Intel® PRO/1000 PT Dual Port Network Connection
- Intel® PRO/1000 PT Dual Port Server Adapter
- Intel® PRO/1000 PT Dual Port Server Connection
- Intel® PRO/1000 PF Dual Port Server Adapter
- Intel® PRO/1000 PT Network Connection
- Intel® PRO/1000 PT Server Adapter
- Intel® PRO/1000 PF Network Connection
- Intel® PRO/1000 PF Server Adapter
- Intel® PRO/1000 PB Server Connection
- Intel® PRO/1000 PM Network Connection
- Intel® PRO/1000 PL Network Connection
- Intel® PRO/1000 PT Quad Port Server Adapter
- Intel® PRO/1000 PF Quad Port Server Adapter
- Intel® PRO/1000 PT Desktop Adapter
- Intel® PRO/1000 EB1 Backplane Connection with I/O Acceleration
- Intel® PRO/1000 PT Quad Port LP Server Adapter
- Intel® 82566DM-2 Gigabit Network Connection
- Intel® 82562V 10/100 Network Connection
- Intel® 82562V-2 10/100 Network Connection
- Intel® 82562G-2 10/100 Network Connection
- Intel® 82562GT-3 10/100 Network Connection
- Intel® 82562GT-2 10/100 Network Connection
- Intel® 82562GT 10/100 Network Connection
- Intel® 82562G 10/100 Network Connection
- Intel® Gigabit PT Quad Port Server ExpressModule
- Intel® 82566DC-2 Gigabit Network Connection
- Intel® 82598EB 10 Gigabit AF Dual Port Network Connection
- Intel® 10 Gigabit XF SR Dual Port Server Adapter
- Intel® 10 Gigabit XF SR Server Adapter
- Intel® 82598EB 10 Gigabit AF Network Connection

- Intel® 10 Gigabit AT Server Adapter
- Intel® 82598EB 10 Gigabit AT Network Connection
- Intel® 10 Gigabit AT2 Server Adapter
- Intel® 82598EB 10 Gigabit AT2 Network Connection
- Intel® 82598EB 10 Gigabit AT CX4 Network Connection
- Intel® 10 Gigabit SR Dual Port Express Module
- Intel® 10 Gigabit CX4 Dual Port Server Adapter
- Intel® 82598EB 10 Gigabit KX4 Network Connection
- Intel® 10 Gigabit AF DA Dual Port Server Adapter
- Intel® 10 Gigabit XF LR Server Adapter
- Intel® 82598EB 10 Gigabit Dual Port Network Connection

Installation

This chapter covers how to install Intel Ethernet adapters, drivers, and other software.

At a high level, installation involves the steps shown below, which are covered in more detail later in this chapter.

- If you are installing a network adapter, follow this procedure from step 1.
- If you are upgrading the driver software, start with step 5.

1. Make sure that you are installing the latest driver software for your adapter. Visit Intel's support website to download the latest drivers.
2. Review system requirements.
3. Insert the adapter(s) in the computer.
4. Attach the appropriate network cable(s).
5. Install the driver.
6. For Windows systems, install the Intel® PROSet software.

Refer to the subsections below for details.

Note:

If you update the firmware, you must update the driver software to the same family version.

    - PCI Hot Plug Support
- Connect Network Cables
    - Connect Supported SFP+, SFP28, QSFP+, and QSFP28 Modules
    - Connect the Direct Attach Cable
    - Connect the RJ-45 Network Cable
    - Connect the Fiber Optic Network Cable
- Install Drivers and Software
    - Intel Download Center
    - Install via the Complete Driver Pack
    - Install Individual Drivers by OS
    - Install Tools and Other Software

If you have any problems with basic installation, see Troubleshooting.

After completing the basic steps above, you can set up advanced features, if necessary. The available features and the configuration process varies with the device and your operating system. Refer to Device Features for more information; for Linux and FreeBSD drivers, refer to the README file inside the driver tarball.

Hardware Compatibility

Before installing the adapter, check your system for the following:

- The latest BIOS for your system
- One open PCI Express slot (see the specifications of your card for slot compatibility)

Note:

- For devices that support bifurcation, make sure PCI slot bifurcation is enabled in your system's BIOS.
- The Intel® 10 Gigabit AT Server Adapter will only fit into x8 or larger PCI Express slots. Some systems have physical x8 PCI Express slots that actually support lower speeds. Please check your system manual to identify the slot.

Install the Adapter

Select the Correct Slot

One open PCI-Express slot, x4, x8, or x16, depending on your adapter.

Note:

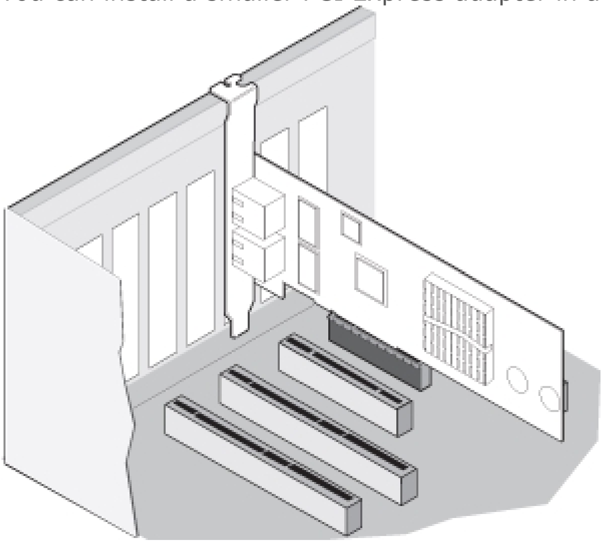
Some systems have physical x8 PCI Express slots that actually only support lower speeds. Please check your system manual to identify the slot.

Insert the Adapter into the Computer

1. If your computer supports PCI Hot Plug, see your computer documentation for special installation instructions.
2. Turn off and unplug your computer. Then remove the cover.

   Note:
   Turn off and unplug the power before removing the computer's cover. Failure to do so could endanger you and may damage the adapter or computer.

3. Remove the cover bracket from an available slot.
4. Insert the adapter, pushing it into the slot until the adapter is firmly seated. You can install a smaller PCI Express adapter in a larger PCI Express slot.



   Note:
   Some PCI Express adapters may have a short connector, making them more fragile than PCI adapters. Excessive force could break the connector. Use caution when pressing the board in the slot.

5. Secure the adapter bracket with a screw, if required.
6. Replace the computer cover and plug in the power cord.
7. Power on the computer.

PCI Hot Plug Support

Most Intel Ethernet Server Adapters are enabled for use in selected servers equipped with Hot Plug support. Exceptions: Intel Gigabit Quad Port Server adapters do not support Hot Plug operations.

If you replace an adapter in a Hot Plug slot, do not place the removed adapter back into the same network until the server has rebooted (unless you return it to the same slot and same team as before). This prevents a conflict in having two of the same Ethernet addresses on the same network.

The system will require a reboot if you:

- Change the primary adapter designator.
- Add a new adapter to an existing team and make the new adapter the primary adapter.
- Remove the primary adapter from the system and replace it with a different type of adapter.

Note:

To replace an existing SLA-teamed adapter in a Hot Plug slot, first unplug the adapter cable. When the adapter is replaced, reconnect the cable.

PCI Hot Plug Support for Microsoft Windows Operating Systems

Intel® network adapters are enabled for use in selected servers equipped with PCI Hot Plug support and running Microsoft Windows operating systems. For more information on setting up and using PCI Hot Plug support in your server, see your hardware and/or Hot Plug support documentation for details. PCI Hot Plug only works when you hot plug an identical Intel network adapter.

Note:

- The MAC address and driver from the removed adapter will be used by the replacement adapter unless you remove the adapter from the team and add it back in. If you do not remove and restore the replacement adapter from the team, and the original adapter is used elsewhere on your network, a MAC address conflict will occur.
- For SLA teams, ensure that the replacement NIC is a member of the team before connecting it to the switch.

Connect Network Cables

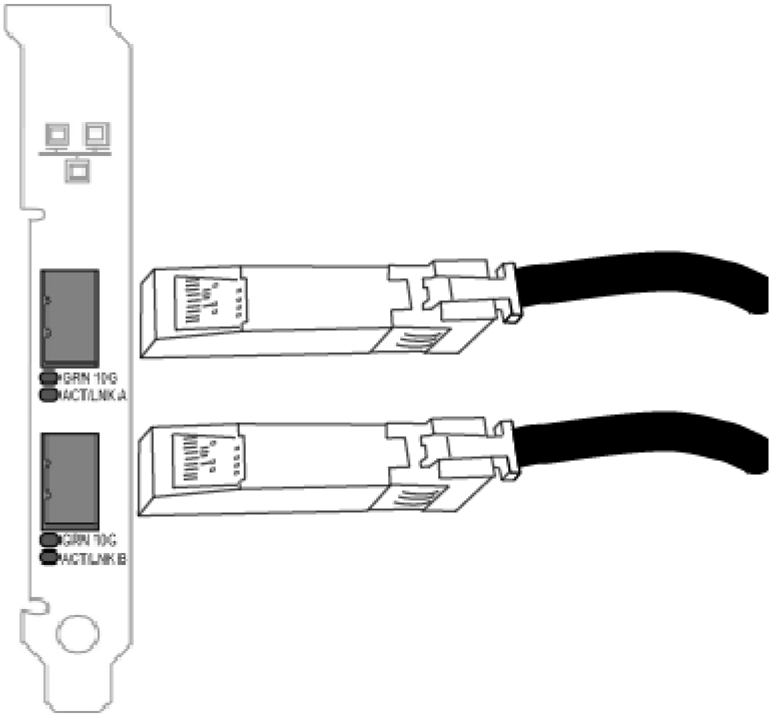
Connect the appropriate network cable, as described in the following sections.

Connect Supported SFP+, SFP28, QSFP+, and QSFP28 Modules

See the Feature Support Matrix for more information on supported media types for your device.

Connect the Direct Attach Cable

Insert the Direct Attach network cable as shown:



The following table shows the types of direct attached cabling you can use.

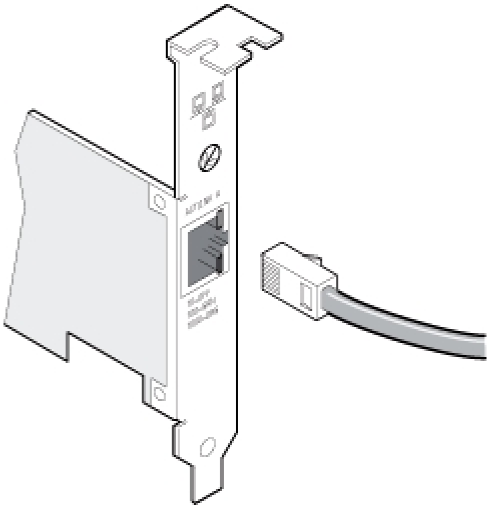
Speed	Cable Type	Max Cable Length
100 Gbps	QSFP28 Direct Attach Cable	5 meters
40 Gbps	SFP+ Direct Attach Cable (Twinaxial)	7 meters
25 Gbps	SFP28 Direct Attach Cable (Twinaxial)	5 meters
10 Gbps	SFP+ Direct Attach Cable (Twinaxial)	7 meters

Note:

For optimal performance with 25 Gbps SFP28 cables, you must use CA-25G-L with RS-FEC and 25GBASE-C.

Connect the RJ-45 Network Cable

Connect the RJ-45 network cable as shown:



The following table shows the maximum lengths for each cable type at a given transmission speed.

Speed	Category 5	Category 6	Category 6a	Category 7
1 Gbps	100m	100m	100m	100m
10 Gbps	NA	55m	100m	100m
25 Gbps	NA	NA	NA	50m
40 Gbps	NA	NA	NA	50m

Note:

If using less than 4-pair cabling, you must manually configure the speed and duplex setting of the adapter and the link partner. In addition, with 2- and 3-pair cabling, the adapter can only achieve speeds of up to 100Mbps.

Note:

For the Intel® 10 Gigabit AT Server Adapter, to ensure compliance with CISPR 24 and the EU’s EN55024, this product should be used only with Category 6a shielded cables that are properly terminated according to the recommendations in EN50174-2.

In all cases:

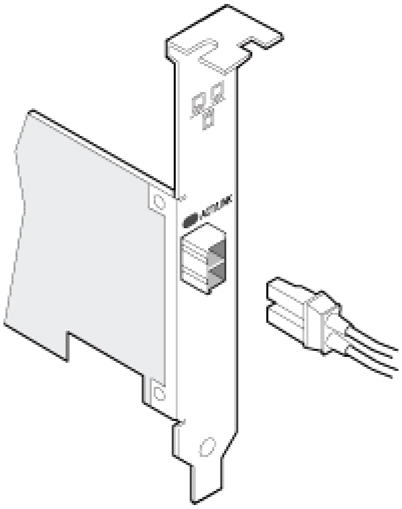
- The adapter must be connected to a compatible link partner, preferably set to auto-negotiate speed and duplex for Intel gigabit adapters.
- Intel Gigabit and 10 Gigabit Server Adapters using copper connections automatically accommodate either MDI or MDI-X connections. The auto-MDI-X feature of Intel gigabit copper adapters allows you to directly connect two adapters without using a cross-over cable.

Connect the Fiber Optic Network Cable

Note:

The fiber optic ports contain a Class 1 laser device. When the ports are disconnected, always cover them with the provided plug. If an abnormal fault occurs, skin or eye damage may result if in close proximity to the exposed ports.

Remove and save the fiber optic connector cover. Insert a fiber optic cable into the ports on the network adapter bracket as shown:



Most connectors and ports are keyed for proper orientation. If the cable you are using is not keyed, check to be sure the connector is oriented properly (transmit port connected to receive port on the link partner, and vice versa).

The adapter must be connected to a compatible link partner operating at the same laser wavelength as the adapter.

Conversion cables to other connector types (such as SC-to-LC) may be used if the cabling matches the optical specifications of the adapter, including length limitations.

The following table shows the connection requirements for fiber optic cables.

Device	Laser Wavelength	Connector Type	Cable Type	Max Cable Length
Intel Ethernet LR Server Adapters	1310 nanometer (not visible)	LC	Single-mode fiber with 9.0µm core diameter	10 km
Intel Ethernet 10 Gigabit SR Server Adapters	850 nanometer (not visible)	LC or SC	• Multi-mode fiber with 62.5µm core diameter • Multi-mode fiber with 50µm core diameter	• 33 m • 300 m
Intel Ethernet Gigabit SR Server Adapters	850 nanometer (not visible)	LC or SC	• Multi-mode fiber with 62.5µm core diameter • Multi-mode fiber with 50µm core diameter	• 275 m • 550 m

Install Drivers and Software

There are multiple ways to download drivers and tools for Intel Ethernet software releases:

- **Download and install the complete driver pack.**
- **Download and install individual drivers.** Exact steps will vary by operating system.

Intel Download Center

You can download all publicly available software from the Intel Download Center at https://www.intel.com/content/www/us/en/download-center/home.html.

Install via the Complete Driver Pack

Downloading the complete driver pack from the Intel Download Center will include drivers for Microsoft Windows, Linux, and FreeBSD, but it is a very large download. We recommend downloading smaller files for your operating system if you don't need software for every OS.

The Intel® Ethernet Adapter Complete Driver Pack is available at https://www.intel.com/content/www/us/en/download/15084/intel-ethernet-adapter-complete-driver-pack.html.

Install Individual Drivers by OS

See the following subsections for details.

- Install Linux Drivers
    - Locations for Linux Driver Files
    - Manually Build the Linux Driver from Source
    - Build a Binary RPM Package of the Linux Driver
    - Install Linux Drivers via Prebuilt RPM Packages
    - Linux Secure Boot Mode
- Install FreeBSD Drivers
- Install VMware Drivers
- Install Windows Drivers
    - Before Installing Drivers
    - Install via the Complete Driver Pack
    - Install Only Windows Drivers
    - Install Base Drivers via the Command Line
    - Command Line Syntax for SaveRestore.ps1
    - Examples for SaveRestore.ps1

Install Tools and Other Software

Intel provides a number of tools and applications that allow you to configure or debug the Intel Ethernet devices in your system.

See Tools & Apps for more information about available tools and how to install them.

Tools may be available as individual downloads from the Intel Download Center, or they are included as part of the complete driver pack. Publicly available tools are located in the `APPS` subfolder of the complete driver pack.

Install Linux Drivers

The instructions below apply to the following device series and drivers:

Device Series	Driver Name
Intel Adapter Virtual Functions	iavf
Intel Ethernet 800 Series	ice
Intel Ethernet 700 Series	i40e
Intel Ethernet 500 Series	ixgbe
Intel Ethernet 300 Series	igb

You have several options to install Linux drivers:

- Manually Build the Linux Driver from Source
- Build a Binary RPM Package of the Linux Driver
- Install Linux Drivers via Prebuilt RPM Packages

In the instructions below:

- `<driver>` is the driver name, such as ice or iavf
- `x.x.x` is the driver version as indicated in the name of the driver tar file

Locations for Linux Driver Files

Source code for Linux drivers is available to download from the Intel Download Center or SourceForge at https://sourceforge.net/projects/e1000/.

Note: RPM packages are not available on SourceForge.

Manually Build the Linux Driver from Source

1. Move the base driver tar file to the directory of your choice. For example, use `/home/username/<driver>` or `/usr/local/src/<driver>`.
2. Untar/unzip the archive:

```
tar zxf <driver>-<x.x.x>.tar.gz
```

3. Change to the driver src directory:

   ```
   cd <driver>-<x.x.x>/src/
   ```

4. Compile the driver module:

   ```
   make install
   ```

   The binary will be installed as:

   ```
   /lib/modules/<KERNEL VER>/updates/drivers/net/ether
   net/intel/<driver>/<driver>.ko
   ```

   The install location listed above is the default location. This may differ for various Linux distributions.

   Note: For ice devices:

   - To build the driver using the schema for unified ethtool statistics defined in https://sourceforge.net/p/e1000/wiki/Home/, use the following command:

     ```
     make CFLAGS_EXTRA='-DUNIFIED_STATS' install
     ```

   - To compile the ice driver with ADQ (Application Device Queues) flags set, use the following command, where `<nproc>` is the number of logical cores:

     ```
     make -j<nproc> CFLAGS_EXTRA='-DADQ_PERF_COUN
     TERS' install
     ```

     (This will also apply the above `make install` command.)

   Note:

   For i40e devices: To gather and display additional statistics, use the `I40E_ADD_PROBES` pre-processor macro:

   ```
   make CFLAGS_EXTRA=-DI40E_ADD_PROBES
   ```

   Please note that this additional statistics gathering can impact performance.

   Note:

   For devices that support RDMA: You may see warnings from **depmod** related to unknown RDMA symbols during the make of the out-of-tree (OOT) base driver. These warnings are normal and appear because the in-tree RDMA driver will not work with the OOT base driver. To address the issue, you need to install the latest OOT versions of the base and RDMA drivers.

5. Load the module using the **modprobe** command.

   To check the version of the driver and then load it:

   ```
   modinfo <driver>modprobe <driver>
   ```

   Alternately, make sure that any older device drivers are removed from the kernel before loading the new module:

   ```
   rmmod <driver>; modprobe <driver>
   ```

6. Assign an IP address to the interface by entering the following, where `<ethX>` is the interface name that was shown in dmesg after modprobe:

   ```
   ip address add <IP_address>/<netmask bits> dev <eth
   X>
   ```

8. Verify that the interface works. Enter the following, where `<IP_address>` is the IP address for another machine on the same subnet as the interface that is being tested:

   ```
   ping <IP_address>
   ```

Note:

For certain distributions like (but not limited to) Red Hat Enterprise Linux 7 and Ubuntu, once the driver is installed, you may need to update the initrd/initramfs file to prevent the OS loading old versions of the driver.

For Red Hat distributions:

```
dracut --force
```

For Ubuntu:

```
update-initramfs -u
```

Build a Binary RPM Package of the Linux Driver

Note: RPM functionality has only been tested in Red Hat distributions.

1. Run the following command, where `<x.x.x>` is the version number for the driver tar file:

   ```
   rpmbuild -tb <driver>-<x.x.x>.tar.gz
   ```

   Note:
   For the build to work properly, the currently running kernel MUST match the version and configuration of the installed kernel sources. If you have just recompiled the kernel, reboot the system before building.

2. After building the RPM, the last few lines of the tool output contain the location of the RPM file that was built. Install the RPM with one of the following commands, where `<RPM>` is the location of the RPM file:

   ```
   rpm -Uvh <RPM>
   ```

   or:

   ```
   dnf/yum localinstall <RPM>
   ```

3. **For ice, i40e, and iavf devices only**: If your distribution or kernel does not contain inbox support for auxiliary bus, you must also install the auxiliary RPM:

   ```
   rpm -Uvh <driver RPM> <auxiliary RPM>
   ```

   or:

   ```
   dnf/yum localinstall <driver RPM> <auxiliary RPM>
   ```

   Note:
   On some distributions, the auxiliary RPM may fail to install due to missing kernel-devel headers. To workaround this issue, specify `--excludepath` during installation. For example:

   ```
   rpm -Uvh auxiliary-1.0.0-1.x86_64.rpm--excludepat
   h=/lib/modules/3.10.0-957.el7.x86_64/source/inclu
   de/linux/auxiliary_bus.h
   ```

- To compile the driver on some kernel/arch combinations, you may need to install a package with the development version of **libelf** (e.g., **libelf-dev**, **libelf-devel**, **elfutils-libelf-devel**).
- When compiling an out-of-tree driver, details will vary by distribution. However, you will usually need a kernel-devel RPM or some RPM that provides the kernel headers at a minimum. The RPM kernel-devel will usually fill in the link at

`/lib/modules/'uname -r'/build.`

Install Linux Drivers via Prebuilt RPM Packages

Intel provides prebuilt RPM Package Manager (RPM) files for the following distributions on the Intel Download Center:

- For Red Hat distributions: KMOD RPM files
- For SUSE distributions: Kernel Module Package (KMP) RPM files

See the following webpages on the RHEL and SLES websites for more information on installing RPM packages:

- Red Hat: How to install or upgrade an RPM package?
- SUSE: Introduction to RPM Packaging

Linux Secure Boot Mode

This software release includes RPM packages that contain:

- Device driver signed with Intel's private key in precompiled kernel module form
- RDMA driver
- Complete source code for above drivers
- Intel's public key

This release includes the Intel public key to allow you to authenticate the signed driver in secure boot mode. To authenticate the signed driver, you must place Intel's public key in the UEFI Secure Boot key database.

- The driver kernel module for a specific kernel version can be used with errata kernels within the same minor OS version, unless the errata kernel broke kABI. Whenever you update your kernel with an errata kernel, you must reinstall the driver RPM package.
- The RDMA driver will be installed if you reinstall the driver RPM package. If you want to remove the RDMA driver, you will have to do so every time you install the RPM package (for example, when you update your kernel with an errata kernel).
- If you decide to recompile the *.ko* module from the provided source files, the new *.ko* module will not be signed with any key. To use this *.ko* module in Secure Boot mode, you must sign it yourself with your own private key and add your public key to the UEFI Secure Boot key database.

Install FreeBSD Drivers

Note:

The FreeBSD driver package is to be used only as a standalone archive and the user should not attempt to incorporate it into the kernel source tree.

The instructions below apply to the following device series and drivers:

Device Series	Driver Name
Intel Adapter Virtual Functions	iavf
Intel Ethernet 800 Series	ice
Intel Ethernet 700 Series	ixl
Intel Ethernet 500 Series	ix
Intel Ethernet 300 Series	igb

In the instructions below:

- `<driver>` is the driver name, such as ice or iavf
- `x.x.x` is the driver version as indicated in the name of the driver tar file

1. Move the base driver tar file to the directory of your choice. For example, use `/home/username/<driver>` or `/usr/local/src/<driver>`.
2. Untar/unzip the archive:

   ```
   tar xzf <driver>-x.x.x.tar.gz
   ```

   This will create the `<driver>-x.x.x` directory.
3. To install the man page:

   ```
   cd <driver>-x.x.xgzip -c <driver>.4 > /usr/share/man/man4/<driver>.4.gz
   ```

4. To load the driver onto a running system:

   ```
   cd <driver>-x.x.xmakekldload ./if_<driver>.ko
   ```

   For the iavf driver, to install the driver without using iflib:

   ```
   cd iavf-x.x.x/srcmake legacykldload ./if_iavf.ko
   ```

> **Note:**
> Ensure the driver isn’t compiled into the currently running kernel. You can do that by adding `nodevice <driver>` to your kernel configuration file and rebuilding your kernel. See sections 10.4 and 10.5 in Configuring the FreeBSD Kernel in the FreeBSD kernel documentation for more information.

> **Note:**
> For ice devices, running the **make** command will not install the Dynamic Device Personalization (DDP) package and could cause the driver to fail to load. See step #7 below for more information.

5. To assign an IP address to the interface, enter the following, where `X` is the interface number for the device:

   ```
   ifconfig <driver>X <IP_address>
   ```

   For example:

   ```
   ifconfig ice0 <IP_address>
   ```

6. Verify that the interface works. Enter the following, where `<IP_address>` is the IP address for another machine on the same subnet as the interface that is being tested:

   ```
   ping <IP_address>
   ```

7. If you want the driver to load automatically when the system is booted, do the following.
   a. For all devices except ice:

      ```
      cd <driver>-x.x.x/srcmakemake install
      ```

      For ice devices:

      ```
      cd <driver>-x.x.xmakemake install
      ```

> **Note:**
> For ice devices, it’s important to do `make install` so that the driver loads the DDP package automatically.

   b. Edit `/boot/loader.conf` and add the following line:

      ```
      if_<driver>_load="YES"
      ```

8. If you want the device to connect to a network without manual intervention after a boot, edit `/etc/rc.conf` and create the appropriate `ifconfig_<driver>X` entry:

   ```
   ifconfig_<driver>X="<ifconfig_settings>"
   ```

For example:

```
ifconfig_ice0="inet 192.168.10.1 netmask 255.255.255.0"
```

Note: For assistance, see the ifconfig man page.

Install VMware Drivers

ESXi drivers for Intel Ethernet devices are only available from VMware's download site: https://customerconnect.vmware.com/home.

Please refer to VMware's download site for the latest drivers and installation information.

Install Windows Drivers

There are multiple ways to install device drivers on Microsoft Windows:

- Install via the Complete Driver Pack. This option will install Windows drivers and Intel® PROSet but is a very large download.
- Install Only Windows Drivers. This option will not install Intel PROSet but is a much smaller download.
- Install Base Drivers via the Command Line
- You can also Save and Restore an Adapter's Configuration Settings via the command line.

This page describes the above installation methods for Windows drivers. See Installing Intel PROSet for instructions on how to install Intel PROSet.

Note:

- To successfully install or uninstall the drivers or software, you must have administrative privileges on the computer completing installation.
- Installing the drivers will update the drivers for all supported Intel Ethernet adapters in your system.
- The Roll Back Driver feature of Windows Server (available on the Adapter Properties dialog's **Driver** tab) will not work correctly if an adapter team or

Intel PROSet is present on the system. Before you use the Roll Back Driver feature, remove any teams. Then remove Intel PROSet using **Programs and Features** from the Control Panel of Windows. See Installing Intel PROSet for details regarding Intel PROSet.

- Using Microsoft Windows Update to upgrade or downgrade your Ethernet network drivers is not supported. Please download the latest driver package from the support website.

Before Installing Drivers

Before installing or updating the drivers, insert your adapter(s) in the computer and plug in the network cable. When Windows discovers the new adapter, it attempts to find an acceptable Windows driver already installed with the operating system.

If found, the driver is installed without any user intervention. If Windows cannot find the driver, the Found New Hardware Wizard window is displayed.

Regardless of whether Windows finds the driver, it is recommended that you follow the procedures below to install the driver. Drivers for all Intel adapters supported by this software release are installed.

Install via the Complete Driver Pack

To download and install via the complete driver pack:

1. Download the latest software package from the support website and transfer it to the system. See Install via the Complete Driver Pack for the URL for the complete driver pack.
2. Extract the downloaded software package to your hard drive.
3. If the Found New Hardware Wizard screen is displayed, click **Cancel**.
4. Navigate in your extracted files to `\APPS\SETUP\SETUPBD` and then the Windows subfolder corresponding to your version of Windows (32-bit or 64-bit).
5. Inside the `Winx64` or `Win32` folder, double-click on *SetupBD.exe*.
6. Complete the steps in the installation wizard.
7. If you want to install Intel PROSet, navigate to `\APPS\PROSETDX` and then the Windows subfolder corresponding to your version of Windows (32-bit or 64-bit).
   - See About Intel PROSet® for more information about this tool.
8. Inside the `Winx64` or `Win32` folder, double-click on *DxSetup.exe*.
9. Complete the steps in the installation wizard.

Install Only Windows Drivers

To download and install only Windows drivers:

1. Go to the Intel Download Center and find the relevant download for your version of Windows. See Intel Download Center for the link.

2. Download and extract the **Wired_driver_XX.X_*.zip** file for your version of Windows, where XX.X is the release number. This file will install the base driver(s) for your system.
   - If you are running a 32-bit operating system, download **Wired_driver_XX.X_32.zip**.
   - If you are running a 64-bit operating system, download **Wired_driver_XX.X_x64.zip**.
3. In the extracted driver files, double-click on the **.exe** file to launch the installation.
4. In the dialog box that opens, click on **OK** to install the drivers.
5. The device driver(s) will install. Click **Close** when prompted.

Note:

This method does not install Intel PROSet. See Installing Intel PROSet for additional instructions.

Install Base Drivers via the Command Line

SetupBD.exe is a file that allows you to install the Windows base drivers from the command line. SetupBD is only available when you download and Install via the Complete Driver Pack.

To install Windows base drivers using SetupBD:

1. Download the complete driver pack. See Install via the Complete Driver Pack for details.
2. Extract the downloaded files to your desired location.
3. In the downloaded files, navigate to `APPS\SETUP\SETUPBD` and the appropriate subfolder for your version of Windows (32-bit or 64-bit).
4. From the command line, call SetupBD using the following syntax:

```
SetupBD [parameters]
```

   See below for supported command line parameters and examples.

Parameters

/h, /?

Displays help file for SetupBD.

/l *<path\filename>*

Create a log file with the specified path and filename. If you do not specify the path and filename, SetupBD creates a log file (named *SetupBD_<timestamp>.log*; for example, *SetupBD_18-04-2022_14-29-20.log*) in the current directory.

/m

Non-interactive install. This still displays the installation GUI, but you cannot interact with it. Use the /s switch to suppress the GUI.

/n

Ignore INF excludes and force scan all *.inf* files.

/nr

Suppress the system reboot. If you include this switch you will need to manually reboot the system for the changes to take effect. Must be used with the /s switch. This switch is ignored if it is included with the /r switch.

/r

Force a system reboot after the installation completes. Must be used with the /s switch.

/s

Silent installation. Suppresses all installation messages and errors.

Examples

To install and/or update the Windows driver(s) and display the GUI:

```
SetupBD
```

To install and/or update the Windows driver(s) silently:

```
SetupBD /s
```

To install and/or update the Windows driver(s) silently and create a log file in `c:\temp:

```
SetupBD /s /l c:\temp\install.log
```

To install and/or update the Windows driver(s) silently and force a reboot:

```
SetupBD /s /r
```

To install and/or update the Windows driver(s) silently and force a reboot (/nr is ignored):

```
SetupBD /s /r /nr
```

Save and Restore an Adapter's Configuration Settings

The Save and Restore Command Line Tool (*SaveRestore.ps1*) allows you to copy the current adapter and team settings into a standalone file (such as on a USB drive) as a backup measure. In the event of a hard drive failure, you can reinstate most of your former settings.

The system on which you restore network configuration settings must have the same configuration as the one on which the save was performed. A saved configuration file can be used to restore adapter settings after an operating system upgrade. However, all adapter configuration settings may not be restored depending on the features supported by the new operating system or adapter configuration software.

Note:

- You must have Administrator privileges to run scripts. If you do not have Administrator privileges, you will not receive an error, the script just will not run.
- Only adapter settings are saved (these include Intel ANS teaming and VLANs). The adapter’s driver is not saved.
- Restore using the script only once. Restoring multiple times may result in unstable configuration.
- Intel PROSet must be installed for the *SaveRestore.ps1* script to run. See Installing Intel PROSet for installation instructions.
- For systems running a 64-bit OS, be sure to run the 64-bit version of Windows PowerShell, not the 32-bit (x86) version, when running the *SaveRestore.ps1* script.

Command Line Syntax for SaveRestore.ps1

Use the following syntax to call *SaveRestore.ps1*:

```
SaveRestore.ps1 -Action save|restore [-ConfigPath] [-BDF
]
```

Parameters

-Action

Required. Valid values are:

save:

Saves adapter and team settings that have been changed from the default settings. When you restore with the resulting file, any settings not contained in the file are assumed to be the default.

restore:

Restores the settings.

-ConfigPath

Optional. Specifies the path and filename of the main configuration save file. If not specified, it is the script path and default filename (*saved_config.txt*).

-BDF

Optional. Default configuration file names are *saved_config.txt* and *Saved_StaticIP.txt*.

If you specify `-BDF` during a restore, the script attempts to restore the configuration based on the PCI Bus:Device:Function:Segment values of the saved configuration. If you removed, added, or moved a NIC to a different slot, this may result in the script applying the saved settings to a different device.

> **Note:**
>
> - If the restore system is not identical to the saved system, the script may not restore any settings when the `-BDF` option is specified.
> - Virtual Function devices do not support the `-BDF` option.

Examples for SaveRestore.ps1

Save Example

To save the adapter settings to a file on a removable media device:

1. Open a Windows PowerShell Prompt.
2. Navigate to the directory where *SaveRestore.ps1* is located (generally `c:\Program Files\Intel\Wired Networking\PROSET`).
3. Type the following:

```
SaveRestore.ps1 -Action Save -ConfigPath e:\settings.txt
```

Restore Example

To restore the adapter settings from a file on removable media:

1. Open a Windows PowerShell Prompt.
2. Navigate to the directory where *SaveRestore.ps1* is located (generally `c:\Program Files\Intel\Wired Networking\PROSET`).
3. Type the following:

```
SaveRestore.ps1 -Action Restore -ConfigPath e:\settings.txt
```

Release Notes

View the Intel® Ethernet Controller Products Release Notes on intel.com for information on added or removed features, supported operating systems for each driver, bug fixes, and known issues and limitations for the release.

You can view the latest release or past releases. To view the release notes for past releases:

1. Go to the following page on intel.com:
   https://www.intel.com/content/www/us/en/search.html?ws=idsa-default#q=Intel® Ethernet Controller Products Release Notes&sort=relevancy&f:@tabfilter=[Developers]
2. In the list of results, click on the link for "Intel® Ethernet Controller Products Release Notes."
   - By default, the page lists the newest released version.
   - If you need to view the release notes for a different version, click on the "More Versions" link in the right side of the search result and select your desired version.

Known Issues and Limitations

Known issues are no longer included in this user guide. See the Intel® Ethernet Controller Products Release Notes for all known issues and fixed issues.

Feature Support Matrix

We provide Feature Support Matrix documents for many of our product families. These documents detail supported features, cables, media types, operating systems, and other technical information.

View the following documents in the Intel Resource and Documentation Center:

- Intel Ethernet 800 Series:
  - Intel® Ethernet Controller E810 Feature Support Matrix
- Intel Ethernet 700 Series:
  - Intel® Ethernet Controller X710/XXV710/XL710 Feature Support Matrix
  - Intel® Ethernet Connection X722 Feature Support Matrix
  - Intel® Ethernet Controller X710-TM4/AT2 and V710-AT2 Feature Support Matrix
- Intel Ethernet 500 Series:
  - Intel® Ethernet Controller X550 Feature Support Matrix

Device Features

This chapter describes the features available on Intel Ethernet devices. Major features are organized alphabetically.

Note:

- Available settings are dependent on your device and operating system. Not all settings are available on every device/OS combination.
- Some features in this section refer to Intel PROSet, Intel PROSet for Windows Device Manager, Intel PROSet Adapter Configuration Utility (Intel PROSet ACU), or Intel PROSet for Windows PowerShell* software. Refer to About Intel PROSet® for more information on each of these applications.

- Remote Direct Memory Access (RDMA)
- Accessing Remote NVM Express* Drives Using RDMA
- Setting Speed and Duplex
- Thermal Monitoring
- Timestamps
- Transmit Buffers
- UEFI Network Device Drivers
- VF Loopback Pacing
- Virtualization Support
- Virtual LANs (VLANs)
- Wait for Link

Adapter Teaming

Intel® Advanced Network Services (Intel® ANS) teaming lets you take advantage of multiple adapters in a system by grouping them together. Intel ANS teaming can use features like fault tolerance and load balancing to increase throughput and reliability.

Before creating a team or adding team members, make sure each team member has been configured similarly. Settings to check include VLANs and QoS Packet Tagging, Jumbo Packets, and the various offloads. Pay particular attention when using different adapter models or adapter versions, as adapter capabilities vary.

For more information on VLANs, see Virtual LANs (VLANs).

See the following subsections for details.

- Teaming Modes
  - Overview
  - Adapter Fault Tolerance (AFT)
  - Switch Fault Tolerance (SFT)
  - Adaptive/Receive Load Balancing (ALB/RLB)
  - Virtual Machine Load Balancing (VMLB)
  - Static Link Aggregation (SLA)
  - IEEE 802.3ad Dynamic Link Aggregation
  - Multi-Vendor Teaming (MVT)
- Configuring Teams with Intel PROSet
  - Configuring Teams with Intel PROSet for Windows Device Manager
  - Configuring Teams with Windows PowerShell
  - Configuring Teams with Intel PROSet ACU

Configuration and Compatibility Notes

- Microsoft Windows 10 is the last Windows operating system version that supports Intel ANS. Intel ANS is not supported on Microsoft Windows 11 and later.
- Intel ANS is not supported on Microsoft Windows Server 2016 and later.
- To configure teams in Linux, use Channel Bonding, available in supported Linux kernels. For more information see the channel bonding documentation within the kernel source.
- Not all team types are available on all operating systems.
- Be sure to use the latest available drivers on all adapters.
- Not all Intel devices support Intel ANS. Intel adapters that do not support Intel ANS may still be included in a team. However, they are restricted in the same way non-Intel adapters are. See Multi-Vendor Teaming for more information.
- You cannot create a team that includes both Intel X710/XL710-based devices and Intel® I350-based devices. These devices are incompatible together in a team and will be blocked during team setup.
- NDIS 6.2 introduced new RSS data structures and interfaces. Because of this, you cannot enable RSS on teams that contain a mix of adapters that support NDIS 6.2 RSS and adapters that do not.
- If a team is bound to a Hyper-V virtual NIC, you cannot change the Primary or Secondary adapter.
- To assure a common feature set, some advanced features, including hardware offloading, are automatically disabled when an adapter that does not support the feature is added to a team.
- Hot Plug operations in a Multi-Vendor Team may cause system instability. We recommended that you restart the system or reload the team after performing Hot Plug operations with a Multi-Vendor Team. When you physically remove an adapter that is part of a team or a VLAN, you must reboot or reload the team/VLAN before using that adapter in the same network. This will prevent Ethernet address conflicts.
- Spanning tree protocol (STP) should be disabled on switch ports connected to teamed adapters in order to prevent data loss when the primary adapter is returned to service (failback). Alternatively, an activation delay may be configured on the adapters to prevent data loss when spanning tree is used.
- Data Center Bridging will be automatically disabled when an adapter is added to a team with non-DCB capable adapters.
- NLB will not work when Receive Load Balancing (RLB) is enabled. This occurs because NLB and iANS both attempt to set the server’s multicast MAC address, resulting in an ARP table mismatch.
- Teaming with the Intel® 10 Gigabit AF DA Dual Port Server Adapter is only supported with similar adapter types and models or with switches using a Direct Attach connection.
- If you want to set up VLANs on a team, you must first create the team.
- After adding a VLAN to the team, the Network Connections window shows the team as disabled or network cable unplugged. This is normal. The connection protocols are now bound to the VLAN on the team. You can configure the connection protocols in the Properties for the VLAN.

Teaming and VLAN Considerations When Replacing Adapters

After installing an adapter in a specific slot, Windows treats any other adapter of the same type as a new adapter. Also, if you remove the installed adapter and insert it into a different slot, Windows recognizes it as a new adapter. Make sure that you follow the instructions below carefully.

1. If the adapter is part of a team, remove the adapter from the team.
2. Shut down the system and unplug the power cable.
3. Disconnect the network cable from the adapter.
4. Open the case and remove the adapter.
5. Insert the replacement adapter. (Use the same slot, otherwise Windows assumes that there is a new adapter.)
6. Reconnect the network cable.
7. Close the case, reattach the power cable, and power up the server.

Microsoft Load Balancing and Failover (LBFO) Teams

Intel ANS teaming and VLANs are not compatible with Microsoft's LBFO teams. Intel PROSet will block a member of an LBFO team from being added to an Intel ANS team or VLAN. You should not add a port that is already part of an Intel ANS team or VLAN to an LBFO team, as this may cause system instability. If you use an Intel ANS team member or VLAN in an LBFO team, perform the following procedure to restore your configuration:

1. Reboot the machine
2. Remove LBFO team. Even though LBFO team creation failed, after a reboot Server Manager will report that LBFO is Enabled, and the LBFO interface is present in the "NIC Teaming" GUI.
3. Remove the Intel ANS teams and VLANs involved in the LBFO team and recreate them. This is an optional (all bindings are restored when the LBFO team is removed), but strongly recommended step

Note:

- If you add an Intel AMT enabled port to an LBFO team, do not set the port to Standby in the LBFO team. If you set the port to Standby you may lose AMT functionality.
- DCB is incompatible with Microsoft Server LBFO Teams. Do not create an LBFO team when DCB is installed. Do not install DCB if you use LBFO teaming. Install failures and persistent link loss may occur if DCB and LBFO are used on the same port.

Supported Adapters

Teaming options are supported on Intel server adapters. Selected adapters from other manufacturers are also supported. If you are using a Windows-based computer, adapters that appear in Intel PROSet may be included in a team.

> **Note:**
>
> In order to use adapter teaming, you must have at least one Intel server adapter in your system. Furthermore, all adapters must be linked to the same switch or hub.

Conditions that may prevent you from teaming a device: During team creation or modification, the list of available team types or list of available devices may not include all team types or devices. This may be caused by any of several conditions, including:

- The device does not support the desired team type or does not support teaming at all.
- The operating system does not support the desired team type.
- The devices you want to team together use different driver versions.
- TOE (TCP Offload Engine) enabled devices cannot be added to an Intel ANS team and will not appear in the list of available adapters.
- You can add Intel® Active Management Technology (Intel® AMT) enabled devices to Adapter Fault Tolerance (AFT), Switch Fault Tolerance (SFT), and Adaptive Load Balancing (ALB) teams. All other team types are not supported. The Intel AMT enabled device must be designated as the primary adapter for the team.
- The device's MAC address is overridden by the Locally Administered Address advanced setting.
- The device has "OS Controlled" or "Enabled" selected on the Data Center tab.
- The device has a virtual NIC bound to it.
- The device is part of a Microsoft Load Balancing and Failover (LBFO) team.

Teaming Modes

Overview

The following teaming modes are supported, and are described later in this page:

- Adapter Fault Tolerance (AFT)
- Switch Fault Tolerance (SFT)
- Adaptive/Receive Load Balancing (ALB/RLB)
- Virtual Machine Load Balancing (VMLB)
- Static Link Aggregation (SLA)
- IEEE 802.3ad Dynamic Link Aggregation
- Multi-Vendor Teaming (MVT)

Important:

- Be sure to use the latest available drivers on all adapters.
- Before creating a team, adding or removing team members, or changing advanced settings of a team member, make sure each team member has been configured similarly. Settings to check include VLANs and QoS Packet Tagging, Jumbo Frames, and the various offloads. These settings are available in Intel PROSet’s Advanced tab. *Pay particular attention when using different adapter models or adapter versions, as adapter capabilities vary.*
- If team members implement Advanced features differently, failover and team functionality will be affected. To avoid team implementation issues:
    - Create teams that use similar adapter types and models.
    - Reload the team after adding an adapter or changing any Advanced features. One way to reload the team is to select a new preferred primary adapter. Although there will be a temporary loss of network connectivity as the team reconfigures, the team will maintain its network addressing schema.

Note:

- Hot Plug operations for an adapter that is part of a team are only available in Windows Server.
- For SLA teams, all team members must be connected to the same switch. For AFT, ALB, and RLB teams, all team members must belong to the same subnet. The members of an SFT team must be connected to a different switch.
- Teaming only one adapter port is possible, but provides no benefit.

Primary and Secondary Adapters

Teaming modes that do not require a switch with the same capabilities (AFT, SFT, ALB (with RLB)) use a primary adapter. In all of these modes except RLB, the primary is the only adapter that receives traffic. RLB is enabled by default on an ALB team.

If the primary adapter fails, another adapter will take over its duties. If you are using more than two adapters, and you want a specific adapter to take over if the primary fails, you must specify a secondary adapter. If an Intel AMT enabled device is part of a team, it must be designated as the primary adapter for the team.

There are two types of primary and secondary adapters:

- **Default primary adapter**: If you do not specify a preferred primary adapter, the software will choose an adapter of the highest capability (model and speed) to act as the default primary. If a failover occurs, another adapter becomes the primary. Once the problem with the original primary is resolved, the traffic will not automatically restore to the default (original) primary adapter in most modes. The adapter will, however, rejoin the team as a non-primary.

- **Preferred Primary/Secondary adapters**: You can specify a preferred adapter. Under normal conditions, the Primary adapter handles all traffic. The Secondary adapter will receive traffic if the primary fails. If the Preferred Primary adapter fails, but is later restored to an active status, control is automatically switched back to the Preferred Primary adapter. Specifying primary and secondary adapters adds no benefit to SLA and IEEE 802.3ad dynamic teams, but doing so forces the team to use the primary adapter's MAC address.

Failover and Failback

When a link fails, either because of port or cable failure, team types that provide fault tolerance will continue to send and receive traffic. Failover is the initial transfer of traffic from the failed link to a good link. Failback occurs when the original adapter regains link. You can use the Activation Delay setting (located on the Advanced tab of the team's properties in Device Manager) to specify a how long the failover adapter waits before becoming active. If you don't want your team to failback when the original adapter gets link back, you can set the Allow Failback setting to disabled (located on the Advanced tab of the team's properties in Device Manager).

Adapter Fault Tolerance (AFT)

Adapter Fault Tolerance (AFT) provides automatic recovery from a link failure caused from a failure in an adapter, cable, switch, or port by redistributing the traffic load across a backup adapter.

Failures are detected automatically, and traffic rerouting takes place as soon as the failure is detected. The goal of AFT is to ensure that load redistribution takes place fast enough to prevent user sessions from being disconnected.

AFT supports two to eight adapters per team. Only one active team member transmits and receives traffic. If this primary connection (cable, adapter, or port) fails, a secondary, or backup, adapter takes over. After a failover, if the connection to the user-specified primary adapter is restored, control passes automatically back to that primary adapter.

AFT is the default mode when a team is created. This mode does not provide load balancing.

Note:

- AFT teaming requires that the switch not be set up for teaming and that spanning tree protocol is turned off for the switch port connected to the NIC or LOM on the server.
- All members of an AFT team must be connected to the same subnet.

Switch Fault Tolerance (SFT)

Switch Fault Tolerance (SFT) supports only two NICs in a team connected to two different switches.

In SFT, one adapter is the primary adapter and one adapter is the secondary adapter. During normal operation, the secondary adapter is in standby mode. In standby, the adapter is inactive and waiting for failover to occur. It does not transmit or receive network traffic. If the primary adapter loses connectivity, the secondary adapter automatically takes over. When SFT teams are created, the Activation Delay is automatically set to 60 seconds.

In SFT mode, the two adapters creating the team can operate at different speeds.

Note:

SFT teaming requires that the switch not be set up for teaming and that spanning tree protocol is turned on.

Configuration Monitoring

You can set up monitoring between an SFT team and up to five IP addresses. This allows you to detect link failure beyond the switch. You can ensure connection availability for several clients that you consider critical. If the connection between the primary adapter and all of the monitored IP addresses is lost, the team will failover to the secondary adapter.

Adaptive/Receive Load Balancing (ALB/RLB)

Adaptive Load Balancing (ALB) is a method for dynamic distribution of data traffic load among multiple physical channels. The purpose of ALB is to improve overall bandwidth and end station performance. In ALB, multiple links are provided from the server to the switch, and the intermediate driver running on the server performs the load balancing function. The ALB architecture utilizes knowledge of Layer 3 information to achieve optimum distribution of the server transmission load.

ALB is implemented by assigning one of the physical channels as Primary and all other physical channels as Secondary. Packets leaving the server can use any one of the physical channels, but incoming packets can only use the Primary Channel. With Receive Load Balancing (RLB) enabled, it balances IP receive traffic. The intermediate driver analyzes the send and transmit loading on each adapter and balances the rate across the adapters based on destination address. Adapter teams configured for ALB and RLB also provide the benefits of fault tolerance.

Note:

- ALB teaming requires that the switch not be set up for teaming and that spanning tree protocol is turned off for the switch port connected to the network adapter in the server.
- ALB does not balance traffic when protocols such as NetBEUI and IPX* are used. You may create an ALB team with mixed speed adapters. The load is balanced according to the adapter’s capabilities and bandwidth of the channel.
- All members of ALB and RLB teams must be connected to the same subnet.
- Virtual NICs cannot be created on a team with Receive Load Balancing enabled. Receive Load Balancing is automatically disabled if you create a virtual NIC on a team.

Virtual Machine Load Balancing (VMLB)

Virtual Machine Load Balancing (VMLB) provides transmit and receive traffic load balancing across Virtual Machines bound to the team interface, as well as fault tolerance in the event of switch port, cable, or adapter failure.

The driver analyzes the transmit and receive load on each member adapter and balances the traffic across member adapters. In a VMLB team, each Virtual Machine is associated with one team member for its TX and RX traffic.

If only one virtual NIC is bound to the team, or if Hyper-V is removed, then the VMLB team will act like an AFT team.

Note:

- VMLB does not load balance non-routed protocols such as NetBEUI and some IPX* traffic.
- VMLB supports from two to eight adapter ports per team.
- You can create a VMLB team with mixed speed adapters. The load is balanced according to the lowest common denominator of adapter capabilities and the bandwidth of the channel.
- You cannot use an Intel AMT enabled adapter in a VMLB team.

Static Link Aggregation (SLA)

Static Link Aggregation (SLA) is very similar to Adaptive/Receive Load Balancing (ALB/RLB), taking several physical channels and combining them into a single logical channel.

This mode works with:

- Cisco EtherChannel capable switches with channeling mode set to “on”
- Intel switches capable of Link Aggregation
- Other switches capable of static 802.3ad

Note:

- All adapters in a Static Link Aggregation team must run at the same speed and must be connected to a Static Link Aggregation-capable switch. If the speed capabilities of adapters in a Static Link Aggregation team are different, the speed of the team is dependent on the switch.
- Static Link Aggregation teaming requires that the switch be set up for Static Link Aggregation teaming and that spanning tree protocol is turned off.
- An Intel AMT enabled adapter cannot be used in an SLA team.

IEEE 802.3ad Dynamic Link Aggregation

IEEE 802.3ad is the IEEE standard. Teams can contain two to eight adapters. You must use 802.3ad switches (in dynamic mode, aggregation can go across switches). Adapter teams configured for IEEE 802.3ad also provide the benefits of fault tolerance and load balancing. Under 802.3ad, all protocols can be load balanced.

Dynamic mode supports multiple aggregators. Aggregators are formed by port speed connected to a switch. For example, a team can contain adapters running at 1Gbps and 10Gbps, but two aggregators will be formed, one for each speed. Also, if a team contains 1Gbps ports connected to one switch, and a combination of 1Gbps and 10Gbps ports connected to a second switch, three aggregators would be formed. One containing all the ports connected to the first switch, one containing the 1 Gbps ports connected to the second switch, and the third containing the 10Gbps ports connected to the second switch.

Note:

- IEEE 802.3ad teaming requires that the switch be set up for IEEE 802.3ad (link aggregation) teaming and that spanning tree protocol is turned off.
- Once you choose an aggregator, it remains in force until all adapters in that aggregation team lose link.
- In some switches, copper and fiber adapters cannot belong to the same aggregator in an IEEE 802.3ad configuration. If there are copper and fiber adapters installed in a system, the switch might configure the copper adapters in one aggregator and the fiber-based adapters in another. If you experience this behavior, for best performance you should use either only copper-based or only fiber-based adapters in a system.
- An Intel AMT enabled adapter cannot be used in a DLA team.

Before you begin:

- Verify that the switch fully supports the IEEE 802.3ad standard.
- Check your switch documentation for port dependencies. Some switches require pairing to start on a primary port.
- Check your speed and duplex settings to ensure the adapter and switch are running at full duplex, either forced or set to auto-negotiate. Both the adapter and the switch must have the same speed and duplex configuration. The full-duplex requirement is part of the IEEE 802.3ad specification: http://standards.ieee.org/. If needed, change your speed or duplex setting before you link the adapter to the switch. Although you can change speed and duplex settings after the team is created, Intel recommends you disconnect the cables until settings are in effect. In some cases, switches or servers might not appropriately recognize modified speed or duplex settings if settings are changed when there is an active link to the network.
- If you are configuring a VLAN, check your switch documentation for VLAN compatibility notes. Not all switches support simultaneous dynamic 802.3ad teams and VLANs. If you do choose to set up VLANs, configure teaming and VLAN settings on the adapter before you link the adapter to the switch. Setting up VLANs after the switch has created an active aggregator affects VLAN

functionality.

Multi-Vendor Teaming (MVT)

Multi-Vendor Teaming (MVT) allows teaming with a combination of Intel and non-Intel adapters.

If you are using a Windows-based computer, adapters that appear in the Intel PROSet teaming wizard can be included in a team.

MVT Design Considerations:

- In order to activate MVT, you must have at least one Intel adapter or integrated connection in the team, which must be designated as the primary adapter.
- A multi-vendor team can be created for any team type.
- All members in an MVT must operate on a common feature set (lowest common denominator).
- Manually verify that the frame setting for the non-Intel adapter is the
- same as the frame settings for the Intel adapters.
- If a non-Intel adapter is added to a team, its RSS settings must match the Intel adapters in the team.

Configuring Teams with Intel PROSet

This section describes how to configure Intel ANS teams using Intel PROSet.

Refer to the following subsections for more specific information:

- Configuring Teams with Intel PROSet for Windows Device Manager
- Configuring Teams with Windows PowerShell
- Configuring Teams with Intel PROSet ACU

Configuring Teams with Intel PROSet for Windows Device Manager

Creating a Team

1. Launch Windows Device Manager.
2. Expand **Network Adapters**.
3. Double-click on one of the adapters that will be a member of the team. The adapter properties dialog box appears.
4. Click the **Teaming** tab.
5. Click **Team with other adapters**.
6. Click **New Team**.
7. Type a name for the team, then click **Next**.

8. Click the checkbox of any adapter you want to include in the team, then click **Next**.
9. Select a teaming mode, then click **Next**.
10. Click **Finish**.

The Team Properties window appears, showing team properties and settings.

Once a team has been created, it appears in the Network Adapters category in the Computer Management window as a virtual adapter. The team name also precedes the adapter name of any adapter that is a member of the team.

Changing Which Adapters Are In a Team

1. Launch Windows Device Manager.
2. Open the Team Properties dialog box by double-clicking on a team listing in the Computer Management window.
3. Click the **Settings** tab.
4. Click **Modify Team**, then click the **Adapters** tab.
5. Select the adapters that will be members of the team.
    - Click the checkbox of any adapter that you want to add to the team.
    - Clear the checkbox of any adapter that you want to remove from the team.
6. Click **OK**.

Renaming a Team

1. Open the Team Properties dialog box by double-clicking on a team listing in the Computer Management window.
2. Click the **Settings** tab.
3. Click **Modify Team**, then click the **Name** tab.
4. Type a new team name, then click **OK**.

Note:

If you modify a team name from the team property sheet, it may take several minutes for the name to change in Device Manager. Closing and opening Device Manager will load the new name.

Removing a Team

1. Open the Team Properties dialog box by double-clicking on a team listing in the Computer Management window.
2. Click the **Settings** tab.
3. Select the team you want to remove, then click **Remove Team**.
4. Click **Yes** when prompted.

> **Note:**
>
> If you defined a VLAN or QoS Prioritization on an adapter joining a team, you may have to redefine it when it is returned to a standalone mode.

Specifying a Preferred Primary or Secondary Adapter

You must specify a primary adapter before you can specify a secondary adapter.

1. In the Team Properties dialog box's **Settings** tab, click **Modify Team**.
2. On the **Adapters** tab, select an adapter.
3. Click **Set Primary** or **Set Secondary**.
4. Click **OK**.

The adapter's preferred setting appears in the Priority column on Intel PROSet's **Team Configuration** tab. A "1" indicates a preferred primary adapter, and a "2" indicates a preferred secondary adapter.

Configuring Teams with Windows PowerShell

Using non-Intel cmdlets, such as the `Set-NetAdapterAdvancedProperty` cmdlet provided in Microsoft PowerShell*, to change settings for an ANS-teamed adapter may cause the team to stop using that adapter to pass traffic. You may see this as reduced performance or the adapter being disabled in the PROSet Teaming GUI. You can repair the issue by changing the setting back to its previous state, or by removing the adapter from the ANS team and then adding it back.

Creating a Team

Use the `New-IntelNetTeam` cmdlet. For example:

```
New-IntelNetTeam -TeamMemberNames "<adapter1_name>", "<a
dapter2_name>"-TeamMode AdapterFaultTolerance -TeamName
"<team_name>"
```

Changing Which Adapters Are in a Team

Use the `Add-IntelNetTeamMember` or `Remove-IntelNetTeamMember` cmdlet. For example:

```
Add-IntelNetTeamMember -TeamName "<team_name>" -Name "<a
dapter_name>"
```

Renaming a Team

Use the `Set-IntelNetTeam` cmdlet. For example:

```
Set-IntelNetTeam -TeamName "<team_name>" -NewTeamName "<
new_team_name>"
```

Removing a Team

Use the `Remove-IntelNetTeam` cmdlet. For example:

```
Remove-IntelNetTeamMember -Name "<adapter_name>"
```

> **Note:**
>
> If you defined a VLAN or QoS Prioritization on an adapter joining a team, you may have to redefine it when it is returned to a standalone mode.

Specifying a Preferred Primary or Secondary Adapter

Use the `Set-IntelNetTeam` cmdlet. For example:

```
Set-IntelNetTeam -TeamName "Team 1" -PrimaryAdapterName
"<adapter1_name>"-SecondaryAdapterName "<adapter2_name>"
```

Configuring Teams with Intel PROSet ACU

Creating a Team with Intel PROSet ACU

1. Launch Intel PROSet ACU.
2. Select an adapter to start the team.
3. Click the **Teaming/VLANs** tab.
4. In the Teaming panel, click **Create Team**.
5. Select the adapters to include in the team, then click **Next**.
6. Type a name for the team.
7. Select the teaming mode, then click **Next**.
8. [Optional] Designate Primary and Secondary adapters for the team.
9. Click **Finish**.

Changing Which Adapters Are in a Team

1. Launch Intel PROSet ACU.
2. Select the team you wish to modify.
3. In the Team Members panel, click **Modify Members**.
4. Select the adapters that will be members of the team.

5. Click **Apply Changes**.

Renaming a Team

1. Launch Intel PROSet ACU.
2. Select the team you wish to modify.
3. In the Team Information panel, type a new team name.
4. Click **Apply Changes**.

Removing a Team

1. Launch Intel PROSet ACU.
2. Select the team you wish to remove.
3. Click **Remove Team**.

Note:

If you defined a VLAN or QoS Prioritization on an adapter joining a team, you may have to redefine it when it is returned to a standalone mode.

Specifying a Preferred Primary or Secondary Adapter

You must specify a primary adapter before you can specify a secondary adapter.

1. Launch Intel PROSet ACU.
2. Select the team you wish to modify.
3. Select your preferred Primary Adapter.
4. Select your preferred Secondary Adapter.
5. Click **Apply Changes**.

Adaptive Inter-Frame Spacing

This setting compensates for excessive Ethernet packet collisions on the network.

The default setting works best for most computers and networks. By enabling this feature, the network adapter dynamically adapts to the network traffic conditions. However, in some rare cases you might obtain better performance by disabling this feature. This setting forces a static gap between packets.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Adaptive Inter-Frame Spacing" -DisplayValue "Ena
bled"
```

Possible values for this setting are:

- Enabled
- Disabled

Data Center Bridging (DCB)

Data Center Bridging (DCB) is a collection of standards-based extensions to classical Ethernet. It provides a lossless data center transport layer that enables the convergence of LANs and SANs onto a single unified fabric.

Furthermore, DCB is a configuration Quality of Service implementation in hardware. It uses the VLAN priority tag (802.1p) to filter traffic. That means that there are 8 different priorities that traffic can be filtered into. It also enables priority flow control (802.1Qbb) which can limit or eliminate the number of dropped packets during network stress. Bandwidth can be allocated to each of these priorities, which is enforced at the hardware level (802.1Qaz).

DCB includes the following capabilities:

- Priority-based flow control (PFC; IEEE 802.1Qbb)
- Enhanced transmission selection (ETS; IEEE 802.1Qaz)
- Congestion notification (CN)
- Extensions to the Link Layer Discovery Protocol (LLDP) standard (IEEE 802.1AB) that enable Data Center Bridging Capability Exchange Protocol (DCBX)

Adapter firmware implements LLDP and DCBX protocol agents as per 802.1AB and 802.1Qaz respectively.

There are two supported versions of DCBX:

- CEE Version
- IEEE Version

Note:

The OS DCBX stack defaults to the CEE version of DCBX, and if a peer is transmitting IEEE TLVs, it will automatically transition to the IEEE version.

For more information on DCB, including the DCB Capability Exchange Protocol Specification, go to http://www.ieee802.org/1/pages/dcbridges.html

Configuring DCB for Windows

To change this setting in Intel PROSet:

This setting is found on the Data Center tab of the device's Device Manager property sheet or in the Data Center panel in Intel PROSet ACU.

You can use Intel PROSet to perform the following tasks:

- Display status:
  - Enhanced Transmission Selection
  - Priority Flow Control
  - Non-operational status: If the Status indicator shows that DCB is non-operational, there may be a number of possible reasons:
    - DCB is not enabled - select the checkbox to enable DCB.
    - One or more of the DCB features is in a non-operational state.

    A non-operational status is most likely to occur when **Use Switch Settings** is selected or **Using Advanced Settings** is active. This is generally a result of one or more of the DCB features not getting successfully exchanged with the switch. Possible problems include:

    - One of the features is not supported by the switch.
    - The switch is not advertising the feature.
    - The switch or host has disabled the feature (this would be an advanced setting for the host).
- Disable/enable DCB
- Troubleshooting information

Note:

- On X710 based devices running Microsoft Windows, DCB is only supported on NVM version 4.52 and newer. Older NVM versions must be updated before the adapter is capable of DCB support in Windows.
- If *QOS/DCB is not available, it may be for one of the following reasons:
  - The Firmware LLDP (FW-LLDP) agent was disabled from a pre-boot environment (typically UEFI).
  - This device is based on the Intel® Ethernet Controller X710 and the current link speed is 2.5 Gbps or 5 Gbps.

Hyper-V (DCB and VMQ)

Note:

Configuring a device in the VMQ + DCB mode reduces the number of VMQs available for guest OSes.

DCB for Linux

Intel Ethernet drivers support firmware-based or software-based DCBX in Linux, depending on the underlying PF device. The following table summarizes DCBX support by driver.

Linux Driver	Firmware-Based DCBX	Software-Based DCBX
ice	Supported	Supported
i40e	Supported	Supported
ixgbe	Not supported	Supported

In **firmware-based** mode, firmware intercepts all LLDP traffic and handles DCBX negotiation transparently for the user. In this mode, the adapter operates in "willing" DCBX mode, receiving DCB settings from the link partner (typically a switch). The local user can only query the negotiated DCB configuration.

In **software-based** mode, LLDP traffic is forwarded to the network stack and user space, where a software agent can handle it. In this mode, the adapter can operate in either "willing" or "nonwilling" DCBX mode and DCB configuration can be both queried and set locally. Software-based mode requires the FW-based LLDP Agent to be disabled and kernel `CONFIG_DCB` enabled.

Note:

- Only one LLDP/DCBX agent can be active on a single interface at a time.
- Software-based and firmware-based DCBX modes are mutually exclusive.
- When the firmware DCBX agent is active, software agents will not be able to receive or transmit LLDP frames. See Firmware Link Layer Discovery Protocol (FW-LLDP), as well as the Linux driver readme in your installation, for information on enabling or disabling the FW-LLDP agent.
- In software-based DCBX mode, you can configure DCB parameters using software LLDP/DCBX agents that interface with the Linux kernel's DCB Netlink API. We recommend using OpenLLDP as the DCBX agent when running in software mode. For more information, see the OpenLLDP man pages and https://github.com/intel/openlldp.
- For information on configuring DCBX parameters on a switch, please consult the switch manufacturer's documentation.

iSCSI Over DCB

Intel® Ethernet adapters support iSCSI software initiators that are native to the underlying operating system. Data Center Bridging is most often configured at the switch. If the switch is not DCB capable, the DCB handshake will fail but the iSCSI connection will not be lost.

Note:

DCB does not install in a VM. iSCSI over DCB is only supported in the base OS. An iSCSI initiator running in a VM will not benefit from DCB Ethernet enhancements.

Configuring iSCSI Over DCB in Windows

iSCSI installation includes the installation of the iSCSI DCB Agent (*iscsidcb.exe*) user mode service. The Microsoft iSCSI Software Initiator enables the connection of a Windows host to an external iSCSI storage array using an Intel Ethernet adapter. Please consult your operating system documentation for configuration details.

To change this setting in Intel PROSet:

This setting is found on the Data Center tab of the device's Device Manager property sheet or in the Data Center panel in Intel PROSet ACU.

This setting provides feedback as to the DCB state, operational or non-operational, as well as providing additional details should it be non-operational.

Note:

On Microsoft Windows Server, if you configure Priority using IEEE, the iSCSI policy may not be created automatically. To create the iSCSI policy manually, use Powershell and type:

```
New-NetQosPolicy -Name "UP4" -PriorityValue 8021 Acti
on 4 -iSCSI
```

Using iSCSI over DCB with Intel ANS Teaming

The Intel® iSCSI Agent is responsible for maintaining all packet filters for the purpose of priority tagging iSCSI traffic flowing over DCB-enabled adapters. The iSCSI Agent will create and maintain a traffic filter for an Intel ANS Team if at least one member of the team has an "Operational" DCB status. However, if any adapter on the team does not have an "Operational" DCB status, the iSCSI Agent will log an error in the Windows Event Log for that adapter. These error messages are to notify the administrator of configuration issues that need to be addressed, but do not affect the tagging or flow of iSCSI traffic for that team, unless it explicitly states that the TC Filter has been removed.

See Adapter Teaming for more information on Intel ANS teams.

Configuring iSCSI Over DCB in Linux

In the case of Open Source distributions, virtually all distributions include support for an Open iSCSI Software Initiator and Intel® Ethernet adapters will support them. Please consult your distribution documentation for additional configuration details on their particular Open iSCSI initiator.

Intel® 82599-based adapters support iSCSI within a Data Center Bridging cloud. Used in conjunction with switches and targets that support the iSCSI/DCB application TLV, this solution can provide guaranteed minimum bandwidth for iSCSI traffic between the host and target. This solution enables storage administrators to segment iSCSI traffic from LAN traffic. Previously, iSCSI traffic within a DCB supported environment was treated as LAN traffic by switch vendors. Please consult your switch and target vendors to ensure that they support the iSCSI/DCB application TLV.

Direct Memory Access (DMA) Coalescing

DMA (Direct Memory Access) allows the network device to move packet data directly to the system's memory, reducing CPU utilization. However, the frequency and random intervals at which packets arrive do not allow the system to enter a lower power state. DMA Coalescing allows the NIC to collect packets before it initiates a DMA event. This may increase network latency but also increases the chances that the system will consume less energy. Adapters and network devices based on the Intel® Ethernet Controller I350 (and later controllers) support DMA Coalescing.

Higher DMA Coalescing values result in more energy saved but may increase your system's network latency. If you enable DMA Coalescing, you should also set the Interrupt Moderation Rate to **Minimal**. This minimizes the latency impact imposed by DMA Coalescing and results in better peak network throughput performance.

You must enable DMA Coalescing on all active ports in the system. You may not gain any energy savings if it is enabled only on some of the ports in your system. There are also several BIOS, platform, and application settings that will affect your potential energy savings. A white paper containing information on how to best configure your platform is available on the Intel website.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "DMA Coalescing"-DisplayValue "Enabled"
```

Dynamic Device Personalization (DDP)

Devices based on the Intel® Ethernet 800 Series require a Dynamic Device Personalization (DDP) package file to enable advanced features (such as dynamic tunneling, Intel Ethernet Flow Director, RSS, and ADQ).

DDP allows you to change the packet processing pipeline of a device by applying a profile package to the device at runtime. Profiles can be used to, for example, add support for new protocols, change existing protocols, or change default settings. DDP profiles can also be rolled back without rebooting the system.

The DDP package loads during device initialization or driver runtime, depending on the operating system. The driver checks to see if the DDP package is present and compatible. If this file exists, the driver will load it into the device. If not, the driver will go into Safe Mode where it will use the configuration contained in the device's NVM.

Safe Mode disables advanced and performance features, and supports only basic traffic and minimal functionality, such as updating the NVM or downloading a new driver or DDP package. For more information, see Safe Mode.

A general-purpose, default DDP package is automatically installed with all supported Intel Ethernet 800 Series drivers on supported operating systems. Additional DDP packages are available to address needs for specific market segments or targeted solutions.

Refer to the following for more information on configuring DDP:

- Intel® Ethernet Controller E810 Dynamic Device Personalization (DDP) Technology Guide
  - Search for "DDP" on the Intel Resource and Documentation Center for additional DDP technology and configuration guides.
- The readme file inside the DDP package zip file

Note:

- If you are using DPDK, see the DPDK documentation at https://www.dpdk.org/ for installation instructions and more information.
- In ESXi:
  - Support for DDP packages for specific market segments requires the following:
    - Driver: icen 1.9.1.x or higher
    - Tool: intnet 1.8.3.x or higher
  - Use esxcli to load and unload DDP packages for specific market segments during driver runtime.
  - A package update is not persistent between device resets or system reboots.

Firmware

Firmware is a layer of software that is programmed into a device's memory. It provides low level functionality for the device. In most cases you will not notice the firmware on your device at all. Firmware error states usually occur because of an unsuccessful update.

See the following subsections for details.

- Firmware Security
- Firmware Rollback Mode
- Firmware Recovery Mode
- Using Devlink to Update a Device’s NVM
- Firmware Logging
- Debug Dump

Firmware Security

Intel or your equipment manufacturer will occasionally release a firmware security patch. We recommend that you update your firmware to the latest version available for your device to take advantage of these security patches. Firmware updates for Intel Ethernet devices will have a Security Revision number (SRev).

Minimum Security Revision Enforcement

Firmware security updates can be undone if you install a previous version of the firmware onto your device. Intel firmware releases include a Minimum Security Revision (MinSRev) enforcement feature. This means you can block someone from installing a lower revision of the firmware onto your device. This will limit the rollback capabilities of your device. The firmware update process will block the update if the supplied firmware has a lower security revision (SRev) than the MinSRev value of the firmware currently loaded on the device. Only update the MinSRev value if you are certain you will not need to roll the firmware back to an earlier version.

You can update the MinSRev value during the firmware update process, locking the current security version in as the new MinSRev baseline, by using the `-optinminsrev` command line option.

> **Important:**
>
> The MinSRev value on a device can never be decreased. Once the MinSRev is increased, NVM downgrades attempting to install a lower Security revision (SRev) than the current MinSRev will be rejected by the device. Users who want to downgrade firmware without regard to security revisions should not use this feature.

SRev and MinSRev Examples

To view your device’s current SRev and MinSRev:

You can use the nvmupdate tool’s inventory mode to view your device’s current SRev and MinSRev values as follows:

- Windows:

```
nvmupdatew64e -i  -l update.log -o results.xml -c n
vmupdate.cfg -optinminsrev
```

- Linux:

```
nvmupdate64e -i  -l update.log -o results.xml -c nv
mupdate.cfg
```

Where:

-i

Sets nvmupdate to inventory mode.

-l update.log

Specifies the name of the log file.

-o results.xml

Specifies the name of the results file. This is an XML file that contains the inventory/update results.

-c nvmupdate.cfg

Specifies the name of the configuration file. This is a text file that contains descriptions of networking devices and firmware versions for those devices.

-optinminsrev

Specifies that the MinSRev and SRev values are included in the *results.xml* file.

Examine the *results.xml* file for the SRev and MinSRev values.

> **Note:**
>
> Make sure you specify `-i` for inventory mode. If you specify `-u`, the tool will update the MinSRev value, rather than simply disclose it. You can achieve the same results by specifying `MINSREV:TRUE` in the configuration file.

See Intel® Ethernet NVM Update Tool for more information on how to use the nvmupdate tool.

To update your device's MinSRev:

1. Download and extract the NVM Update Package for your device.
2. Use the command line to update your device's MinSRev:
    - Windows:

```
nvmupdatew64e -u -optinminsrev -l update.log -o
 results.xml -c nvmupdate.cfg
```

- Linux:

```
nvmupdate64e -u -optinminsrev -l update.log -o
results.xml -c nvmupdate.cfg
```

Where:

`-u`

Sets nvmupdate to update mode.

`-optinminsrev`

Tells the tool to update the MinSRev value.

-l update.log

Specifies the name of the log file.

-o results.xml

Specifies the name of the results file. This is an XML file that contains the inventory/update results.

-c nvmupdate.cfg

Specifies the name of the configuration file. This is a text file that contains descriptions of networking devices and firmware versions for those devices.

See Intel® Ethernet NVM Update Tool for more information on how to use the nvmupdate tool.

Firmware Rollback Mode

When a port is in firmware rollback mode, it may have reduced functionality.

Usually a device enters firmware rollback mode when a firmware update does not complete correctly. Rebooting or power cycling the system may allow the port to use the previous firmware.

You may need to reapply the firmware update to regain full functionality on the device. Use the appropriate NVM Update Package to update the device’s firmware. Download the latest NVM Update Package from your vendor’s support website and follow the instructions in it.

After restoring the NVM image, you may need to perform an A/C power cycle of the system.

Firmware Recovery Mode

A device will enter Firmware Recovery mode if it detects a problem that requires the firmware to be reprogrammed. When a device is in Firmware Recovery mode, it will not pass traffic or allow any configuration; you can only attempt to recover the device’s firmware.

Detecting Recovery Mode

During initialization, a device can enter recovery mode if the device firmware detects a problem with the LAN device, mandating NVM reprogramming to restore normal operation. After thorough internal testing of the NVM (typically less than 10 minutes, but in some cases longer), the NIC enters Recovery Mode.

Firmware Recovery Mode Errors and Messages

When a device is in Firmware Recovery mode, the device drivers, preboot software, and utilities may log or display messages such as the following:

“Firmware recovery mode detected. Limiting functionality. Refer to the Intel® Ethernet Adapters and Devices User Guide for details on firmware recovery mode.”

“Firmware recovery mode detected. The underlying hardware has been deactivated. Refer to the Intel® Ethernet Adapters and Devices User Guide for details on firmware recovery mode.”

“Firmware recovery mode detected. Initialization failed.”

“Firmware recovery mode detected. Limiting functionality.”

“Initialization failure due to repeated FW resets.”

The last message above is usually an indication that the device is about to enter Recovery Mode. The device may be able to return to normal functionality without intervention. This may take several minutes. No action is required unless the device does enter Recovery Mode.

Resolving Firmware Recovery Mode Issues

If your device is in Firmware Recovery mode, you can restore it to factory defaults using the latest NVM Update Package. Download the latest NVM Update Package from your vendor’s support website and follow the instructions in it.

The process for resolution of Firmware Recovery Mode Issues is outlined in the subsections below.

NVM Self Check

The process begins after power-on or reboot. At this time, the firmware will perform tests to assess whether there is damage or corruption of the device N VM image.

Actions:

- If NVM image damage or corruption **is not** detected, the device will initialize and operate normally. No further action is required.

- If NVM image damage or corruption **is** detected, the device will not initialize. Proceed with the additional recovery steps listed under Recovery Mode below.

Recovery Mode

The device NVM image has exhibited damage or corruption.

Actions:

1. Wait 10 minutes for the NVM self-check process to complete. If during this period normal operation is achieved, the device will initialize and operate normally. No further action is required.
2. If after 10 minutes normal operation is *not* achieved:
   a. Check the System Event log for Windows OSs or driver message and kernel logs for Linux and ESXi based distributions. Recovery Mode is confirmed by presence of message/log entries as listed in the Firmware Recovery Mode Errors and Messages section above.

   b. Reboot the system and proceed with the additional recovery steps listed under NVM Image Restoration below.

Note:

- While in Recovery Mode, for Windows OSs, clicking on the adapter in Device Manager may present a dialog box indicating that Firmware Recovery Mode is active.
  - Once the dialog is dismissed, while the device appears to be functioning normally, it is in fact limited to only enable NVM image recovery.
- If the system is rebooted (versus power cycled), the driver status may not show a Code 10/yellow bang status as expected. Refer to events logged in System Event log for Windows OSs or driver message and kernel logs for Linux and ESXi based distributions to accurately assess the adapter status.
- When the adapter is in recovery mode, the link LED will not be lit and the adapter will not appear in the following locations:
  - F2 System Setup > Device Settings
  - System BIOS as a NIC for PXE Boot in UEFI boot mode

NVM Image Restoration

At this point, the device is in Firmware Recovery mode and its functionality is limited to only supporting restoration of the NVM image.

Actions:

1. Before initiating device recovery, the integrity of the host operating system, device drivers and firmware utilities must be verified and reinstalled if necessary. Fully functional operating system, device drivers and tools are required for device recovery. Please consult your operating system specific

instructions on how to scan and repair potentially damaged system files.

2. If your device is in Firmware Recovery mode, you can restore it to factory defaults using the latest NVM Update Package. Download the latest NVM Update Package from your vendor's support website and follow the instructions in it.
3. After restoring the NVM image, perform an A/C power cycle of the system. Details for this are in the Other General Notes section below.

Note:

User configured settings (i.e. iSCSI target information, user-defined port/alternate MAC addresses) will not be restored to pre-recovery mode values.

Other General Notes

To perform an AC power cycle, do the following:

1. Shut down the system if it is powered up.
2. Unplug all AC power cords from the system.
3. Leave the AC power cords unplugged for 15 seconds to allow the system power supply to discharge completely.
4. Plug in AC power cords to the system.

Using Devlink to Update a Device's NVM

On a Linux system, when you update the NVM on some devices, the update may use the **devlink** interface, rather than the **ethtool** interface. This will happen if the following are true:

- You are updating an Intel Ethernet 800 Series device.
- Your system is running a distro that supports the `devlink dev flash` command.
- The firmware currently installed on the device supports it.
- The new NVM conforms to the correct PLDM format.

Most of the functionality and commands are the same with the following exceptions:

- You cannot update a device in Recovery Mode. (To update a device in recovery mode, you must download and install the Intel Ethernet driver set. See Recovery Mode for more information.)
- You cannot update the OROM or Netlist as a separate update, only as part of a full NVM update.
- If you specified a preservation level of `PRESERVE_ALL`, the system will immediately perform an EMPR reset after the NVM update.

On devices that support it, you can also use the **devlink** command line directly to update the device NVM:

```
devlink dev flash pci/0000:3b:00.0 file filename.bin
```

Where:

pci/0000:3b:00.0

The device you wish to update. You can get a list of devices with the `devlink dev info` command.

filename.bin

The file that contains the new NVM image.

However, we recommend that you use the NVMUpdate tool. See Intel® Ethernet NVM Update Tool for more information on this tool.

Firmware Logging

Intel® Ethernet 800 Series devices allow you to generate firmware logs for supported categories of events, to help debug complex issues with Customer Support. Firmware logging is enabled by default.

Note:

- Both the device and the driver need to support firmware logging for the functionality to work. If you are not able to set the configuration and the problem persists, reinstall the driver.
- You must have the latest base driver and NVM installed.
- Firmware logging events and levels are device-wide settings. Changing the log configuration on one port will apply it to all ports on a device.

Capturing a Firmware Log

To capture a firmware log, you must do the following:

1. Set the configuration for the firmware log. See later in this page for more information.
2. Perform the necessary steps to generate the issue you're trying to debug.
3. Capture the firmware log. (Exact steps will vary by operating system.)
4. Stop capturing the firmware log.
5. Reset your firmware log settings as needed.
6. Work with Customer Support to debug your issue.

Note:

Firmware logs are generated in a binary format and must be decoded by Customer Support. Information collected is related only to firmware and hardware for debug purposes.

Configuring Settings for Firmware Logs

Firmware logs capture information about different categories of events (called “modules”). A module corresponds to a general category of functionality, such as link topology detection or manageability.

The device’s NVM sets default verbosity levels for each module. You can change the verbosity level per module; refer to OS-Specific Information below for more details. You can set only one log level per module, and each level includes the verbosity levels lower than it.

Available verbosity levels are:

- 0 = none
- 1 = error
- 2 = warning
- 3 = normal
- 4 = verbose

If you see errors or suspect the issue could fall into the below categories, setting the firmware logs to capture more verbosity for the corresponding module(s) in the right column will provide more information in the firmware log.

Modules for Events

Category of Event	Corresponding Module
Initialization	Control
NVM	• NVM • Authentication • VPD
I/O	• I2C • SDP • MDIO

Category of Event	Corresponding Module
Link Management	• Link Management • Link Control Technology • Link Topology Detection
Rx	• Parser • Switch • ACL • Post
Tx	• Scheduler • Tx Queue Management
AQ Interface	• Admin Queue • HDMA
Manageability	Manageability
Protocols	• LLDP • DCBx
Infrastructure	• Watchdog • Task Dispatcher • General • IOSF • PF Registration • Module Versions
XLR	XLR
QoS	DCB

Category of Event	Corresponding Module
Diagnostics	• SyncE • Health
TimeSync	Time Sync

Tips for Firmware Logs

- Firmware logs are for the hardest issues to debug. If you are experiencing issues, refer to the following sections for preliminary methods to diagnose problems:
    - Firmware
    - Troubleshooting
    - Health Status Messages
- We generally do not recommend to capture firmware logs at all times. If you suspect an issue, set the module(s) for the suspected event to a higher verbosity level, capture the firmware log, and then stop the log.
- Collecting firmware logs should not materially impact performance or CPU utilization.
- In general, set the logging level to Verbose only for the configuration group(s) or module(s) you need to debug. Setting too many modules to Verbose can overrun the buffer.
- You can try writing logs to a remote location or an external storage device, if your disk is full or your system does not have sufficient storage.

OS-Specific Information

Linux

At a high level, do the following to capture a firmware log in Linux:

1. Set log levels. For example:

```
echo normal > /sys/kernel/debug/ice/0000:18:00.0/fw
log/modules/all
```

2. Turn on firmware logging:

```
echo 1 > /sys/kernel/debug/ice/0000:18:00.0/fwlog/e
nable
```

3. Perform the necessary steps to generate the issue you are trying to debug.
4. Turn off firmware logging:

   ```
   echo 0 > /sys/kernel/debug/ice/0000:18:00.0/fwlog/enable
   ```

5. Save data to a file:

   ```
   cat /sys/kernel/debug/ice/0000:18:00.0/fwlog/data > fwlog.bin
   ```

6. Work with Customer Support to debug your issue.

Refer to the README in the driver tarball for more information on configuring firmware logs.

FreeBSD

Refer to the README in the driver tarball for more information on configuring firmware logs.

> **Note:** In FreeBSD, the driver can register/unregister to receive events per PF.

Windows

In Windows, you use PowerShell and Intel Ethernet cmdlets to configure firmware logging and capture firmware logs.

At a high level, do the following to capture a firmware log in Windows:

1. Set the configuration for the firmware log, using the `Set-IntelEthernetLogConfig` cmdlet in PowerShell.
2. Perform the necessary steps to generate the issue you're trying to debug.
3. Start capturing the firmware log, using the `Start-IntelEthernetLog` cmdlet.
4. Stop capturing the firmware log, using the `Stop-IntelEthernetLog` cmdlet.
5. Work with Customer Support to decode your firmware log file and debug the issue.

> **Note:**
>
> - Firmware logs will be captured in the file you designated with `Start-IntelEthernetLog`.
> - To disable firmware logging, use the `Disable-IntelEthernetLogConfig` cmdlet. To verify that firmware logging is disabled, run the `Get-IntelEthernetLogConfig cmdlet`; its results should say "Disabled."

ESXi

In ESXi, use `esxcfg-module` to set the configuration for firmware logs. Firmware logs are printed to kernel logs, with the tag `FWLOG`; use dmesg or read the file at `/var/log/vmkernel.log`.

At a high level, do the following to capture a firmware log in ESXi:

> **Note:** Refer to ESXi Example Commands below for all commands and parameters.

1. Set the configuration for the firmware log, using `esxcfg-module`. The ESXi driver uses the following module parameters for firmware logging:

   - `FWLogEnable`: Enables firmware logging functionality on the designated PF (0 = Disable, 1 = Enable). Use commas to separate the values for each PF; the first value is for PF0, second for PF1, and so on.
   - `FWLogEvents`: Designates the firmware events to log, using a bitmask. Binary math is required to set.
   - `FWLogLevel`: Sets the verbosity level for the firmware event's log.

1. Redirect the kernel log or dmesg to a separate file for capturing the firmware log.
2. Reboot the system for changes to take effect.
3. After the system has rebooted, perform the necessary steps to generate the issue you're trying to debug.
4. Work with Customer Support to decode your firmware log file and debug the issue.

   > **Note:** Firmware logs will be captured in the file you designated in step 2.

ESXi Example Commands

Use the following commands in ESXi for tasks related to firmware logging:

- To enable firmware logging and set the verbosity level for your desired events:

  ```
  esxcfg-module icen -s 'FWLogEnable=<values> FWLogEvents=<bitmask> FWLogLevel=<value>'
  ```

  For example, to enable firmware logging on PF0 and set all events to log warning messages, use:

  ```
  esxcfg-module icen -s 'FWLogEnable=1,0,0,0,0,0,0,0 FWLogEvents=255 FWLogLevel=2'
  ```

- To show the current configuration of the firmware log parameters:

```
esxcfg-module -g <driver name>
```

Note:
If firmware logging is disabled, the `FWLogEnable` parameter should say "0" (disabled).

- To show a description of module parameters for firmware logging:

```
esxcfg-module -i <driver name>
```

Note: Look for the parameters that begin with `FWLog`.

- To redirect the firmware log to a file:

```
tail -f /var/log/vmkernel.log > filename.log
```

- To disable firmware logging:

```
esxcfg-module icen -s 'FWLogEnable=0 FWLogEvents=0
FWLogLevel=0'
```

Debug Dump

Some Intel Ethernet 800 Series devices support debug dump, which allows you to obtain runtime register values from the firmware for "clusters" of events and then write the results to a single dump file, for debugging complicated issues in the field.

This debug dump contains a snapshot of the device and its existing hardware configuration, such as switch tables, transmit scheduler tables, and other information. Debug dump captures the current state of the specified cluster(s) and is a stateless snapshot of the whole device.

Note:

- The contents of the debug dump are not human-readable. You must work with Customer Support to decode the file.
- Debug dump is per device, not per PF.
- Debug dump writes all information to a single file.

Exact steps will vary by OS, but do the following to generate a debug dump log file:

1. Using the method appropriate for your OS (see OS-Specific Information below), specify the clusters for which you want to dump the hardware configuration. Supported clusters will vary by OS and hardware family.
2. Specify the path and filename for the dump file to be written to (optional depending on your OS).
3. Execute the command to write the debug dump file.
4. After the log file is written, work with Customer Support to decode the dump file.

OS-Specific Information

Use the following tools or commands to write the debug dump results to a dump file.

- Windows Server, Azure Stack HCI: Use either of the following:
  - `Write-IntelEthernetDebugDump` Ethernet cmdlet in PowerShell (see the cmdlet help for more information)
  - Intel® Ethernet Inspector
- Windows: Not supported
- Linux: Use **debugfs** (see the Linux base driver README for more information)
- ESXi: Use **esxcli** (see below)
- FreeBSD: Use **sysctl** (see the FreeBSD base driver README for more information)

Note:

- 810 Series devices support debug dump on the operating systems listed above.
- 820 Series devices only support debug dump on Linux.

ESXi

Note:

For this functionality to work, you must have installed version 1.10.x or higher of the intnet tool, which is a plugin to the esxcli tool. You can download the latest version from the Intel Download Center here.

In esxcli, use the following command to generate the debug dump file for your specified cluster(s):

```
esxcli intnet debug fw dump <Cmd options>
```

Where `<Cmd options>` are:

-n, --vmnic *<string>*

Specifies the vmnic name to operate on. This field is required.

-c, --clusters *<string>*

[Optional] Specifies the clusters to dump.

To specify multiple clusters, enclose a single string in quotes, separated by commas with no spaces. For example:

```
esxcli intnet debug fw dump -n vmnic0 --clusters "ACL,L2P"
```

If `-c` is not specified, the driver dumps all clusters.

-l, --list

Displays the complete list of valid clusters on the screen.

To show the complete list of valid clusters, use the following:

```
esxcli intnet debug fw dump -n <vmnicX> -l
```

esxcli will output the debug dump results to a single file in the `/scratch/core` directory. The file naming convention is *vmnicX-<time-stamp>-dump.bin*, where `vmnicX` is the VMware device alias of the affected device.

Firmware Link Layer Discovery Protocol (FW-LLDP)

Devices based on the Intel® Ethernet 800 and 700 Series use a Link Layer Discovery Protocol (LLDP) agent that runs in the firmware. When it is running, it prevents the operating system and applications from receiving LLDP traffic from the network adapter.

- The FW-LLDP setting is per port and persists across reboots.
- The FW-LLDP Agent is required for Data Center Bridging (DCB) to function.

Adapters Based on the Intel® Ethernet 800 Series

FW-LLDP is disabled in NVM by default. To enable/disable the FW-LLDP Agent:

- Linux: Use ethtool to persistently set or show the `fw-lldp-agent` private flag.
- FreeBSD: Use sysctl to persistently set or show the `fw_lldp_agent` flag.
- ESX: Use the esxcli command to persistently set or get the `fw-lldp-agent` setting.

- Microsoft Windows: The base driver does not persistently change FW-LLDP. Use the LLDP Agent attribute in UEFI HII to persistently change the FW-LLDP setting. If you enable DCB when FW-LLDP is disabled, the base driver temporarily starts the LLDP Agent while DCB functionality is enabled.

Adapters Based on the Intel® Ethernet 700 Series

FW-LLDP is enabled in NVM by default. To enable/disable the FW-LLDP Agent:

- Linux: Use ethtool to set or show the `disable-fw-lldp` private flag.
- FreeBSD: Use sysctl to set or show the `fw_lldp` flag.
- ESX: Use the esxcfg-module command to set or get the `LLDP` module parameter.
- Microsoft Windows: Use the LLDP Agent attribute in UEFI HII to change the FW-LLDP setting. Note: You must enable the UEFI HII `LLDP AGENT` attribute for the FW-LLDP setting to take effect. If "LLDP AGENT" is set to disabled in UEFI HII, you cannot enable FW-LLDP from the OS.
- You must enable the LLDP Agent from UEFI HII to use DCB.

Flow Control

This feature enables adapters to more effectively regulate traffic. Adapters generate flow control frames when their receive queues reach a pre-defined limit. Generating flow control frames signals the transmitter to slow transmission. Adapters respond to flow control frames by pausing packet transmission for the time specified in the flow control frame.

By enabling adapters to adjust packet transmission, flow control helps prevent dropped packets. You may improve RDMA performance by enabling flow control on all nodes and on the switch to which they connect.

Note:

- For adapters to benefit from this feature, link partners must support flow control frames.
- On systems running a Microsoft Windows Server operating system, enabling *QoS/priority* flow control will disable link level flow control.
- Some devices support Auto Negotiation. Selecting this will cause the device to advertise the value stored in its NVM (usually "Disabled").

To change this setting in Intel PROsec:

This setting is found on the PROset tab of the device's Device Manager property sheet or in the Intel PROset Adapter Configuration Utility.

Possible values for this setting are:

- Disabled

- RX Enabled
- TX Enabled
- Auto-neegotiation (only available in some adapters)

Forward Error Correction (FEC) Mode

This feature allows you to set the Forward Error Correction (FEC) mode. FEC improves link stability, but increases latency. Many high quality optics, direct attach cables, and backplane channels provide a stable link without FEC.

The driver allows you to set the following FEC Modes:

- **Auto FEC**: Sets the FEC Mode based on the capabilities of the attached cable.
- **CL108 RS-FEC**: Selects only RS-FEC ability and request capabilities.
- **CL74 FC-FEC/BASE-R**: Selects only BASE-R ability and request capabilities.
- **No FEC**: Disables FEC.

Note:

- For devices to benefit from this feature, link partners must have FEC enabled.
- Intel® Ethernet 800 Series devices only enable Forward Error Correction (FEC) configurations that are supported by the connected media and which are expected to yield healthy Bit Error Rate (BER) connections.

  If you enable the registry keyword `AllowNoFECModulesInAuto`, Auto FEC negotiation will include *No FEC* in case your link partner does not have FEC enabled or is not FEC-capable.

  - To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

    ```
    Set-IntelNetAdapterSetting -Name "<adapter_name>" -RegistryKeywordAllowNoFECModulesInAuto -RegistryValue 1
    ```

  - To change this setting in Linux, use ethtool. For example:

    ```
    ethtool --set-priv-flags <ethX> allow-no-fec-modules-in-auto on
    ```

- If you are having link issues (including no link) at link speeds faster than 10Gbps, check your switch configuration and/or specifications. Many optical connections and direct attach cables require RS-FEC for connection speeds faster than 10Gbps. One of the following may resolve the issue:
  - Configure your switch to use RS-FEC mode.

- Specify a 10Gbps, or slower, link speed connection.
- If you are attempting to connect at 25 Gbps, try using an SFP28 CA-S or CS-N Direct Attach cable. These cables do not require RS-FEC.
- If your switch does not support RS-FEC mode, check with your switch vendor for the availability of a SW or FW upgrade.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "FEC Mode" -DisplayValue "Auto FEC"
```

Gigabit PHY Mode

This feature determines whether an adapter or link partner is designated as the primary. The other device is designated as the secondary. By default, the IEEE 802.3ab specification defines how conflicts are handled. Multi-port devices, such as switches have higher priority over single port devices and are assigned as the primary. If both devices are multi-port devices, the one with higher seed bits becomes the primary.

The default setting is called *Hardware Default*. Setting the value to any other than the default will override the hardware default.

Note:

When Gigabit PHY Mode is forced to Primary mode on both the Intel adapter and its link partner, the link speed obtained by the Intel adapter may be lower than expected or link may not be established.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Gigabit PHY Mode"-DisplayValue "Auto Detect"
```

Possible values for this setting are:

- Force Primary Mode
- Force Secondary Mode
- Auto-Detect

Note:

When Gigabit PHY Mode is forced to Primary mode on both the Intel adapter and its link partner, the link speed obtained by the Intel adapter may be lower than expected or link may not be established.

Intel® Ethernet Flow Director

The Intel® Ethernet Flow Director (Intel® Ethernet FD) performs the following tasks:

- Directs receive packets according to their flows to different queues
- Enables tight control on routing a flow in the platform
- Matches flows and CPU cores for flow affinity
- Depending on the device family: Supports multiple parameters for flexible flow classification and load balancing (in SFP mode only)

Depending on the driver and device family, the driver might support the following flow types:

- IPv4
- TCPv4
- UDPv4
- SCTPv4
- IPv6
- TCPv6
- UDPv6
- SCTPv6

Each flow type supports valid combinations of IP addresses (source or destination) and UDP/TCP ports (source and destination). You can supply only a source IP address, a source IP address and a destination port, or any combination of one or more of these four parameters.

The following table summarizes supported Intel Ethernet Flow Director features across Intel Ethernet controllers.

Feature	500 Series	700 Series	800 Series

Feature	500 Series	700 Series	800 Series
VF Flow Director	Supported	Routing to VF not supported	Not supported
IP Address Range Filter	Supported	Not supported	Field masking
IPv6 Support	Supported	Supported	Supported
Configurable Input Set	Configured per port	Configured globally	Configured per port
ATR	Supported	Supported	Not supported
Flex Byte Filter	Starts at beginning of packet	Starts at beginning of payload	Starts at beginning of packet
Tunneled Packets	Filter matches outer header	Filter matches inner header	Filter matches inner header

See the Linux driver READMEs for more information on configuring this feature.

Interrupt Moderation Rate

This feature sets the Interrupt Throttle Rate (ITR). This setting moderates the rate at which Transmit and Receive interrupts are generated.

When an event such as packet receiving occurs, the adapter generates an interrupt. The interrupt interrupts the CPU and any application running at the time, and calls on the driver to handle the packet. At greater link speeds, more interrupts are created, and CPU rates also increase. This results in poor system performance. When you use a higher ITR setting, the interrupt rate is lower and the result is better CPU performance.

Note:

A higher ITR rate also means that the driver has more latency in handling packets. If the adapter is handling many small packets, it is better to lower the ITR so that the driver can be more responsive to incoming and outgoing packets.

Altering this setting may improve traffic throughput for certain network and system configurations, however the default setting is optimal for common network and system configurations. Do not change this setting without verifying that the desired change will have a positive effect on network performance.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "Interrupt Moderation Rate"-DisplayValue "Adaptive"
```

Possible values for this setting are:

- Adaptive
- Extreme
- High
- Medium
- Low
- Minimal
- Off

Jumbo Frames

The Jumbo Frames feature enables or disables Jumbo Packet capability. The standard Ethernet frame size is about 1514 bytes, while Jumbo Packets are larger than this. Jumbo Packets can increase throughput and decrease CPU utilization. However, additional latency may be introduced.

Enable Jumbo Packets only if **all** devices across the network support them and are configured to use the same frame size. When setting up Jumbo Packets on other network devices, be aware that network devices calculate Jumbo Packet sizes differently. Some devices include the frame size in the header information while others do not. Intel adapters do not include frame size in the header information.

Jumbo Packets can be implemented simultaneously with VLANs and teaming. If a team contains one or more non-Intel adapters, the Jumbo Packets feature for the team is not supported. Before adding a non-Intel adapter to a team, make sure that you disable Jumbo Packets for all non-Intel adapters using the software that was shipped with the adapter.

Restrictions

- Jumbo frames are not supported in multi-vendor team configurations.

- Supported protocols are limited to IP (TCP, UDP).
- Jumbo frames require compatible switch connections that forward Jumbo Frames. Contact your switch vendor for more information.
- When standard-sized Ethernet frames (64 to 1518 bytes) are used, there is no benefit to configuring Jumbo Frames.
- The Jumbo Packets setting on the switch must be set to at least 8 bytes larger than the adapter setting for Microsoft Windows operating systems, and at least 22 bytes larger for all other operating systems.
- Jumbo Frames are not supported over Intel ANS VLANs under Microsoft Windows 10.

Note:

- End-to-end hardware must support this capability; otherwise, packets will be dropped.
- Intel adapters that support Jumbo Packets have a frame size limit of 9238 bytes, with a corresponding MTU size limit of 9216 bytes.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in the Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Jumbo Packet"-DisplayValue "4088 Bytes"
```

Possible values for this setting are:

- Disabled (1514 bytes)
- 4088 Bytes
- 9014 Bytes

Note: Set the switch 4 bytes higher for CRC, plus 4 bytes if using VLANs.

Link State on Interface Down

This feature sets if link is enabled or disabled when the interface is brought down. If this is set to **Disabled** and you bring an interface down (using an administrative tool, or in another way), then the port will lose its link. This allows an attached switch to detect that the interface is no longer up. However, if Wake on LAN or manageability is enabled on this port, the link will remain up.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "Link State onInterface Down" -DisplayValue "Enabled"
```

Possible values for this setting are:

- Enabled
- Disabled

Locally Administered Address

The Locally Administered Address overrides the initial MAC address with a user-assigned MAC address. To enter a new network address, type a 12-digit hexadecimal number in this box.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "LocallyAdministered Address" -DisplayValue "<desired address>"
```

Possible values for this setting are:

- 0000 0000 0001 - FFFF FFFF FFFD

Note: Exceptions:

- Do not use a multicast address (Least Significant Bit of the high byte = 1). For example, in the address 0Y123456789A, *Y* cannot be an odd number. (*Y* must be 0, 2, 4, 6, 8, A, C, or E.)
- Do not use all zeros or all Fs.
-

If you do not enter an address, the address is the original network address of the adapter. For example:

```
Multicast: 0123 4567 8999 Broadcast: FFFF FFFF FF
FFUnicast (legal): 0070 4567 8999
```

Note: In a team, Intel PROSet uses either of the following:

- The primary adapter’s permanent MAC address if the team does not have an LAA configured
- The team’s LAA if the team has an LAA configured

Intel PROSet does not use an adapter’s LAA if the adapter is the primary adapter in a team and the team has an LAA.

Log Link State Event

This setting is used to enable/disable the logging of link state changes. If enabled, a link-up change event or a link-down change event generates a message that is displayed in the system event logger. This message contains the link’s speed and duplex. Administrators view the event message from the system event log.

The following events are logged:

- The link is up.
- The link is down.
- Mismatch in duplex.
- Spanning Tree Protocol detected.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in the Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Log LinkState Event" -DisplayValue "Enabled"
```

Possible values for this setting are:

- Enabled

- Disabled

Low Latency Interrupts

Low Latency Interrupts (LLI) enable the network device to bypass the configured interrupt moderation scheme based on the type of data being received. LLI configures which arriving TCP packets trigger an immediate interrupt, enabling the system to handle the packet more quickly. Reduced data latency enables some applications to gain faster access to network data.

Note: When LLI is enabled, system CPU utilization may increase.

LLI can be used for data packets that contain a TCP PSH flag in the header or for specified TCP ports:

- **Packets with TCP PSH Flag**: Any incoming packet with the TCP PSH flag will trigger an immediate interrupt. The PSH flag is set by the sending device.
- **TCP Ports**: Every packet received on the specified ports will trigger an immediate interrupt. Up to eight ports may be specified.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "Low LatencyInterrupts" -DisplayValue "Port-Based"
```

Possible values for this setting are:

- Disable
- PSH Flag-Based
- Port-Based

Malicious Driver Detection (MDD) for VFs

Some Intel Ethernet devices use Malicious Driver Detection (MDD) to detect malicious traffic from the VF and disable Tx/Rx queues or drop the offending packet until a VF driver reset occurs. You can view MDD messages in the PF’s event log.

- If the device supports automatic VF resets and the driver detects an MDD event on the receive path, the PF will automatically reset the VF and reenable queues. If automatic VF resets are disabled, the PF will not automatically reset the VF when it detects MDD events. See the table below for supported MDD features.
- If the PF driver logs MDD events from the VF, confirm that the correct VF driver is installed.
- To restore functionality, you can manually reload the VF or VM or, if supported by the device, enable automatic VF resets.

The following table shows MDD capabilities by device family:

- Intel Ethernet 800 Series
- Intel Ethernet 700 Series
- Intel Ethernet 500 Series
- Intel I350 Gigabit Network Connection

Feature	**800 Series**	**700 Series**	**500 Series**	**I350**
Automatically resets the VF and reenables queues after MDD events	If enabled	If enabled	Yes	Yes
Can disable automatic VF reset after MDD events	Yes	Yes	No	No

MDD Auto Reset VFs

Automatically resets the virtual machine immediately after the adapter detects a Malicious Driver Detection (MDD) event on the receive path.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "MDD Auto Reset VFs"-DisplayValue "Enabled"
```

Possible values for this setting are:

- Enabled
- Disabled

Max Number of RSS Queues Per Vport

This setting sets the maximum number of Receive Side Scaling (RSS) queue pairs per VF.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Max Number ofRSS Queues Per Vport" -DisplayValue
 "4 Queues"
```

Possible values for this setting are:

- 2 Queues
- 4 Queues
- 8 Queues
- 16 Queues

Offloads

- IPv4 Checksum Offload
- Large Send Offload (IPv4 and IPv6)
- NVGRE Encapsulated Task Offload
- QoS Offload
- TCP Checksum Offload (IPv4 and IPv6)
- UDP Checksum Offload (IPv4 and IPv6)
- UDP Segmentation Offload (IPv4 and IPv6)
- VXLAN Encapsulated Task Offload

In addition to the offloads listed above, see the following pages for related information:

- Priority and VLAN Tagging

- Virtual Machine Queue Offloading

IPv4 Checksum Offload

This setting allows the adapter to compute the IPv4 checksum of incoming and outgoing packets. This feature enhances IPv4 receive and transmit performance and reduces CPU utilization.

With Offloading off, the operating system verifies the IPv4 checksum.

With Offloading on, the adapter completes the verification (on RX) and computation (on TX) for the operating system.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "IPv4 Checksum Offload"-DisplayValue "Tx Enabled"
```

Possible values for this setting are:

- Disabled
- RX Enabled
- TX Enabled
- RX & TX Enabled

Large Send Offload (IPv4 and IPv6)

This setting sets the adapter to offload the task of segmenting TCP messages into valid Ethernet frames. The maximum frame size limit for large send offload is set to 64,000 bytes.

Since the adapter hardware is able to complete data segmentation much faster than operating system software, this feature may improve transmission performance. In addition, the adapter uses fewer CPU resources.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Large Send Offload V2(IPv4)" -DisplayValue "Enab
led"
```

Possible values for this setting are:

- Enabled
- Disabled

NVGRE Encapsulated Task Offload

Network Virtualization using Generic Routing Encapsulation (NVGRE) increases the efficient routing of network traffic within a virtualized or cloud environment. Some Intel Ethernet Network devices perform NVGRE processing, offloading it from the operating system. This reduces CPU utilization.

To change this setting in Intel PROSet:

This setting is found in the Adapter Settings panel in Intel PROSet ACU. On the device’s Device Manager property sheet, it is found on the Advanced tab, under the Offloading Options > Encapsulated Task Offload setting.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "NVGREEncapsulated Task Offload" -DisplayValue "E
nabled"
```

Possible values for this setting are:

- Enabled
- Disabled

QoS Offload

This setting configures the Quality of Service (QoS) offload for the miniport adapter. This feature allows you to set a bandwidth cap and reservation to one or more virtual machines on a physical device, including both software VMs and SR-IOV interfaces.

To change this setting in Intel PROSet:

This setting is found on the Adapter tab and in the Adapter Settings panel of the Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "QoS Offload"-DisplayValue "Enabled"
```

Possible values for this setting are:

- Enabled
- Disabled

TCP Checksum Offload (IPv4 and IPv6)

This setting allows the adapter to verify the TCP checksum of incoming packets and compute the TCP checksum of outgoing packets. This feature enhances receive and transmit performance and reduces CPU utilization.

With Offloading off, the operating system verifies the TCP checksum.

With Offloading on, the adapter completes the verification for the operating system.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "TCP ChecksumOffload (IPv4)" -DisplayValue "Tx En
abled"
```

Possible values for this setting are:

- Disabled
- RX Enabled
- TX Enabled
- RX & TX Enabled

UDP Checksum Offload (IPv4 and IPv6)

This setting allows the adapter to verify the UDP checksum of incoming packets and compute the UDP checksum of outgoing packets. This feature enhances receive and transmit performance and reduces CPU utilization.

With Offloading off, the operating system verifies the UDP checksum.

With Offloading on, the adapter completes the verification for the operating system.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "UDP Checksum Offload(IPv4)" -DisplayValue "Tx Enabled"
```

Possible values for this setting are:

- Disabled
- RX Enabled
- TX Enabled
- RX & TX Enabled

UDP Segmentation Offload (IPv4 and IPv6)

This setting allows the adapter to segment UDP packets with payloads up to 64K into valid Ethernet frames. Because the adapter hardware is able to complete data segmentation much faster than operating system software, this feature may improve transmission performance. In addition, the adapter may use fewer CPU resources.

With Offloading off, the operating system segments UDP packets into valid Ethernet frames.

With Offloading on, the adapter segments UDP packets for the operating system.

Note: UDP Segmentation Offload requires:

- Microsoft* Windows Server* 2019, Version 1903, or later
- Linux* kernel 4.18, or later

To change this setting in Intel PROSet:

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "UDP SegmentationOffload (IPv4)" -DisplayValue "Enabled"
```

Possible values for this setting are:

- Enabled
- Disabled

VXLAN Encapsulated Task Offload

Virtual Extensible LAN (VXLAN) allows you to extend an L2 network over an L3 network, which may be useful in a virtualized or cloud environment. Some Intel Ethernet devices perform VXLAN processing, offloading it from the operating system. This reduces CPU utilization.

VXLAN may be useful in multi-tenant environments such as cloud service providers where the number of VLANs exceeds the 4094 limit imposed by the 12-bit VLAN ID used in Ethernet data frames.

To change this setting in Intel PROSet:

This setting is found in the Adapter Settings panel in Intel PROSet ACU. On the device's Device Manager property sheet, it is found on the Advanced tab, under the Offloading Options > Encapsulated Task Offload setting.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "VXLAN EncapsulatedTask Offload" -DisplayValue "Enabled"
```

Possible values for this setting are:

- Enabled
- Disabled

Performance Options

Optimizing Performance

You can configure advanced settings on Intel network adapters to help optimize server performance. This section provides tips for:

- General Optimization
- Optimization for Specific Usage Models

Note:

- Linux users: See the README file in the Linux driver package for Linux-specific performance enhancement details.
- The recommendations below are guidelines and should be treated as such. Additional factors such as installed applications, bus type, network topology, and operating system also affect system performance.
- These adjustments should be performed by a highly skilled network administrator. They are not guaranteed to improve performance. Not all settings shown here may be available through network driver configuration, operating system or system BIOS.
- When using performance test software, refer to the documentation of the application for optimal results.

General Optimization

- Install the adapter in an appropriate slot.

  Note:
  Some PCIe x8 slots are actually configured as x4 slots. These slots have insufficient bandwidth for full line rate with some dual port devices. The driver can detect this situation and will write the following message in the system log: “PCI-Express bandwidth available for this card is not sufficient for optimal performance. For optimal performance a x8 PCI-Express slot is required.” If this error occurs, moving your adapter to a true x8 slot will resolve the issue.

- For an Intel Ethernet 700 Series adapter to reach its full potential, you must install it in a PCIe Gen3 x8 slot. Installing it in a shorter slot, or a Gen2 or Gen1 slot, will impact the throughput the adapter can attain.
- Use the proper cabling for your device.
- Increase the number of TCP and Socket resources from the default value. For Windows based systems, we have not identified system parameters other than the TCP Window Size which significantly impact performance.
- Increase the allocation size of Driver Resources (transmit/receive buffers). However, most TCP traffic patterns work best with the transmit buffer set to its default value, and the receive buffer set to its minimum value.

Jumbo Frames

Enabling jumbo frames may increase throughput. You must enable jumbo frames on all of your network components to get any benefit.

RSS Queues

If you have multiple 10 Gbps (or faster) ports installed in a system, the RSS queues of each adapter port can be adjusted to use non-overlapping sets of processors within the adapter’s local Non-Uniform Memory Access (NUMA) Node/Socket. Change the RSS Base Processor Number for each adapter port so that the combination of the base processor and the max number of RSS processors settings ensure non-overlapping cores.For Microsoft Windows systems, do the following:

1. Identify the adapter ports to be adjusted and inspect their RssProcessorArray using the `Get-NetAdapterRSS` PowerShell cmdlet.
2. Identify the processors with NUMA distance 0. These are the cores in the adapter’s local NUMA Node/Socket and will provide the best performance.
3. Adjust the RSS Base processor on each port to use a non-overlapping set of processors within the local set of processors. You can do this manually or using the following PowerShell command:

   ```
   Set-NetAdapterAdvancedProperty -Name <Adapter Name>
    -DisplayName "RSS BaseProcessor Number" -DisplayVa
   lue <RSS Base Proc Value>
   ```

4. Use the `Get-NetAdapterAdvancedproperty` cmdlet to check that the right values have been set:

   ```
   Get-NetAdapterAdvancedproperty -Name <Adapter Name>
   ```

For example: For a 4-port adapter with Local processors 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, and `Max RSS processor` of 8, set the RSS base processors to 0, 8, 16 and 24.

CPU Affinity

When passing traffic on multiple network ports using an I/O application that runs on most or all of the cores in your system, consider setting the CPU Affinity for that application to fewer cores. This should reduce CPU utilization and in some cases may increase throughput for the device. The cores selected for CPU Affinity must be local to the affected network device’s Processor Node/Group. You can use the PowerShell command `Get-NetAdapterRSS` to list the cores that are local to a device. You may need to increase the number of cores assigned to the application to maximize throughput. Refer to your operating system documentation for more details on setting the CPU Affinity.

Optimization for Specific Usage Models

The following sections describe possible tasks you can try to optimize performance for specific server usage models.

Optimize for Quick Response and Low Latency

Useful for: Video, audio, and High Performance Computing Cluster (HPCC) servers

- Minimize or disable interrupt moderation rate.
- Disable offload TCP segmentation.
- Disable jumbo packets.
- Increase transmit descriptors.
- Increase receive descriptors.
- Increase RSS queues.

Optimize for Throughput

Useful for: Data backup/retrieval and file servers

- Enable jumbo packets.
- Increase transmit descriptors.
- Increase receive descriptors.
- On systems that support NUMA, set the Preferred NUMA Node on each adapter to achieve better scaling across NUMA nodes.

Optimize for CPU Utilization

Useful for: Application, web, mail, and database servers

- Maximize interrupt moderation rate.
- Keep the default setting for the number of receive descriptors; avoid setting large numbers of receive descriptors.
- Decrease RSS queues.
- In Hyper-V environments, decrease the max number of RSS CPUs.

Tuning Performance with SR-IOV

When SR-IOV is enabled in Hyper-V, the following steps can help to improve performance between VM to VM and VM to Host.

From host OSes on both Host 1 and Host 2:

1. Enable RSS on the PF and vSwitch:

   ```
   Enable-NetAdapterRss -name "ADAPTER_NAME"
   ```

2. Enable 4 queues per VF:

```
Set-VMNetworkAdapter -VMName "YOUR_TEST_VM_NAME" -I
ovQueuePairsRequested 4Get-VmNetworkAdapter -VMName
 * | where {$_.SwitchName -eq "YOUR_TEST_SWITCH_NAM
E"} |  Set-VmNetworkAdapter -IovQueuePairsRequested
 4
```

3. Ensure the VMs have at least twice as many vCPUs as RSS queues. In this case, set the number of total processors in the VM to 8. To do this:
   a. Turn off the VM.
   b. In the VM, click **Settings**.
   c. Under **Hardware**, select **Processor**.
   d. Change the value of **Number of virtual processors** to 8.
   e. Apply the change.
4. For Windows Server 2022, issue the following command while the VM is in the off state:

```
Set-VMProcessor -VMName "YOUR_VM_Name" -HwThreadCou
ntPerCore 1 -Count 8
```

In both guest OSes VM1 and VM2:

1. Set RSS queues to 4 for all VFs in the guest OSes:

```
Set-NetAdapterRss -InterfaceDescription *adaptive*
-NumberOfReceiveQueues 4
```

2. Update the number of queues in the guest OS:

```
Set-NetAdapterAdvancedProperty -Name "your_adapter_
name_from_guest_os" -DisplayName"Maximum Number of
RSS Queues" -DisplayValue "8 Queues"
```

Note:

In the locations where there are settings for the number of queues, that value can be anything from 1 to 16. If you want more total throughput, increase the number of queues. When updating the number of queues, you **must** set `IovQueuePairsRequested` to a value that is equal to or greater than the number of queues you want to use in the VM.

Transmit Balancing

Some Intel® Ethernet 800 Series devices allow you to enable a transmit balancing feature to improve transmit performance under certain conditions. When the feature is enabled, you should experience more consistent transmit performance across queues and/or PFs and VFs.

By default, transmit balancing is disabled in the NVM. To enable this feature, use one of the following to persistently change the setting for the device:

- Use the Ethernet Port Configuration Tool (EPCT) to enable the `tx_balancing` option. Refer to Ethernet Port Configuration Tool (EPCT) for more information.
- Enable the Transmit Balancing device setting in UEFI HII.
- Enable transmit balancing via Linux devlink. Refer to the Linux readme inside the driver tarball for more information.

When the driver loads, it reads the transmit balancing setting from the NVM and configures the device accordingly.

Note:

- The user selection for transmit balancing in EPCT, HII, or Linux devlink is persistent across reboots. You must reboot the system for the selected setting to take effect.
- This setting is device wide.
- The driver, NVM, and DDP package must all support this functionality to enable the feature.

Performance Profile

Performance Profiles are supported on Intel® 10GbE adapters and allow you to quickly optimize the performance of your Intel® Ethernet Adapter. Selecting a performance profile will automatically adjust some Advanced Settings to their optimum setting for the selected application. For example, a standard server has optimal performance with only two RSS (Receive-Side Scaling) queues, but a web server requires more RSS queues for better scalability.

To change this setting in Intel PROSet:

You must install Intel PROSet to use Performance Profiles.

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Profile" -DisplayValue "Standard Server"
```

Possible values for this setting are:

- Standard Server: This profile is optimized for typical servers.
- Web Server: This profile is optimized for IIS and HTTP-based web servers.
- Virtualization Server: This profile is optimized for Microsoft's Hyper-V virtualization environment.
- Storage Server: This profile is optimized for Fibre Channel over Ethernet or for iSCSI over DCB performance. Selecting this profile will disable SR-IOV and VMQ.
- Storage + Virtualization: This profile is optimized for a combination of storage and virtualization requirements.
- Low Latency: This profile is optimized to minimize network latency.

Note:

- Not all options are available on all adapter/operating system combinations.
- If you have selected the Virtualization Server profile or the Storage + Virtualization profile, and you uninstall the Hyper-V role, you should select a new profile.

Teaming Considerations

When you create a team with all members of the team supporting Performance Profiles, you will be asked which profile to use at the time of team creation. The profile will be synchronized across the team. If there is not a profile that is supported by all team members then the only option will be Use Current Settings. The team will be created normally. Adding an adapter to an existing team works in much the same way.

If you attempt to team an adapter that supports performance profiles with an adapter that doesn't, the profile on the supporting adapter will be set to Custom Settings and the team will be created normally.

Note:

This feature is not configurable through Intel PROSet ACU. On Microsoft Windows Server 2019, Microsoft Windows* 10 Version 1809, and later, use Windows PowerShell.

See Adapter Teaming for more information on teams.

Power Options

The Power Management tab in the device’s Device Manager property sheet or the Adapter Settings panel in Intel PROSet ACU includes several settings that control the device’s power consumption. For example, you can set the adapter to reduce its power consumption if the cable is disconnected.

See the following for more information on power options:

- Wake on LAN (WoL) Options
  - WoL Supported Devices
- Other Power Options
  - Wake from S0ix on Magic Packet
  - Reduce Power if Cable Disconnected & Reduce Link Speed During Standby
  - Ultra Low Power Mode When Cable is Disconnected
  - Selective Suspend
  - Selective Suspend Idle Timeout
  - Energy Efficient Ethernet

ACPI Power States

Advanced Configuration and Power Interface (ACPI) supports a variety of power states. Each state represents a different level of power, from fully powered up to completely powered down, with partial levels of power in each intermediate state.

The ACPI power states are:

S0:

On and fully operational

S1:

System is in low-power mode (sleep mode). The CPU clock is stopped, but RAM is powered on and being refreshed.

S2:

Similar to S1, but power is removed from the CPU.

S3:

Suspend to RAM (standby mode). Most components are shut down. RAM remains operational.

S4:

Suspend to disk (hibernate mode). The memory contents are swapped to the disk drive and then reloaded into RAM when the system is awakened.

S5:

Power off

Microsoft Windows Server is ACPI-capable. It does not support waking from a power-off (S5) state, only from standby (S3) or hibernate (S4). When shutting down the system, these states shut down ACPI devices, including Intel Ethernet adapters. This disarms the adapter's remote wake-up capability. However, in some ACPI-capable computers, the BIOS may have a setting that allows you to override the operating system and wake from an S5 state anyway. If there is no support for wake from S5 state in your BIOS settings, you are limited to Wake From Standby when using these operating systems in ACPI computers.

Wake on LAN (WoL) Options

The ability to remotely wake computers is an important development in computer management. This feature has evolved from a simple remote power-on capability to a complex system interacting with a variety of device and operating system power states.

The Intel PROSet Power Management tab or the Adapter Settings panel in Intel PROSet ACU includes Wake on Magic Packet and Wake on Directed Packet settings. These control the type of packets that wake up the system from standby.

For some adapters, the Power Management tab in Intel PROSet or the Adapter Settings panel in Intel PROSet ACU includes a setting called Wake on Magic Packet from power off state. Enable this setting to explicitly allow wake-up with a Magic Packet from shutdown under APM power management mode.

Note:

- To use the Wake on Directed Packet feature, WoL must first be enabled in the EEPROM using BootUtil. See Intel® Ethernet Flash Firmware Utility for more information on BootUtil.
- If Reduce speed during standby is enabled, then Wake on Magic Packet and/or Wake on directed packet must be enabled. If both of these options are disabled, power is removed from the adapter during standby.
- Wake on Magic Packet from power off state has no effect on this option.

WoL Supported Devices

All devices support Wake on LAN on all ports, with the following exceptions.

Intel Ethernet 800 Series

The following devices do not support WoL:

- Intel® Ethernet Network Adapter E810-2C-Q2
- Intel® Ethernet Network Adapter E810-C-Q2
- Intel® Ethernet Network Adapter E810-C-Q2T
- Intel® Ethernet Network Adapter E810-XXV-4
- Intel® Ethernet Network Adapter E810-XXV-4T

- Intel® Ethernet Network Adapter E810-XXV-2

Intel Ethernet 700 Series

The following devices do not support WoL:

- Intel® Ethernet Network Adapter I710-T4L for OCP 3.0
- Intel® Ethernet Network Adapter I710-T4L
- Intel® Ethernet Converged Network Adapter X710-2
- Intel® Ethernet Converged Network Adapter X710-4
- Intel® Ethernet Converged Network Adapter X710-T4
- Intel® Ethernet Converged Network Adapter X710
- Intel® Ethernet Converged Network Adapter XL710-Q1
- Intel® Ethernet Converged Network Adapter XL710-Q2
- Intel® Ethernet Network Adapter X710-T2L
- Intel® Ethernet Network Adapter X710-T4L
- Intel® Ethernet Network Adapter X710-TL

Intel Ethernet 500 Series

The following devices do not support WoL:

- Intel® Ethernet Server Adapter X520-2
- Intel® Ethernet Server Adapter X520-1
- Intel® Ethernet Server Adapter X540-T1
- Intel® Ethernet Converged Network Adapter X540-T2
- Intel® Ethernet Converged Network Adapter X540-T1

> **Note:**
>
> Most Intel 10GbE Network Adapters do not support Wake on LAN on any port.
>
> The following 10GbE Network Adapters *do* support Wake on LAN on all ports:
>
> - Intel® Ethernet Server Adapter X550-T2 for OCP
> - Intel® Ethernet Server Adapter X550-T1 for OCP

Intel Ethernet 300 Series

The following support WoL only on Port A:

- Intel® Ethernet Server Adapter I350-T2

- Intel® Ethernet Server Adapter I350-T4
- Intel® Ethernet Server Adapter I340-T2
- Intel® Ethernet Server Adapter I340-T4
- Intel® Ethernet Server Adapter I340-F4

Other Power Options

Wake from S0ix on Magic Packet

Enables this device to bring the system out of an S0ix power state when the device receives a Magic Packet.

Possible values for this setting are:

- Enabled
- Disabled

Reduce Power if Cable Disconnected & Reduce Link Speed During Standby

Enables the adapter to reduce power consumption when the LAN cable is disconnected from the adapter and there is no link. When the adapter regains a valid link, adapter power usage returns to its normal state (full power usage).

The Hardware Default option is available on some adapters. If this option is selected, the feature is disabled or enabled based on the system hardware.

Possible values for this setting vary with the operating system and adapter.

Ultra Low Power Mode When Cable is Disconnected

Enabling Ultra Low Power (ULP) mode significantly reduces power consumption when the network cable is disconnected from the device.

> **Note:**
>
> If you experience link issues when two ULP-capable devices are connected back to back, disable ULP mode on one of the devices.

Selective Suspend

Enables the device to enter a low power state when the network is idle.

To change this setting in Intel PROSet:

Possible values for this setting are:

- Enabled
- Disabled

Selective Suspend Idle Timeout

Sets the length of time that the network is idle before the device enters a low power state. You must enable Selective Suspend for this setting to take effect.

To change this setting in Intel PROSet:

Possible values for this setting are:

- 1 - 60 in seconds

Energy Efficient Ethernet

The Energy Efficient Ethernet (EEE) feature allows a capable device to enter Low-Power Idle between bursts of network traffic. Both ends of a link must have EEE enabled for any power to be saved. Both ends of the link will resume full power when data needs to be transmitted. This transition may introduce a small amount of network latency.

Note:

- Both ends of the EEE link must automatically negotiate link speed.
- EEE is not supported on every adapter.

Priority and VLAN Tagging

This setting enables the adapter to offload the insertion and removal of priority and VLAN tags for transmit and receive.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To set this in Windows Powershell, first disable DCB, then set priority and VLAN tagging. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "DCB"-DisplayValue "Disabled"Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "PacketPriority & VLAN" -DisplayValue "VLAN Enabled"
```

Possible values for this setting are:

- Priority & VLAN Disabled
- Priority Enabled
- VLAN Enabled

- Priority & VLAN Enabled

Quality of Service

Quality of Service (QoS) allows the adapter to send and receive IEEE 802.3ac tagged frames. 802.3ac tagged frames include 802.1p priority-tagged frames and 802.1Q VLAN-tagged frames.

To implement QoS, the adapter must be connected to a switch that supports and is configured for 802.1p QoS. Priority-tagged frames allow programs that deal with real-time events to make the most efficient use of network bandwidth. High priority packets are processed before lower priority packets.

To change this setting in Intel PROSet:

Tagging is enabled and disabled using the following fields:

- **Windows Server:** Use the "QoS Packet Tagging" field in the Advanced tab in Intel PROSet or in the Adapter Settings panel in Intel PROSet ACU.
- **Other versions of Windows:** Use the "Priority/VLAN Tagging" setting on the Advanced tab in Intel PROSet or in the Adapter Settings panel in Intel PROSet ACU.

To set this in Windows Powershell, first disable DCB, then set QoS using the Priority and VLAN tagging `DisplayName` in the cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "DCB"-DisplayValue "Disabled"Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "PacketPriority & VLAN" -DisplayValue "VLAN Enabled"
```

Once QoS is enabled, you can specify levels of priority based on IEEE 802.1p/802.1Q frame tagging.

The supported operating systems, including Windows Server, have a utility for 802.1p packet prioritization. For more information, see the Windows system help and Microsoft's knowledge base.

Receive Buffers

This setting defines the number of Receive Buffers, which are data segments. They are allocated in the host memory and used to store the received packets. Each received packet requires at least one Receive Buffer, and each buffer uses 2KB of memory.

You might choose to increase the number of Receive Buffers if you notice a significant decrease in the performance of received traffic. If receive performance is not an issue, use the default setting appropriate to the adapter.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Receive Buffers"-DisplayValue "256"
```

Possible values for this setting are:

- 128-4096, in intervals of 64, for all adapters

We recommend the following values:

- Teamed adapter: 256
- Using IPSec and/or multiple features: 352

Receive Side Scaling

When Receive Side Scaling (RSS) is enabled, all of the receive data processing for a particular TCP connection is shared across multiple processors or processor cores. Without RSS all of the processing is performed by a single processor, resulting in less efficient system cache utilization.

LAN RSS

LAN RSS applies to a particular TCP connection.

> **Note:** This setting has no effect if your system has only one processing unit.

LAN RSS Configuration

If your adapter does not support RSS, or if the SNP or SP2 is not installed, the RSS setting will not be displayed. If RSS is supported in your system environment, the following will be displayed:

- **Port NUMA Node.** This is the NUMA node number of a device.
- **Starting RSS CPU.** This setting allows you to set the preferred starting RSS processor. Change this setting if the current processor is dedicated to other processes. The setting range is from 0 to the number of logical CPUs - 1.
- **Max number of RSS CPU.** This setting allows you to set the maximum number of CPUs assigned to an adapter and is primarily used in a Hyper-V environment. By decreasing this setting in a Hyper-V environment, the total number of interrupts is reduced which lowers CPU utilization. The default is 8 for Gigabit adapters and 16 for 10 Gigabit, or faster, adapters.

- **Preferred NUMA Node.** This setting allows you to choose the preferred NUMA (Non-Uniform Memory Access) node to be used for memory allocations made by the network adapter. In addition, the system will attempt to use the CPUs from the preferred NUMA node first for the purposes of RSS. On NUMA platforms, memory access latency is dependent on the memory location. Allocation of memory from the closest node helps improve performance. The Windows Task Manager shows the NUMA Node ID for each processor.

  Note:

  - This setting only affects NUMA systems. It will have no effect on non-NUMA systems.
  - Choosing a value greater than the number of NUMA nodes present in the system selects the NUMA node closest to the device.

- **Receive Side Scaling Queues.** This setting configures the number of RSS queues, which determine the space to buffer transactions between the network adapter and CPU(s).

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Receive Side Scaling"-DisplayValue "Enabled"
```

Possible values for this setting are:

- 1 queue is used when low CPU utilization is required.
- 2 queues are used when good throughput and low CPU utilization are required.
- 4 or more queues are used for applications that demand maximum throughput and transactions per second.

Note:

- Not all settings are available on all adapters.
- 8, or more, queues are only available when Intel PROSet for Windows Device Manager or Intel PROSet ACU is installed. If Intel PROSet is not installed, only 4 queues are available.
- Using 8 or more queues requires the system to reboot.

LAN RSS and Teaming

- If RSS is not enabled for all adapters in a team, RSS will be disabled for the team.

- If an adapter that does not support RSS is added to a team, RSS will be disabled for the team.
- If you create a multi-vendor team, you must manually verify that the RSS settings for all adapters in the team are the same.

Remote Boot

Remote Boot allows you to boot a system using only an Ethernet adapter. You connect to a server that contains an operating system image and use that to boot your local system.

See the following subsections for more information.

- Flash Images
- Intel® Boot Agent

> **Note:**
>
> See Configuring the UEFI Network Stack for PXE for information on enabling a PXE network boot.

Enabling Remote Boot

- If you have an Intel Desktop Adapter installed in your client computer, the flash ROM device is already available in your adapter, and no further installation steps are necessary.
- For Intel Server Adapters, the flash ROM can be enabled using the BootUtil utility. Do the following:
  1. From the command line, enter the following to enumerate the ports available in your system:

     ```
     BOOTUTIL -E
     ```

  2. Choose a port.
  3. Enter the following to select the port you wish to enable. For example:

     ```
     BOOTUTIL -NIC=1 -FLASHENABLE
     ```

See Intel® Ethernet Flash Firmware Utility for more information on using BootUtil.

Flash Images

“Flash” is a generic term for nonvolatile RAM (NVRAM), firmware, and option ROM (OROM). Depending on the device, it can be on the NIC or on the system board.

This page describes how to update the flash for the following OSes:

- Windows
- Linux
- UEFI

Note:

You cannot update the flash of a device in the “Pending Reboot” state. Reboot your system before attempting to update the device’s flash.

Updating the Flash in Microsoft Windows

Intel PROSet for Windows Device Manager can update the flash on an Intel Ethernet network adapter. However, if you need to enable or disable the Boot ROM, use BootUtil. See Intel® Ethernet Flash Firmware Utility for more information on using BootUtil.

Intel PROSet for Windows Device Manager can only be used to program add-in Intel Ethernet network adapters. LOM (LAN On Motherboard) network connections cannot be programmed with the UEFI network driver option ROM.

Note: This functionality is not supported in Intel PROSet ACU.

Using Intel PROSet to Flash the UEFI Network Driver Option ROM

Intel PROSet for Windows Device Manager can install the UEFI network driver on an Intel network adapter’s option ROM. The UEFI network driver will load automatically during system UEFI boot when installed in the option ROM.

UEFI-specific *.FLB* images are included in the downloaded release media. The “Boot Options” tab in Intel PROSet for Windows Device Manager will allow the UEFI *.FLB* image to be installed on the network adapter.

Note: This functionality is not supported in Intel PROSet ACU.

Updating the Flash from Linux

The BootUtil command line utility can update the flash on an Intel Ethernet network adapter. Run BootUtil with the following command line options to update the flash on all supported Intel network adapters. For example, enter the following at the command line:

```
bootutil64e -up=efi -all
```

See Intel® Ethernet Flash Firmware Utility for more information on using BootUtil.

Note:

BootUtil can only be used to program add-in Intel Ethernet network adapters. LOM (LAN On Motherboard) network connections cannot be programmed with the UEFI network driver option ROM.

Installing the UEFI Network Driver Option ROM from the UEFI Shell

The BootUtil command line utility can install the UEFI network driver on an Intel network adapter’s option ROM. The UEFI network driver will load automatically during system UEFI boot when installed into the option ROM.

For example, run BootUtil with the following command line options to install the UEFI network driver on all supported Intel network adapters:

```
FS0:\>bootutil64e -up=efi -all
```

See Intel® Ethernet Flash Firmware Utility for more information on using BootUtil.

Note:

BootUtil can only be used to program add-in Intel Ethernet network adapters. LOM (LAN On Motherboard) network connections cannot be programmed with the UEFI network driver option ROM.

Intel® Boot Agent

The Intel® Boot Agent is a software product that allows your networked client computer to boot using a program code image supplied by a remote server.

Intel Boot Agent complies with the Pre-boot eXecution Environment (PXE) Version 2.1 Specification. It is compatible with legacy boot agent environments that use BOOTP protocol.

Configuring the Intel Boot Agent Client

The Intel Boot Agent software provides configuration options that allow you to customize the behavior of the Intel Boot Agent software. You can configure the Intel Boot Agent in any of the following environments:

- A Microsoft Windows environment
- A preboot environment (before operating system is loaded)

The Intel Boot Agent supports PXE in preboot and Microsoft Windows environments. In each of these environments, a single user interface allows you to configure PXE protocols on Intel Ethernet adapters.

Important:

If spanning tree protocol is enabled on a switch port through which a port is trying to use PXE, the delay before the port starts forwarding can cause a DHCP timeout. Either disable spanning tree or turn on the feature that allows the port to begin forwarding of packets immediately (“port fast learning” for Cisco switches), rather than wait until the spanning tree discovery is complete.

Configuring the Intel Boot Agent in a Microsoft Windows Environment

If you use the Windows operating system on your client computer, you can use Intel® PROSet for Windows* Device Manager to configure and update the Intel Boot Agent software. Intel PROSet is available through the device manager. Intel PROSet provides a special tab, called the **Boot Options** tab, used for configuring and updating the Intel Boot Agent software.

To access the **Boot Options** tab:

1. Open Intel PROSet for Windows Device Manager by opening the **System** Control Panel. On the **Hardware** tab, click **Device Manager**.
2. Select the appropriate adapter and click the **Boot Options** tab. If the tab does not appear, update your network driver.
3. The **Boot Options** tab shows a list of current configuration parameters and their corresponding values. Corresponding configuration values appear for the selected setting in a drop-down box.
4. Select a setting you want to change from the **Settings** selection box.
5. Select a value for that setting from the **Value** drop-down list.
6. Repeat the preceding two steps to change any additional settings.
7. Once you have completed your changes, click **Apply Changes** to update the adapter with the new values.

Configuring the Intel Boot Agent in a Preboot PXE Environment

Note: Intel Boot Agent may be disabled in the BIOS.

You can customize the behavior of the Intel Boot Agent software through a preboot (operating system independent) configuration setup program contained within the adapter’s flash ROM. You can access this preboot configuration setup program each time the client computer cycles through the boot process.

When the boot process begins, the screen clears and the computer begins its Power On Self Test (POST) sequence. Shortly after completion of the POST, the Intel Boot Agent software stored in flash ROM executes. The Intel Boot Agent then displays an initialization message, similar to the one below, indicating that it is active:

```
Initializing Intel(R) Boot Agent Version X.X.XXPXE 2.0 B
uild 083
```

Note:

This display may be hidden by the manufacturer’s splash screen. Consult your manufacturer’s documentation for details.

The configuration setup menu shows a list of configuration settings on the left and their corresponding values on the right. Key descriptions near the bottom of the menu indicate how to change values for the configuration settings. For each selected setting, a brief “mini-Help” description of its function appears just above the key descriptions.

1. Highlight the setting you need to change by using the arrow keys.
2. Once you have accessed the setting you want to change, press the space bar until the desired value appears.
3. Once you have completed your changes, press **F4** to update the adapter with the new values. Any changed configuration values are applied as the boot process resumes.

The following describes the available configuration settings and their possible values.

Network Boot Protocol:

Select PXE for use with Network management programs, such as LANDesk* Management Suite. Depending on the configuration of the Intel Boot Agent, this parameter may not be changeable.

Possible values include:

- PXE (Preboot eXecution Environment)

Boot Order:

Sets the boot order in which devices are selected during boot up if the computer does not have its own control method.

If your client computer’s BIOS supports the BIOS Boot Specification (BBS), or allows PnP-compliant selection of the boot order in the BIOS setup program, then this setting will always be **Use BIOS Setup Boot Order** and cannot be changed. In this case, refer to the BIOS setup manual specific to your client computer to set up boot options.

If your client computer does not have a BBS- or PnP-compliant BIOS, you can select any one of the other possible values listed for this setting except for **Use BIOS Setup Boot Order**.

Possible values include:

- Use BIOS Setup Boot Order
- Try network first, then local drives
- Try local drives first, then network
- Try network only
- Try local drives only

Legacy OS Wakeup Support:

(For 82559-based adapters only)

If set to 1, the Intel Boot Agent will enable PME in the adapter's PCI configuration space during initialization. This allows remote wakeup under legacy operating systems that don't normally support it. Note that enabling this makes the adapter technically non-compliant with the ACPI specification, which is why the default is disabled.

Possible values include:

- 0 = Disabled (Default Value)
- 1 = Enabled

Note:

If, during PXE boot, more than one adapter is installed in a computer and you want to boot from the boot ROM located on a specific adapter, you can do so by moving the adapter to the top of the BIOS Boot Order or by disabling the flash on the other adapters.

While the configuration setup menu is displayed, diagnostics information is also displayed in the lower half of the screen. This information can be helpful during interaction with Intel Customer Support personnel or your IT team members. For more information about how to interpret the information displayed, refer to Diagnostics Information for Preboot PXE Environments below.

Setting Up the Intel Boot Agent Target/Server

For the Intel Boot Agent software to perform its intended job, a server must be set up on the same network as the client computer. That server must recognize and respond to the PXE or BOOTP boot protocols that are used by the Intel Boot Agent software.

Note:

When the Intel Boot Agent software is installed as an upgrade for an earlier version boot ROM, the associated server-side software may not be compatible with the updated Intel Boot Agent. Contact your system administrator to determine if any server updates are necessary.

Linux Server Setup

Consult your Linux vendor for information about setting up the Linux Server.

Windows Deployment Services

Nothing is needed beyond the standard driver files supplied on the media. Microsoft owns the process and associated instructions for Windows Deployment Services. For more information on Windows Deployment Services perform a search of Microsoft articles at http://technet.microsoft.com/en-us/library/default.aspx.

Intel Boot Agent Troubleshooting

Common Issues

The following list of problems and associated solutions covers a representative set of problems that you might encounter while using the Intel Boot Agent.

After booting, my computer experiences problems

After the Intel Boot Agent product has finished its sole task (remote booting), it no longer has any effect on the client computer operation. Thus, any issues that arise after the boot process is complete are most likely not related to the Intel Boot Agent product.

If you are having problems with the local (client) or network operating system, contact the operating system manufacturer for assistance. If you are having problems with some application program, contact the application manufacturer for assistance. If you are having problems with any of your computer's hardware or with the BIOS, contact your computer system manufacturer for assistance.

Cannot change boot order

If you are accustomed to redefining your computer's boot order using the motherboard BIOS setup program, the default settings of the Intel Boot Agent setup program can override that setup. To change the boot sequence, you must first override the Intel Boot Agent setup program defaults. A configuration setup menu appears allowing you to set configuration values for the Intel Boot Agent. To change your computer's boot order setting, see Configuring the Intel Boot Agent in a Preboot PXE Environment.

My computer does not complete POST

If your computer fails to boot with an adapter installed, but does boot when you remove the adapter, try moving the adapter to another computer and using BootUtil to disable the Flash ROM.

If this does not work, the problem may be occurring before the Intel Boot Agent software even begins operating. In this case, there may be a BIOS problem with your computer. Contact your computer manufacturer's customer support group for help in correcting your problem.

There are configuration/operation problems with the boot process

If your PXE client receives a DHCP address, but then fails to boot, you know the PXE client is working correctly. Check your network or PXE server configuration to troubleshoot the problem. Contact Customer Support if you need further assistance.

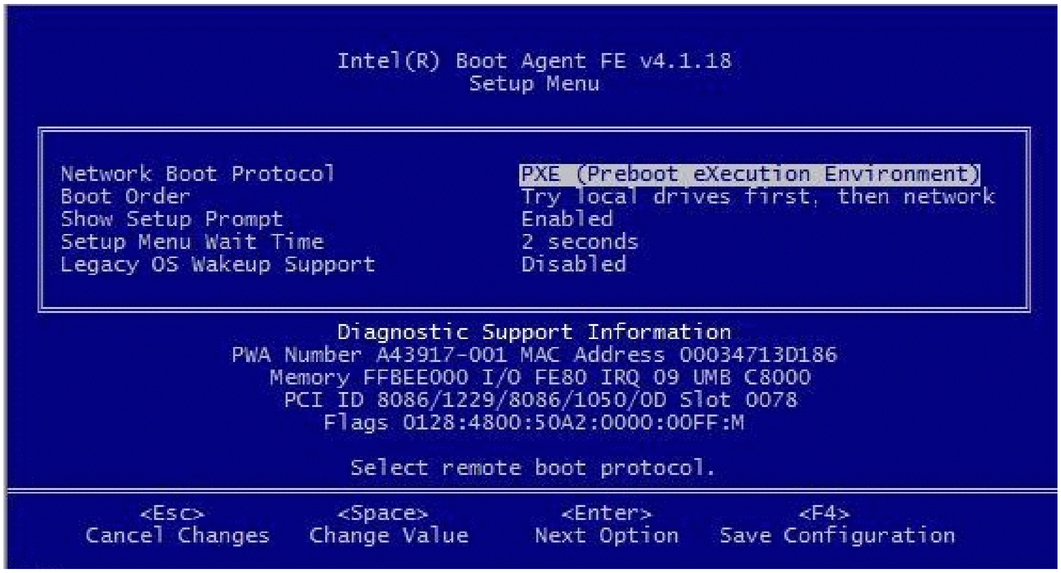
PXE option ROM does not follow the PXE specification with respect to the final "discover" cycle

In order to avoid long wait periods, the option ROM no longer includes the final 32-second discover cycle. (If there was no response in the prior 16-second cycle, it is almost certain that there will be none in the final 32-second cycle.

Diagnostics Information for Preboot PXE Environments

Anytime the configuration setup menu is displayed (see Configuring the Intel Boot Agent in a Preboot PXE Environment above), diagnostics information is also displayed on the lower portion of the screen. The information displayed appears similar to that shown in the lower half of the screen image below. This information can be helpful during interaction with Intel Customer Support personnel or your IT team members.

> **Note:**
>
> Actual diagnostics information may vary, depending upon the adapter(s) installed in your computer.



```
                    Intel(R) Boot Agent FE v4.1.18
                              Setup Menu

  Network Boot Protocol          PXE (Preboot eXecution Environment)
  Boot Order                     Try local drives first, then network
  Show Setup Prompt              Enabled
  Setup Menu Wait Time           2 seconds
  Legacy OS Wakeup Support       Disabled

                    Diagnostic Support Information
           PWA Number A43917-001 MAC Address 00034713D186
              Memory FFBEE000 I/O FE80 IRQ 09 UMB C8000
               PCI ID 8086/1229/8086/1050/0D Slot 0078
                  Flags 0128:4800:50A2:0000:00FF:M

                     Select remote boot protocol.

       <Esc>            <Space>            <Enter>           <F4>
   Cancel Changes    Change Value       Next Option    Save Configuration
```

Diagnostics information may include the following items:

- PWA Number: The Printed Wire Assembly number identifies the adapter’s model and version.
- MAC Address: The unique Ethernet address assigned to the device.
- Memory: The memory address assigned by the BIOS for memory-mapped adapter access.
- I/O: The I/O port address assigned by the BIOS for I/O-mapped adapter access.
- IRQ: The hardware interrupt assigned by the system BIOS.
- UNB: The address in upper memory where the Boot Agent is installed by the BIOS.
- PCI ID: The set of PCI identification values from the adapter in the form:

  ```
  VendorID/DeviceID/SubvendorID/SubdeviceID/Revision
  ```

- Slot: The PCI bus address (slot number)reported by the BIOS.

Note:

The number displayed is the BIOS version of the PCI slot number. Therefore, actual positions of NICs within physical slots may not be displayed as expected. Slots are not always enumerated in an obvious manner, and this will only report what is indicated by the BIOS.

- Flags: A set of miscellaneous data either read from the adapter EEPROM or calculated by the Boot Agent initialization code. This information varies from one adapter to the next and is only intended for use by Intel customer support.

Remote Direct Memory Access (RDMA)

Remote Direct Memory Access, or RDMA, allows a network device to transfer data directly to and from application memory on another system, increasing throughput and lowering latency in certain networking environments.

- Intel® Ethernet 800 Series devices support both iWARP and RoCEv2.
- Intel® Ethernet X722 Series devices only support iWARP.

The major difference is that iWARP performs RDMA over TCP, while RoCEv2 uses UDP.

On devices with RDMA capabilities, RDMA is supported on the following operating systems:

- Linux*
- FreeBSD*
- VMware* ESXi*
- Microsoft* Windows Server*

To avoid performance degradation from dropped packets, enable link level flow control or priority flow control on all network interfaces and switches.

Note:

- On systems running a Microsoft Windows Server operating system, enabling *QoS/priority flow control will disable link level flow control.
- Devices based on the Intel® Ethernet 800 Series do not support RDMA when operating in multiport mode with more than 4 ports.
- On Linux systems, RDMA and link aggregation (LAG, also known as *bonding*) are not compatible on most devices. If RDMA is enabled, bonding will not be functional.
  - On Intel Ethernet 810 Series devices, RDMA and LAG are compatible if all the following are true:
    - You are using an Intel Ethernet 810 Series device with the latest drivers and NVM installed.
    - RDMA technology is set to RoCEv2.

- LAG configuration is either active-backup or active-active.
- Bonding is between two ports within the same device.
- The QoS configuration of the two ports matches prior to the bonding of the devices.

See the README inside the Linux ice driver tarball for more information.

RDMA on Linux or FreeBSD

For Intel Ethernet devices that support RDMA on Linux or FreeBSD, use the drivers shown in the following table.

Device	**Linux**		**FreeBSD**		**S upported P rotocols**
	Base Driver	RDMA Driver	Base Driver	RDMA Driver	
Intel® Ethernet 800 Series	ice	irdma	ice	irdma	RoCEv2, iWARP
Intel® Ethernet X722 Series	i40e	irdma	ixl	not s upported	iWARP

Basic Installation Instructions

At a high level, installing and configuring RDMA on Linux or FreeBSD consists of the following steps. See the README file inside the appropriate RDMA driver tarball for full details.

1. Install the base driver.
2. Install the RDMA driver.
3. Install and patch any user-mode RDMA libraries. Exact steps will vary by operating system; refer to the RDMA driver readme for details.
4. Enable flow control on your device. Refer to the base driver README for details and supported modes.

5. If you are using RoCE, enable flow control (PFC or LFC) on the device and endpoint your system is connected to. See your switch documentation and, for Linux, the Intel® Ethernet 800 Series Linux Flow Control Configuration Guide for RDMA Use Cases for details.

RDMA for Virtualized Environments in Linux

Devices based on the Intel Ethernet 800 Series support RDMA in a Linux VF on supported Windows or Linux hosts. Refer to the README file inside the Linux RDMA driver tarball for more information on how to load and configure RDMA in a Linux VF.

RDMA on Microsoft Windows

RDMA for Network Direct (ND) User-Mode Applications

Network Direct (ND) allows user-mode applications to use RDMA features.

> **Note:**
>
> User-mode applications may have prerequisites such as Microsoft HPC Pack or Intel MPI Library, refer to your application documentation for more details.

Supported Operating Systems

Intel® Ethernet User Mode RDMA Provider is supported on Microsoft Windows Server.

RDMA User Mode Installation

Follow the steps below to install user-mode Network Direct features:

1. Download the software package you want for the release. See Install Windows Drivers for more information.
   a. If you are installing via the complete driver pack:
      i. In the extracted files from the download, navigate to `\APPS\PROSETDX` and then the Windows subfolder corresponding to your version of Windows (32-bit or 64-bit).
      ii. Inside the `Winx64` or `Win32` folder, double-click on *DxSetup.exe* to launch the install wizard.
   b. If you are installing via the separate webpacks for base drivers and Intel PROSet:
      i. Download and extract the webpack for Intel PROSet.
      ii. In the extracted files, double-click on the **.exe** file to launch the install wizard.
2. On the Setup Options screen, select **Intel® Ethernet User Mode RDMA Provider**.
3. On the RDMA Configuration Options screen, select **Enable RDMA routing across IP Subnets** if desired. Note that this option is displayed during base

driver installation even if user mode RDMA was not selected, as this option is applicable to Network Direct Kernel functionality as well.

4. If Windows Firewall is installed and active, select **Create an Intel® Ethernet RDMA Port Mapping Service rule in Windows Firewall** and the networks to which to apply the rule.

   > **Note:**
   >
   > If Windows Firewall is disabled or you are using a third party firewall, you will need to add this rule manually.

5. Continue with driver and software installation.

RDMA Network Direct Kernel (NDK)

RDMA Network Direct Kernel (NDK) functionality is included in the Intel base networking drivers and requires no additional features to be installed.

RDMA Routing across IP Subnets

If you want to allow NDK's RDMA functionality across subnets, you will need to select **Enable RDMA routing across IP Subnets** on the RDMA Configuration Options screen during base driver installation. See Install Windows Drivers for basic installation instructions.

Enabling Priority Flow Control (PFC) on a Microsoft Windows Server

To avoid performance degradation from dropped packets, enable priority flow control (PFC) or link level flow control on all network interfaces and switches.

> **Note:**
>
> On systems running a Microsoft Windows Server operating system, enabling *QoS/priority flow control will disable link level flow control.

Use the following PowerShell commands to enable PFC on Microsoft Windows Server:

```
Install-WindowsFeature -Name Data-Center-Bridging -IncludeManagementToolsNew-NetQoSPolicy "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3Enable-NetQosFlowControl -Priority 3Disable-NetQosFlowControl -Priority 0,1,2,4,5,6,7New-NetQosTrafficClass -Name "SMB" -Priority 3 -BandwidthPercentage 60 -Algorithm ETSSet-NetQosDcbxSetting -Willing $FALSEEnable-NetAdapterQos -Name "Slot1 4 2 Port 1"
```

Verifying RDMA Operation with Microsoft PowerShell

You can check that RDMA is enabled on the network interfaces using the following Microsoft PowerShell command:

```
Get-NetAdapterRDMA
```

Use the following PowerShell command to check if the network interfaces are RDMA capable and multichannel is enabled:

```
Get-SmbClientNetworkInterface
```

Use the following PowerShell command to check if Network Direct is enabled in the operating system:

```
Get-NetOffloadGlobalSetting | Select NetworkDirect
```

Use **netstat** to make sure each RDMA-capable network interface has a listener at port 445 (Windows Client OSs that support RDMA may not post listeners). For example:

```
netstat.exe -xan | ? {$_ -match "445"}
```

RDMA for Virtualized Environments in Windows

To enable RDMA functionality on virtual adapter(s) connected to a VMSwitch, you must:

- Enable SR-IOV (Single Root IO Virtualization) and VMQ (Virtual Machine Queues) advanced properties on each port.
- Set the number of VFs to enable with RDMA capabilities. You can enable up to 32 VFs with RDMA capabilities.

Under certain circumstances, you may disable these settings by default. You can manually set these options in the Adapter Settings panel of Intel PROSet ACU, in the Advanced tab of the adapter properties dialog box, or with the following PowerShell commands:

```
Set-NetAdapterAdvancedProperty -Name <nic_name> -RegistryKeyword *SRIOV -RegistryValue 1Set-NetAdapterAdvancedProperty -Name <nic_name> -RegistryKeyword *VMQ -RegistryValue 1Set-NetAdapterAdvancedProperty -Name <nic_name> -RegistryKeyword RdmaMaxVfsEnabled -RegistryValue <1-32>
```

Configuring RDMA Guest Support (NDK Mode 3)

NDK Mode 3 allows kernel mode Windows components to use RDMA features inside Hyper-V guest partitions. To enable NDK mode 3 on an Intel Ethernet device, do the following:

1. Enable SR-IOV in your system's BIOS or UEFI.
2. Enable the SR-IOV advanced setting on the device.
3. Enable SR-IOV on the VMSwitch bound to the device by performing the following for all physical functions on the same device:

   ```
   New-VMSwitch -Name <switch_name> -NetAdapterName <d
   evice_name> -EnableIov $true``
   ```

4. Configure the number of RDMA virtual functions (VFs) on the device by setting the `RdmaMaxVfsEnabled` advanced setting. All physical functions must be set to the same value. The value is the maximum number of VFs that can be capable of RDMA at one time for the entire device. Enabling more VFs will restrict RDMA resources from physical functions (PFs) and other VFs:

   ```
   Set-NetAdapterAdvancedProperty -Name <device_name>
   -RegistryKeyword RdmaMaxVfsEnabled  -RegistryValue
   <Value: 0 - 32>
   ```

5. Disable all PF adapters on the host and re-enable them. This is required when the registry keyword `RdmaMaxVfsEnabled` is changed or when creating or destroying a VMSwitch:

   ```
   Get-NetAdapterRdma | Disable-NetAdapterGet-NetAdapt
   erRdma | Enable-NetAdapter
   ```

6. Create VM Network Adapters for VMs that require RDMA VF support:

   ```
   Add-VMNetworkAdapter -VMName <vm_name> -VMNetworkAd
   apterName <device_name> -SwitchName <switch_name>
   ```

7. If you plan to use Microsoft Windows 10 Creators Update (RS2) or later on a guest partition, set the RDMA weight on the VM Network Adapter by entering the following command on the host:

   ```
   Set-VMNetworkAdapterRdma -VMName <vm_name> -VMNetwo
   rkAdapterName <device_name> -RdmaWeight 100
   ```

8. Set SR-IOV weight on the VM Network Adapter (Note: SR-IOV weight must be set to 0 before setting the RdmaWeight to 0):

   ```
   Set-VMNetworkAdapter -VMName <vm_name> -VMNetworkAd
   apterName <device_name> -IovWeight 100
   ```

9. Install the VF network adapter with the PROSET Installer in the VM.

10. Enable RDMA on the VF driver and Hyper-V Network Adapter using PowerShell in the VM:

    ```
    Set-NetAdapterAdvancedProperty -Name <device_name>
    -RegistryKeyword RdmaVfEnabled -RegistryValue 1Get-
    NetAdapterRdma | Enable-NetAdapterRdma
    ```

RDMA for NDK Features such as SMB Direct (Server Message Block)

NDK allows Windows components (such as SMB Direct storage) to use RDMA features.

Testing NDK: Microsoft Windows SMB Direct with DiskSPD

This section outlines the recommended way to test RDMA for Intel Ethernet functionality and performance on Microsoft Windows operating systems.

Note that since SMB Direct is a storage workload, the performance of the benchmark may be limited to the speed of the storage device rather than the network interface being tested. Intel recommends using the fastest storage possible in order to test the true capabilities of the network device(s) under test.

Test instructions:

1. Set up and connect at least two servers running a supported Microsoft Windows Server operating system, with at least one RDMA-capable Intel Ethernet device per server.
2. On the system designated as the SMB server, set up an SMB share. Note that the performance of the benchmark may be limited to the speed of the storage device rather than the network interface being tested. Storage setup is outside of the scope of this document. You can use the following PowerShell command:

   ```
   New-SmbShare -Name <SMBsharename> -Path <SMBsharefi
   lepath> -FullAccess  <domainname>\Administrator,Eve
   ryone
   ```

   For example:

   ```
   New-SmbShare -Name RAMDISKShare -Path R:\RAMDISK -F
   ullAccess group\Administrator,Everyone
   ```

3. Download and install the Diskspd Microsoft utility from here: https://gallery.technet.microsoft.com/DiskSpd-a-robust-storage-6cd2f223
4. Using CMD or Powershell, use the **cd** command to change to the DiskSpd folder, and then run tests. (Refer to Diskspd documentation for more details on parameters)

   For example: Set the block size to 4K, run the test for 60 seconds, disable all hardware and software caching, measure and display latency statistics, leverage 16 overlapped IOs and 16 threads per target, random 0% writes and 100% reads and create a 10GB test file at \

\<SMBserverTestIP>\<SMBsharename>\test.dat:

```
.\diskspd.exe -b4K -d60 -h -L -o16 -t16 -r -w0 -c10
G \\<SMBserverTestIP>\<SMBsharename>\test.dat
```

5. Verify that RDMA traffic is running using perfmon counters such as “RDMA Activity” and “SMB Direct Connection”. Refer to Microsoft documentation for more details.

RDMA Windows Performance Monitoring

You can use **perfmon**, or other performance monitoring tool, to monitor and display RDMA counters and statistics. Refer to Microsoft documentation for more details. Use the `Register-IntelEthernetRDMACounterSet` cmdlet to register the RDMA statistics counters for the specific device with perfmon. Refer to Configuring Features with Windows PowerShell for more information about how to install and use Intel cmdlets. You can use the following PowerShell command to register the RDMA statistics for all supported devices:

```
Register-IntelEthernetRDMACounterSet
```

You can use the following PowerShell cmdlet to unregister the RDMA statistics:

```
Unregister-IntelEthernetRDMACounterSet
```

Accessing Remote NVM Express* Drives Using RDMA

RDMA provides a high throughput, low latency means to directly access NVM Express* (NVMe*) drives on a remote server.

Refer to the following for details on supported operating systems and how to set up and configure your server and client systems:

- NVM Express over TCP for Intel® Ethernet Products Configuration Guide
- NVM Express over Fabrics for Intel® Ethernet Products with RDMA Configuration Guide

Both guides are available on the Intel Technical Library.

Setting Speed and Duplex

The Link Speed and Duplex setting lets you choose how the adapter sends and receives data packets over the network.

In the default mode, an Intel network adapter using copper connections will attempt to auto-negotiate with its link partner to determine the best setting. If the adapter cannot establish link with the link partner using auto- negotiation, you may need to manually configure the adapter and link partner to the identical setting to establish link and pass packets. This should only be needed when attempting to link with an older switch that does not support auto- negotiation or one that has been forced to a specific speed or duplex mode.

Auto-negotiation is disabled by selecting a discrete speed and duplex mode in the adapter properties sheet. The settings available when auto-negotiation is disabled are dependent on your device. Not all speeds are available on all devices. Your link partner must match the setting you choose.

Note:

- Only experienced network administrators should force speed and duplex manually.
- Fiber-based adapters operate only in full duplex at their native speed. You cannot change the speed or duplex of Intel adapters that use fiber cabling.
- Some devices may list 10Mbps and 100Mbps in full or half duplex as options. These settings are not recommended.
- Link speed information in Intel PROSet may display a blue informational icon with a mouse-over message “This device is not linked at its maximum capable speed.” In that case, if your device is set to auto-negotiate, you can adjust the speed of the device’s link partner to the device’s maximum speed. If the device is not set to auto-negotiate, you can adjust the device’s speed manually, but you must ensure the link partner is set at the same speed.
- Auto-negotiation and Auto-Try are not supported on devices based on the Intel® Ethernet Connection X552 and Intel® Ethernet Connection X553.

Manually Configuring Duplex and Speed Settings

Note:

The settings at the switch must always match the adapter settings. Adapter performance may suffer, or your adapter might not operate correctly if you configure the adapter differently from your switch.

The default setting is for auto-negotiation to be enabled. Only change this setting to match your link partner’s speed and duplex setting if you are having trouble connecting.

In Intel PROSet for Windows Device Manager:

1. In Windows Device Manager, double-click the adapter you want to configure.
2. On the **Link Speed** tab, select a speed and duplex option from the **Speed and Duplex** drop-down menu.
3. Click **OK**.

In Intel PROSet ACU, link speed is reported on the Adapter Information panel. Change speed and duplex in the Adapter Settings panel.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Speed & Duplex"-DisplayValue "Auto Negotiation"
```

Thermal Monitoring

Adapters and network controllers based on the Intel® Ethernet Controller I350 (and later controllers) can display temperature data and automatically reduce the link speed if the controller temperature gets too hot.

Note:

This feature is enabled and configured by the equipment manufacturer. It is not available on all adapters and network controllers. There are no user configurable settings.

Monitoring and Reporting

Temperature information is displayed on the Link tab in Intel PROSet for Windows Device Manager or in the Adapter Information panel in Intel PROSet ACU. There are three possible conditions:

- **Temperature: Normal** - Indicates normal operation.
- **Temperature: Overheated, Link Reduced** - Indicates that the device has reduced link speed to lower power consumption and heat.
- **Temperature: Overheated, Adapter Stopped** - Indicates that the device is too hot and has stopped passing traffic so it is not damaged.

If either of the overheated events occur, the device driver writes a message to the system event log.

Timestamps

- PTP Hardware Timestamp
- Software Timestamp

PTP Hardware Timestamp

This setting allows applications that use PTPv2 (Precision Time Protocol) to use hardware generated timestamps to synchronize clocks throughout your network. If this setting is enabled, it takes precedence over the Software Timestamp setting.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "PTP HardwareTimestamp" -DisplayValue "Enabled"
```

Possible values for this setting are:

- Enabled
- Disabled

Software Timestamp

This setting allows applications that use PTPv2 (Precision Time Protocol) to use software generated timestamps to synchronize clocks throughout your network. If the PTP Hardware Timestamp setting is enabled, it takes precedence over this setting.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "Software Timestamp"-DisplayValue "RxAll"
```

Possible values for this setting are:

- Disabled
- RxAll
- TxAll
- RxAll & TxAll
- TaggedTx
- RxAll & TaggedTx

Transmit Buffers

This setting defines the number of Transmit Buffers, which are data segments that enable the adapter to track transmit packets in the system memory. Depending on the size of the packet, each transmit packet requires one or more Transmit Buffers.

You might choose to increase the number of Transmit Buffers if you notice a possible problem with transmit performance. Although increasing the number of Transmit Buffers can enhance transmit performance, Transmit Buffers do consume system memory. If transmit performance is not an issue, use the default setting. This default setting varies with the type of adapter.

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device's Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "Transmit Buffers"-DisplayValue "128"
```

Possible values for this setting are:

- 128-16384, in intervals of 64, for 10 Gigabit Server Adapters
- 128-4096, in intervals of 64, for all other adapters

UEFI Network Device Drivers

UEFI Network Stack

As of UEFI 2.1, there are two network stack configurations under UEFI. The most common configuration is the PXE based network stack. The alternate network stack provides IPv4 TCP, UDP, and MTFTP network protocol support. As of UEFI 2.1, the PXE and IP-based network stacks cannot be loaded or operate simultaneously. The following two sections describe each UEFI network stack configuration.

You can download reference implementations of the PXE and IP based network stack source code at https://www.tianocore.org/.

Loading the UEFI Network Driver

Load the network driver using the UEFI shell **load** command. For example:

```
load e3040e2.efi
```

Configuring the UEFI Network Stack for PXE

The PXE (Preboot eXecution Environment) based UEFI network stack provides support for UEFI network boot loaders downloaded from a WFM-compliant PXE server. Services that can be enabled include:

- Windows Deployment Services (WDS)
- Linux network installation (Elilo)
- TFTP file transfers

To enable UEFI PXE services, the following network protocol drivers must be loaded with:

- snp.efi
- bc.efi
- pxedhcp4.efi

These drivers can be loaded from the UEFI **load** shell command, but are often included as part of the UEFI system firmware.

Use the UEFI shell command **drivers** to determine if the UEFI PXE drivers are included in the UEFI implementation. The **drivers** command will output a table listing drivers loaded in the system. The following entries must be present in order to network boot a UEFI system over PXE:

DRV	VERSION	TYPE	CFG	DIAG	#D	#C	DRIVER NAME	IMAGE NAME
F5	00000010	D	-	-	2	-	Simple Network Protocol Driver	SNP
F7	00000010	D	-	-	2	-	PXE Base Code Driver	BC

DRV	VERSION	TYPE	CFG	DIAG	#D	#C	DRIVER NAME	IMAGE NAME
F9	00000010	D	-	-	2	-	PXE DHCP v4 Driver	PxeDhcp4
FA	03004000	B	X	X	2	2	Intel(R) Network Connection 3.0.00	/e3000e2.efi

A network boot option will appear in the boot options menu when the UEFI PXE network stack and Intel UEFI network driver have been loaded. Selecting this boot option will initiate a PXE network boot.

Configuring the UEFI Network Stack for TCP, UDP, and MTFTP

An IP-based network stack is available to applications requiring IP-based network protocols such as TCP, UDP, or MTFTP. The following UEFI network drivers must be built into the UEFI platform implementation to enable this stack:

- SNP (Simple Network Protocol)
- MNP (Managed Network Protocol)
- ARP
- DHCP4
- IPv4
- ip4config
- TCPv4
- UDPv4
- MTFTPv4

These drivers will show up in the UEFI **drivers** command output if they are included in the platform UEFI implementation:

DRV	VERSION	TYPE	CFG	DIAG	#D	#C	DRIVER NAME	IMAGE NAME

DRV	VERSION	TYPE	CFG	DIAG	#D	#C	DRIVER NAME	IMAGE NAME
F5	00000010	D	-	-	2	-	IP4 CONFIG Network Service Driver	Ip4Config
F7	00000010	D	-	-	2	-	Simple Network Protocol Driver	SNP
F8	00000010	D	-	-	2	-	ARP Network Service Driver	Arp
F9	00000010	D	-	-	2	-	Tcp Network Service Driver	Tcp4
FA	00000010	D	-	-	2	-	IP4 Network Service Driver	Ip4
FB	00000010	D	-	-	2	-	DHCP Protocol Driver	Dhcp4

DRV	VERSION	TYPE	CFG	DIAG	#D	#C	DRIVER NAME	IMAGE NAME
FC	00000010	D	-	-	6	-	UDP Network Service Driver	Udp4
FD	00000010	D	-	-	2	-	MTFTP4 Network Service	Mtftp4
FE	00000010	B	-	-	2	6	MNP Network Service Driver	/mnp.efi
FF	03099900	B	X	X	2	2	Intel(R) Network Connection 3.0.00	/e3000e2.efi

The **ifconfig** UEFI shell command must be used to configure each network interface. Running `ifconfig -?` from the UEFI shell will display usage instructions for **ifconfig**.

Unloading the UEFI Network Driver

To unload a network driver from memory, use the UEFI **unload** command. The syntax for using the **unload** command is:

```
unload [driver handle]
```

Where `[driver handle]` is the number assigned to the driver in the far left column of the **drivers** output screen.

Force Speed and Duplex

The UEFI network driver supports forced speed and duplex capability. You can access the force speed and duplex menu with the **drvcfg** UEFI shell command:

```
drvcfg -s [driver handle] [control handle]
```

The speed and duplex setting selected must match the speed and duplex setting of the connecting network port. A speed and duplex mismatch between ports will result in dropped packets and poor network performance. It is recommended to set all ports on a network to autonegotiate. Connected ports must be set to autonegotiate in order to establish a 1 gigabit per second connection.

Fiber-optic and 10 gigabit (and faster) Ethernet adapters do not support forced speed and duplex.

Diagnostic Capability

The UEFI network driver features built in hardware diagnostic tests. The diagnostic tests are called with the UEFI shell **drvdiag** command.

The following performs a basic hardware register test:

```
drvdiag -s
```

The following performs an internal loopback transmit and receive test:

```
drvdiag -e
```

VF Loopback Pacing

On Linux and Windows Server, the Intel Ethernet 800 Series supports adjusting the loopback rate for a designated port, which allows you to prioritize that port for maximum bandwidth and achieve higher speeds.

- On Linux, use the **devlink** command and the `loopback` parameter to change this setting.
- On Windows Server, use Ethernet Cmdlets for Intel® Ethernet or Intel® PROSet to change the `Loopback` setting.

Note:

This feature is only supported on Windows Server 2019 and newer versions.

After changing the loopback setting, the driver will reconfigure all underlying VFs to align to the desired port settings and add more bandwidth to VF-to-VF traffic. Setting this parameter to `prioritized` enables higher hairpin-bandwidth on related PFs.

Note:

- This configuration applies only for 8x10G and 4x25G adapter cards.
- Typically you would set some ports to prioritized loopback and then disable loopback on other ports, to allow the driver to utilize spare bandwidth for VF-to-VF traffic.
- Intel recommends using the `prioritized` loopback setting on a port with minimal network traffic.
- You should first configure loopback on the PF and then configure any other settings, such as VFs/VMs or assigning MAC addresses.

To change this setting with Ethernet cmdlets or Intel PROSet:

This setting is found in the Adapter Settings panel in Intel PROSet ACU.

To change this setting with Ethernet cmdlets, use the `Set-IntelEthernetSetting` cmdlet. For example:

```
Set-IntelEthernetSetting -Name "<adapter_name>" -DisplayName "Loopback" -DisplayValue "Prioritized"
```

To change this setting in Intel® PROSet for Windows PowerShell* software, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "Loopback" -DisplayValue "Prioritized"
```

Possible values for this setting are:

- Enabled
- Disabled
- Prioritized

Virtualization Support

Virtualization makes it possible for one or more operating systems to run simultaneously on the same physical system as virtual machines. This allows you to consolidate several servers onto one system, even if they are running different operating systems. Intel® Network Adapters work with, and within, virtual machines with their standard drivers and software.

See the following subsections for more information.

- Single Root I/O Virtualization (SR-IOV)
- Intel® Scalable I/O Virtualization Support
- Virtual Machine Queue Offloading

- Using Intel Network Adapters in a Microsoft* Hyper-V* Environment

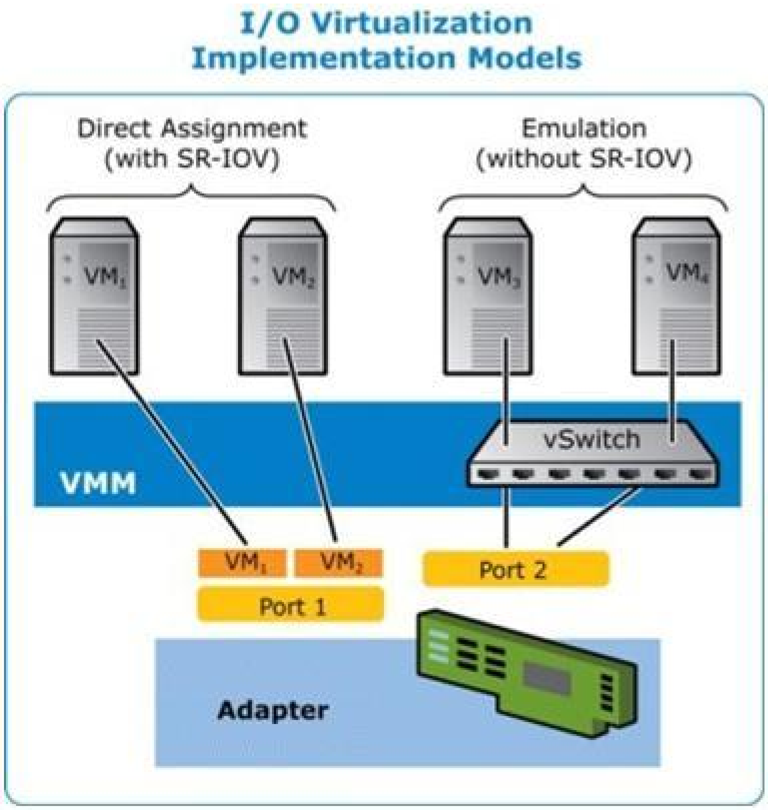
Note:

- Some virtualization options are not available on some adapter/operating system combinations.
- The jumbo frame setting inside a virtual machine must be the same, or lower than, the setting on the physical port.
- When you attach a Virtual Machine to a tenant overlay network through the Virtual NIC ports on a Virtual Switch, the encapsulation headers increase the Maximum Transmission Unit (MTU) size on the virtual port. The Encapsulation Overhead feature automatically adjusts the physical port’s MTU size to compensate for this increase.

Single Root I/O Virtualization (SR-IOV)

SR-IOV Overview

Single Root I/O Virtualization (SR-IOV) is a PCI SIG specification allowing PCI Express devices to appear as multiple separate physical PCI Express devices. SR-IOV allows efficient sharing of PCI devices among Virtual Machines (VMs). It manages and transports data without the use of a hypervisor by providing independent memory space, interrupts, and DMA streams for each virtual machine.



SR-IOV architecture includes two functions:

- Physical Function (PF) is a full featured PCI Express function that can be discovered, managed, and configured like any other PCI Express device.
- Virtual Function (VF) is similar to PF but cannot be configured and only has the ability to transfer data in and out. The VF is assigned to a Virtual Machine.

Configuring SR-IOV

SR-IOV lets a single network port appear to be several virtual functions in a virtualized environment. If you have an SR-IOV capable NIC, each port on that NIC can assign a virtual function to several guest partitions. The virtual functions bypass the Virtual Machine Manager (VMM), allowing packet data to move directly to a guest partition's memory, resulting in higher throughput and lower CPU utilization. SR-IOV also allows you to move packet data directly to a guest partition's memory. See your operating system documentation for system requirements.

For devices that support it, SR-IOV is enabled in the host partition. Some devices may need to have SR-IOV enabled in a preboot environment.

Note:

- **Configuring SR-IOV for improved network security:** In a virtualized environment, on Intel® Server Adapters that support SR-IOV or Intel® Scalable I/O Virtualization (Intel® Scalable IOV), the virtual function (VF) may be subject to malicious behavior. Software-generated layer two frames, like IEEE 802.3x (link flow control), IEEE 802.1Qbb (priority based flow-control), and others of this type, are not expected and can throttle traffic between the host and the virtual switch, reducing performance. To resolve this issue, and to ensure isolation from unintended traffic streams, configure all SR-IOV or Intel Scalable IOV enabled ports for VLAN tagging from the administrative interface on the PF. This configuration allows unexpected, and potentially malicious, frames to be dropped.
- SR-IOV must be enabled in the BIOS.
- You must enable VMQ for SR-IOV to function.
- For best performance, on the host use `Set-VMNetworkAdapter -IovQueuePairsRequested 4` on the VF to allow the virtual network to use 4 queues (maximum supported value) and assign 4 or more virtual CPUs to the connected VM. In the VM, set "Maximum number of Receive Queues" in the VF's adapter properties to 4.
- Binding more than two virtual functions (VFs) to a virtual machine (VM) is not recommended. Binding more VFs to a VM may cause system instability.
- SR-IOV is not supported with Intel ANS teams.
- VMWare ESXi does not support SR-IOV on 1GbE ports.

Configuring SR-IOV in Windows

Use Intel PROSet to change this setting in Windows.

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -Displ
ayName "SR-IOV" -DisplayValue "Enabled"
```

Intel® Scalable I/O Virtualization Support

Intel® Scalable I/O Virtualization (Intel® Scalable IOV) allows you to share a physical device across multiple virtual machines and applications. Intel Scalable IOV provides your system the ability to share device resources with different address domains using different abstractions. For example, application processes may access a device using system calls and VMs may access a device through virtual device interfaces.

Intel Scalable IOV and SR-IOV (Single Root I/O Virtualization) are mutually exclusive. If both are enabled on your system, and all of the Intel Scalable IOV requirements are met, the PF driver will use Intel Scalable IOV. If the Intel Scalable IOV requirements are not met, the PF driver will use SR-IOV.

For more information, please refer to the Intel Scalable I/O Virtualization Technical Specification (login may be required).

Intel® Scalable IOV is not available in the kernel driver. Download and install the current Linux ice driver to use this feature. Refer to the Intel Download Center section for where to download the current driver.

Requirements:

- Your system platform must support Intel Scalable IOV
- A network device based on an Intel® Ethernet 800 Series controller
- The host operating system must be a Linux distro using kernel version 5.12 - 5.15
- The host PF driver must be version 1.9.0, or later
- The guest operating system must be Linux
- The guest iAVF driver must be version 4.5.0, or later

Enabling Intel Scalable IOV

You can use Intel’s Ethernet Port Configuration Tool (EPCT) to enable Intel Scalable IOV. If the EPCT tool is not available, you can also enable Intel Scalable IOV through your system’s HII interface (if it has one). The recommended method is to use the EPCT tool.

To enable or disable Intel Scalable IOV using the EPCT tool, use one of these commands:

```
epct -nic=1 -set 'siov enable'
```

or:

```
epct -nic=1 -set 'siov disable'
```

Where `-nic=1` specifies the Intel Ethernet device.

If the EPCT tool is not available, and your system has an HII interface, you can use the HII interface to enable/disable Intel Scalable IOV. Find the “Intel Scalable IOV (Scalable IOV)” setting and select your desired value.

See Ethernet Port Configuration Tool (EPCT) for more information on this tool.

Virtual Machine Queue Offloading

Enabling Virtual Machine Queue (VMQ) offloading increases receive and transmit performance, as the adapter hardware is able to perform these tasks faster than the operating system. Offloading also frees up CPU resources. Filtering is based on MAC and/or VLAN filters.

Each Intel® Ethernet Adapter has a pool of virtual ports that are split between the various features, such as VMQ Offloading, SR-IOV, and Data Center Bridging (DCB). Increasing the number of virtual ports used for one feature decreases the number available for other features. On devices that support it, enabling DCB reduces the total pool available for other features to 32.

Note:

This does not apply to devices based on the Intel® Ethernet X710 or XL710 controllers.

Enabling VMQ Offloading in Windows

For devices that support it, VMQ offloading is enabled in the host partition in the Adapter Settings panel in Intel PROSet ACU or on the Advanced tab of the adapter’s Device Manager property sheet, under Virtualization properties.

Virtualization properties also displays the number of virtual ports available for virtual functions, and allows you to set the distribution of available virtual ports between VMQ and SR-IOV.

Teaming Considerations

- If VMQ is not enabled for all adapters in a team, VMQ will be disabled for the team.
- If an adapter that does not support VMQ is added to a team, VMQ will be disabled for the team.
- Virtual NICs cannot be created on a team with Receive Load Balancing enabled. Receive Load Balancing is automatically disabled if you create a virtual NIC on

a team.

- If a team is bound to a Hyper-V virtual NIC, you cannot change the Primary or Secondary adapter.

See Adapter Teaming for more information on teams.

Virtual Machine Multiple Queues

Virtual Machine Multiple Queues (VMMQ) enables Receive Side Scaling (RSS) for virtual ports attached to a physical port. This allows RSS to be used with SR-IOV and inside a VMQ virtual machine, and offloads the RSS processing to the network adapter. RSS balances receive traffic across multiple CPUs or CPU cores. This setting has no effect if your system has only one processing unit.

Using Intel Network Adapters in a Microsoft* Hyper-V* Environment

When a Hyper-V Virtual NIC (VNIC) interface is created in the host OS, the VNIC takes on the MAC address of the underlying physical NIC (PF, or physical function). The same is true when a VNIC is created on a team or VLAN. Since the VNIC uses the MAC address of the underlying interface, any operation that changes the MAC address of the interface (for example, setting LAA on the interface), will cause the VNIC to lose connectivity. In order to prevent this loss of connectivity, Intel PROSet will not allow you to change settings that change the MAC address.

Note:

- When sent from inside a virtual machine, LLDP and LACP packets may be a security risk. The Intel Virtual Function driver blocks the transmission of such packets.
- The Virtualization setting on the Advanced tab of the adapter's Device Manager property sheet is not available if the Hyper-V role is not installed.

The Virtual Machine Switch

The virtual machine switch is part of the network I/O data path. It sits between the physical NIC and the virtual machine NICs and routes packets to the correct MAC address. Enabling Virtual Machine Queue Offloading in Intel PROSet will automatically enable VMQ in the virtual machine switch. For driver-only installations, you must manually enable VMQ in the virtual machine switch.

Note:

Intel ANS VLANs are not compatible with the Microsoft Hyper-V virtual machine switch. If you want to bind the virtual machine switch to a VLAN, you must create the VLAN from within the Virtual Switch Manager.

Using Intel ANS VLANs

Note:

See Adapter Teaming and Virtual LANs (VLANs) for more information on Intel ANS and VLANs.

If you create Intel ANS VLANs in the host OS, and you then create a Hyper-V Virtual NIC interface on an Intel ANS VLAN, then the Virtual NIC interface must have the same VLAN ID as the VLAN. Using a different VLAN ID or not setting a VLAN ID on the Virtual NIC interface will result in loss of communication on that interface.

Virtual Switches bound to an Intel ANS VLAN will have the same MAC address as the VLAN, which will have the same address as the underlying NIC or team. If you have several VLANs bound to a team and bind a virtual switch to each VLAN, all of the virtual switches will have the same MAC address. Clustering the virtual switches together will cause a network error in Microsoft's cluster validation tool. In some cases, ignoring this error will not impact the performance of the cluster. However, such a cluster is not supported by Microsoft. Using Device Manager to give each of the virtual switches a unique address will resolve the issue. See the Microsoft TechNet article, Configure MAC Address Spoofing for Virtual Network Adapters for more information.

Virtual Machine Queues (VMQ) and SR-IOV cannot be enabled on a Hyper-V Virtual NIC interface bound to a VLAN configured using the VLANs tab in Windows Device Manager.

Using an Intel ANS Team or VLAN as a Virtual NIC

If you want to use a team or VLAN as a virtual NIC, you must follow these steps:

1. Use Intel PROSet to create the team or VLAN.
2. Open the Network Control Panel.
3. Open the team or VLAN.
4. On the General Tab, uncheck all of the protocol bindings and click OK.
5. Create the virtual NIC. (If you check the "Allow management operating system to share the network adapter." box you can do the following step in the host OS.)
6. Open the Network Control Panel for the Virtual NIC.
7. On the General Tab, check the protocol bindings that you desire.

   Note:
   This step is not required for the team. When the Virtual NIC is created, its protocols are correctly bound.

Note:

- These steps apply only to virtual NICs created on a team or VLAN. Virtual NICs created on a physical adapter do not require these steps.
- Receive Load Balancing (RLB) is not supported in Hyper-V. Disable RLB when using Hyper-V.

Command Line for Microsoft Windows Server Core

Microsoft Windows Server Core does not have a GUI interface. If you want to use an Intel ANS Team or VLAN as a Virtual NIC, you must use Microsoft Windows PowerShell to set up the configuration. Use Windows PowerShell to create the team or VLAN.

The following is an example of how to set up the configuration using Microsoft Windows PowerShell:

1. Get all the adapters on the system and store them into a variable:

```
$a = Get-IntelNetAdapter
```

2. Create a team by referencing the indexes of the stored adapter array:

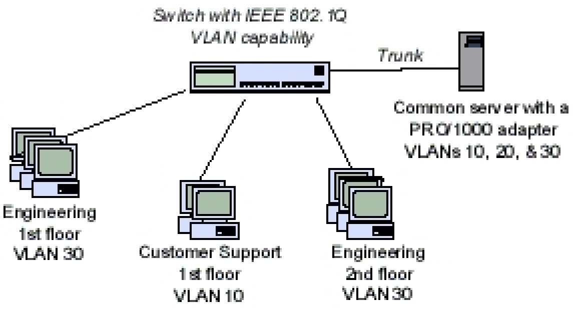
```
New-IntelNetTeam -TeamMembers $a[1],$a[2] -TeamMode
 VirtualMachineLoadBalancing -TeamName "Team1"
```

Virtual LANs (VLANs)

Overview

The term VLAN (Virtual Local Area Network) refers to a collection of devices that communicate as if they were on the same physical LAN. Any set of ports (including all ports on the switch) can be considered a VLAN. LAN segments are not restricted by the hardware that physically connects them.

VLANs offer the ability to group computers together into logical workgroups. This can simplify network administration when connecting clients to servers that are geographically dispersed across the building, campus, or enterprise network. For example:



Typically, VLANs consist of co-workers within the same department but in different locations, groups of users running the same network protocol, or a cross-functional team working on a joint project.

By using VLANs on your network, you can:

- Improve network performance
- Limit broadcast storms
- Improve LAN configuration updates (adds, moves, and changes)
- Minimize security problems
- Ease your management task

VLANs and Intel ANS

For more information on Intel® Advanced Network Services (Intel® ANS), refer to Adapter Teaming.

- Intel ANS is not supported on Microsoft Windows Server 2016 and later.
- Microsoft Windows 10 is the last Windows operating system version that supports Intel ANS. Intel ANS is not supported on Microsoft Windows 11 and later.
- You must install the latest Microsoft Windows 10 updates before you can create Intel ANS Teams or VLANs on Windows 10 systems. Any Intel ANS Teams or VLANs created with a previous software/driver release on a Windows 10 system will be corrupted and cannot be upgraded. The installer will remove these existing teams and VLANs.
    - If you are running Windows 10 Anniversary edition (RS1), you will need to install Intel LAN software v22.1 or newer.
    - If you are running Windows 10 Creators Update (RS2), you will need to install Intel LAN software v22.3 or newer.

  > **Note:**
  >
  > If you are running Windows 10 Creators Update (RS2) and are using Intel LAN software release v22.1 or v22.2, then Intel ANS will not function until you update to the Intel LAN software v22.3 or newer.

- Intel ANS VLANs are not compatible with Microsoft's Load Balancing and Failover (LBFO) teams. Intel PROSet will block a member of an LBFO team from being added to an Intel ANS VLAN. You should not add a port that is already part of an Intel ANS VLAN to an LBFO team, as this may cause system instability.

Other Considerations

- **Configuring SR-IOV for improved network security:** In a virtualized environment, on Intel® Server Adapters that support SR-IOV or Intel® Scalable I/O Virtualization (Intel® Scalable IOV), the virtual function (VF) may be subject to malicious behavior. Software-generated layer two frames, like IEEE 802.3x (link flow control), IEEE 802.1Qbb (priority based flow-control), and others of this type, are not expected and can throttle traffic between the host and the virtual switch, reducing performance. To resolve this issue, and to ensure isolation from unintended traffic streams, configure all SR-IOV or Intel Scalable IOV enabled ports for VLAN tagging from the administrative interface on the PF. This configuration allows unexpected, and potentially malicious, frames to be dropped.

- The VF is not aware of the VLAN configuration if you use LBFO to configure VLANs in a Windows guest. VLANs configured using LBFO on a VF driver may result in failure to pass traffic. You must use Windows Hyper-V on the host to configure VLANs on a Windows guest.
- Intel ANS VLANs are not compatible with the Microsoft Hyper-V virtual machine switch. If you want to bind the virtual machine switch to a VLAN, you must create the VLAN from within the Virtual Switch Manager.
- To set up IEEE VLAN membership (multiple VLANs), the adapter must be attached to a switch with IEEE 802.1Q VLAN capability.
- A maximum of 64 VLANs per network port or team are supported by Intel software.
- Intel ANS VLANs can co-exist with Intel ANS teams (if the adapter supports both). If you do this, the team must be defined first, then you can set up your VLAN.
- You can set up only one untagged VLAN per adapter or team. You must have at least one tagged VLAN before you can set up an untagged VLAN.
- Jumbo Frames are not supported over Intel ANS VLANs under Microsoft Windows 10.

Note:

When using IEEE 802 VLANs, settings must match between the switch and those adapters using the VLANs.

Configuring VLANs in Microsoft Windows

Using Windows PowerShell

To add a VLAN, use the `Add-IntelNetVLAN` cmdlet. For example:

```
Add-IntelNetVLAN -ParentName "Name" -VLANID "1"
```

To remove a VLAN, use the `Remove-IntelNetVLAN` cmdlet. For example:

```
Remove-IntelNetVLAN -ParentName "Name" -VLANID "1"
```

Using Intel PROSet ACU

On the Teaming/VLANs tab, use the VLANs panel.

Using Intel PROSet for Windows Device Manager

This setting is found on the VLANs tab of the device’s Device Manager property sheet.

Note:

Do not use the Network Connections dialog box to enable or disable VLANs. Otherwise, the VLAN driver may not be correctly enabled or disabled.

Note:

- The VLAN ID keyword is supported. The VLAN ID must match the VLAN ID configured on the switch. Adapters with VLANs must be connected to network devices that support IEEE 802.1Q.
- In most environments, a maximum of 64 VLANs per network port or team are supported by Intel ANS.
- Intel ANS VLANs are not supported on adapters and teams that have VMQ enabled. However, VLAN filtering with VMQ is supported via the Microsoft Hyper-V VLAN interface. For more information, see Using Intel Network Adapters in a Microsoft* Hyper-V* Environment.
- You can have different VLAN tags on a child partition and its parent. Those settings are separate from one another, and can be different or the same. The only instance where the VLAN tag on the parent and child MUST be the same is if you want the parent and child partitions to be able to communicate with each other through that VLAN. For more information, see Using Intel Network Adapters in a Microsoft* Hyper-V* Environment.

Wait for Link

This setting determines whether the driver waits for auto-negotiation to be successful before reporting the link state. If this feature is off, the driver does not wait for auto-negotiation. If the feature is on, the driver does wait for auto-negotiation.

If this feature is on and the speed is not set to auto-negotiation, the driver will wait for a short time for link to be established before reporting the link state.

If the feature is set to **Auto Detect**, this feature is automatically set to **On** or **Off** depending on speed and adapter type when the driver is installed. The setting is:

- Off for copper Intel gigabit adapters with a speed of “Auto”
- On for copper Intel gigabit adapters with a forced speed and duplex
- On for fiber Intel gigabit adapters with a speed of “Auto”

To change this setting in Intel PROSet:

This setting is found on the Advanced tab of the device’s Device Manager property sheet or in the Adapter Settings panel in Intel PROSet ACU.

To change this setting in Windows PowerShell, use the `Set-IntelNetAdapterSetting` cmdlet. For example:

```
Set-IntelNetAdapterSetting -Name "<adapter_name>" -DisplayName "Wait for Link"-DisplayValue "Off"
```

Possible values for this setting are:

- On
- Off
- Auto Detect

Tools & Apps

This chapter describes the tools and applications available on Intel® Ethernet devices, adapters, and connections.

Intel® Network Connection Tools

The following tools are supported on all OSs listed in Supported Operating Systems.

- Installing and Uninstalling Intel® Network Connection Tools
- Intel® Ethernet Flash Firmware Utility
- Intel® Dynamic Device Personalization Tool
- Intel® DIAGS
- Ethernet Port Configuration Tool (EPCT)
- Intel® Ethernet NVM Update Tool
- NVM Update Reference Application (NVM URA) for Intel® Ethernet
- NVM Update Package

Windows-only Tools

The following tools are supported only on Microsoft Windows or Windows Server. See Compatibility Notes for Intel Ethernet Cmdlets and Intel PROSet for support limitations.

- About Intel PROSet®
- About Intel® Ethernet Cmdlets

Installing and Uninstalling Intel® Network Connection Tools

This document describes how to install the Intel® Network Connection Tools. See the readme provided for each tool for a detailed explanation of the features available.

Contents

Installing the Tool on Microsoft* Windows*

To install the tools' drivers on Windows, run *install.bat* from the appropriate directory of the install package.

Although the tools are not installed with *install.bat*, the driver that the tool requires is copied into the local machine Windows driver directory. To run the tool, launch a Command Prompt window from the Windows Start Menu. Go to the media and directory where the tool is located and run the utility. The readme files for each tool are found in the same directory as the tool. These tools can be manually installed on the local hard drive in any directory.

The tool uses its own driver file (not the same as the system network driver). If the driver sys file already exists in the drivers directory, *install.bat* may fail to copy. Using the `/y` switch with *install.bat* will override and copy the driver file regardless. However, this can be dangerous if an older version of the driver is being used by another application such as Intel® PROSet. If a driver is already present in the drivers directory, try running the tool from the command prompt. If it runs, then the driver is fine. The tool will not run if the driver version present does not match the driver version expected.

Note that you must have access to the `%systemroot%\system32\drivers` directory. Only the administrator account has these privileges. You must be logged in as administrator or the tools must be run as administrator.

Note that on Windows, any device that is disabled in Device Manager will not be accessible by tools due to no memory resources. You would get an error code *0xC86A800E*. To solve this problem, you can do one of the following:

1. Re-enable the device in Device Manager. Never disable this device when using tools.
2. Install an NDIS device driver for the device and make sure that it does not have a yellow or red bang by it in Device Manager.
3. Delete the device from Device Manager and restart the system. The install new hardware wizard should appear on next reboot. Do not cancel this. Just move the window aside and run the tool(s). Generally, you can click **Cancel** on the wizard but there are some cases where Windows will disable the memory resources, causing you to get back into the same state.

Installing the Tool on EFI

The EFI 1.x tools are not supported in this release.

There is no installation required for EFI tools. The tools can simply be copied from the appropriate directory to the drive that they will run from. The EFI2 binaries are for use with the UEFI Shell 2.X with the UEFI 2.3 HII protocol. EFI2 tools will not run on the EFI Shell 1.X or if the UEFI 2.3 HII protocol is not present.

Note that while EFI supports USB drives, there may be issues running tools from the USB drive. Whether or not there are issues are BIOS specific. If you experience issues, run the tool from hard disk instead.

Installing the Tool on DOS

This tool no longer supports DOS. If you require DOS support, please download a prior version.

Installing the Tool on Linux*

In order to run tools on Linux*, a driver stub must be built and installed on the system. This driver is not related to the network device driver that is used to run the network during live traffic. It is a separate driver used explicitly for tools. Due to the nature of Linux with the number of kernels that can exist, we provide source for the driver module and an install script to build/install it.

The tools support Linux distributions based on kernels 2.6.x. Validation is done randomly on popular distributions such as Red Hat* or Suse*. Configured kernel source that matches the currently installed kernel is required. A working GCC is also required. There are some versions of GCC that had a bug which did not support unnamed structures. These versions of GCC are not supported. If you have compilation errors, try updating your version of GCC. If you have linker errors when installing the driver, you should update your kernel; download the latest stable off www.kernel.org and build/install it.

Note that some distributions such as recent Fedora core versions do not ship with Kernel source. You must download, install, and configure the source in order to get the tools’ driver built on this OS. Installing the kernel source RPM does not solve the problem.

This is the installation procedure:

1. Log in as root and create a temporary directory to build the Intel® Network Connection Tools driver.
2. Copy *install* and *iqvlinux.tar.gz* to the temporary directory. There are 2 versions of Linux supported: Linux32 (x86)and Linux_x64 (x64). Copies of the above files exist in the appropriate directory for your platform.
3. CD to the temporary directory and run `./install`. The driver has been installed now, so the files in the temporary directory can be removed.
4. Copy the tools that you want to use from the appropriate directory of the CD.

Kernel 4.16 or Higher

On Linux kernel 4.16 and higher, the iomem parameter is set to “strict” by default, which may prevent the tool from accessing the MMIO of the device. Trying to update a device when “strict” is set causes the device to lose link during the update process.

If you wish to update a device without losing link, you can either:

1. Install Linux base drivers (igb or ixgbe) from Release 24.1, or newer.

2. Set the iomem kernel parameter to **relaxed** (i.e., `iomem=relaxed`) and reboot the system before running the update utility.

Installing the Tool on FreeBSD*

In order to run tools on FreeBSD*, a driver stub must be built and installed on the system. This driver is not related to the network device driver that is used to run the network during live traffic. It is a separate driver used explicitly for tools. Due to the nature of FreeBSD with the number of kernels that can exist, we provide source for the driver module and an install script to build/install it.

The tools support FreeBSD distributions version 10.1 and later.

This is the installation procedure:

1. Log in as root and create a temporary directory to build the Intel® Network Connection Tools driver.
2. Copy *install* and *iqvfreebsd.tar* to the temporary directory. There are two versions of FreeBSD supported: FreeBSD32 (x86) and FreeBSD64e (x64). Copies of the above files exist in the appropriate directory for your platform.
3. CD to the temporary directory and run `./install`. The driver has been installed now, so the files in the temporary directory can be removed.
4. Copy the tools that you want to use from the appropriate directory of the CD.

Uninstalling Intel Network Connection Tools

Run the *uninstall.bat* batch file if you need to manually remove the old version (*iqvw<architecture>.sys*) of the Intel Network Connection Tools driver.

On Windows, you also need to manually remove the *iqvsw64e.sys* driver.

Intel® Ethernet Flash Firmware Utility

Contents

Overview

The Intel® Ethernet Flash Firmware Utility, or *BootUtil*, is used to update configurations and program the PCI option ROM on the flash memory of supported Intel PCI and PCI-Express-based network adapters.

Intel Ethernet Flash Firmware Utility replaces existing utilities and provides the functionality of the older *IBAUTIL*, *LANUTIL*, and *FLAUTIL*. Intel Ethernet Flash Firmware Utility supports all the adapters supported by the previous utilities.

Note:

- Legacy iSCSI Boot and CLP images are not supported and cannot be loaded or configured.
- Updating the adapter's flash memory using Intel Ethernet Flash Firmware Utility will erase any existing firmware image from the flash memory.
- If your device is flashed with a discrete FCoE firmware (no other Option ROMs flashed), attempts to update your device will fail. This is expected. The data required for FCoE functionality upgrades is no longer part of the latest upgrade packages. The update will fail and FCoE functionality will continue to work.
- If you update a device that is flashed with an FCoE-enabled combo image (for example, an image that contains the EFI driver, legacy PXE, and FCoE boot functionality), the Option ROM combination will change on update, usually to the same combination without FCoE capabilities. This is expected. The data required for FCoE functionality upgrades is no longer part of the latest upgrade packages. The update will succeed and FCoE functionality will be removed from your device.

Intel provides the following flash firmware in FLB file format for programming to the flash memory:

- Intel® Boot Agent as PXE Option ROM for legacy BIOS http://www.intel.com/support/network/adapter/pro100/bootagent/
- Network Connectivity, UEFI network driver http://www.intel.com/support/network/sb/cs-006120.htm

OEMs may provide custom flash firmware images for OEM network adapters. Please refer to the instructions given by OEMs.

Intel Ethernet Flash Firmware Utility allows the user to flash supported firmware to the adapter from the included *BootIMG.FLB* file. This option ROM includes PXE and UEFI, and the image is programmed to the flash memory at once. Intel Ethernet Flash Firmware Utility will also build the required combo images for supported adapter and program those images to the flash, as well. Since both discrete and combo images are supported, the `-bootenable` command ONLY works on combo images.

PXE+EFI image combinations are supported for all OEM generic adapters, however support is limited to devices which support both technologies as discrete images. However, flash size is a limiting factor, as the image size can change without notice.

The *BootIMG.FLB* file is the container for all the Intel® boot Option ROMs. This file replaces the existing FLB files for PXE and EFI.

Intel Ethernet Flash Firmware Utility without command-line options will display a list of all supported Intel network ports in the system. Intel Ethernet Flash Firmware Utility will also allow the user to enable or disable the flash memory on specific ports by using `-FLASHENABLE` or `-FLASHDISABLE` options in order to control access to the firmware from the system.

Intel Ethernet Flash Firmware Utility allows the user to individually set UEFI and PXE boot configurations using the `-NIC=xx -=-<OPTION>=<VALUE>` options.

Note:

- Intel Ethernet Flash Firmware Utility displays up to 128 adapter ports.
- No configuration settings are supported for the UEFI driver.
- `-NIC`, `-MACADDR`, and `-DEVID` are mutually exclusive. You cannot use more than one of these options per command. Specifying more than one option will cause Intel Ethernet Flash Firmware Utility to fail.
- You cannot enable/disable 64-bit BAR addressing on Intel® Ethernet Controller X550 Series devices.
- Intel Ethernet Flash Firmware Utility may indicate that a device is in the “Pending Reboot” state. No update or configuration changes are possible on a device in this state. Please reboot your system.

Check the Intel Customer Support (http://support.intel.com) website for the latest information and component updates.

Note:

If the tool displays the error, *Unable to load the driver. Please close all other applications and try again*, you have a mix of old and new versions of the utility tool on your system.

Quit all open applications and retry your operation. If the issue persists:

1. Download the latest version of the utility tools.
2. Run the `uninstall` script to remove the old version of the tool driver.
3. Run the `install` script from the downloaded tools package.
4. Retry your operation. You may also need to download and install the latest Intel Ethernet driver or Intel PROSet package for your device.

Installation

See Installing and Uninstalling Intel® Network Connection Tools for installation instructions.

Running the Utility

The syntax for issuing Intel Ethernet Flash Firmware Utility command-line options is:

```
bootutil -[option] or -<OPTION>=<VALUE>
```

Using the /? option will display a list of supported command line options.

See Examples and Command Line Options below for more information.

Command Line Options

Intel Ethernet Flash Firmware Utility accepts one executable option and its associated non-executable options in an execution. If conflicting executable options are specified, Intel Ethernet Flash Firmware Utility exits with an error.

The options, where applicable, are listed in logical groups:

- General Options
- Power Management Options
- Adapter Flash Enable/Disable Options
- Flash Programming Options
- PXE Boot Configuration Options

> **Note:**
>
> If you run Intel Ethernet Flash Firmware Utility without any command-line options, the utility displays a list of all supported Intel network ports found in the system.

General Options

-?, -HELP

Displays a list of command-line options.

-ALL

Selects all supported network ports found in the system.

-64d

Disables 64-bit BAR (Base Address Register) addressing on the selected port. This restricts the system BIOS to assigning 32-bit BARs to the port.

-64e

Enables 64-bit BAR addressing on the selected port.

-*BLINK*

Blinks the LED on the selected port for ten seconds.

-DEVID=[Device ID]

Selects the network port specified by `[Device ID]`. `[Device ID]` is a 4-digit hexadecimal number, prepended by 0x.

-*EXITCODES*

Displays exit codes returned to the OS environment by Intel Ethernet Flash Firmware Utility.

-E

Enumerates all supported network ports in the system.

-FILE=[FILENAME]

Specifies the file name for `-SAVEIMAGE`, `-RESTOREIMAGE`, `-SAVECONFIG` , `-RESTORECONFIG`, `-UP` or `-UPDATE` options.

-*IMAGEVERSION*, -*IV*

Displays the firmware image type and version number from the specified source. If no source is specified, displays the image type and version number from the local *BootIMG.FLB* file. Use `-FILE` to specify a different file. Use `-NIC` to specify a port. Use `-ALL` to display the firmware image types and version numbers of images loaded onto all ports in the system. If a port does not have an image loaded, or is not an Intel® Ethernet port, the port will not be included in the output.

-MACADDR=[MAC address string]

Selects the network port specified by `[MAC address string]`. [MAC address string] is a 12-digit number. Separators (`:` or `-`) are allowed but not required.

-NIC=XX

Selects a specific network port in the system.

-*NODPL*

Prevents Intel Ethernet Flash Firmware Utility from displaying the PCI device location.

-O=[FILENAME]

Saves a list of all NICs and their MAC addresses to the specified file. Must be used with `-E`.

`-P`

Causes Intel Ethernet Flash Firmware Utility to pause screen after displaying every 25th line. In pause mode, the last line of the screen will contain information that the screen has been paused and that the user can press **Enter** to continue.

Power Management Options

`-WOLENABLE, -WOLE`

Enables Wake on LAN functionality on the selected port.

`-WOLDISABLE, -WOLD`

Disables Wake on LAN functionality on the selected port.

Adapter Flash Enable/Disable Options

-BOOTENABLE=[PXE|DISABLED]

Selects flash firmware type to enable or disable on the selected port. The combo option ROM could have all capabilities. This command **will NOT work** on monolithic Option ROMs.

`-BOOTENABLE=DISABLED` disables PXE functionalities for legacy BIOS.

> **Note:**
>
> This manual command is not applicable to the UEFI network driver. The UEFI network driver is enabled by default and loaded automatically unless network ports are already used by other flash firmware.

`-FLASHENABLE, -FE`

Enables the capability to run flash firmware on the selected adapter.

> **Note:** This parameter is not supported on the following adapters:
>
> - Intel® Gigabit ET Quad Port Mezzanine Card
> - Intel® Ethernet X520 10GbE Dual Port KX4 Mezz
> - Intel® Ethernet X520 10GbE Dual Port KX4-KR Mezz

`-FLASHDISABLE, -FD`

Disables the capability to run flash firmware on the selected adapter.

> **Note:** This parameter is not supported on the following adapters:

- Intel® Gigabit ET Quad Port Mezzanine Card
- Intel® Ethernet X520 10GbE Dual Port KX4 Mezz
- Intel® Ethernet X520 10GbE Dual Port KX4-KR Mezz

Flash Programming Options

-QUIET

Runs Intel Ethernet Flash Firmware Utility without confirmation prompts. This option is used with `-IMAGEVERSION`, `-UP`, `-UPDATE`, `-RESTOREIMAGE`, or `-HELP` option.

-RESTOREIMAGE, -RI

Restores a previously-saved firmware image to the same network adapter using the `-NIC` option. The source file is specified by the `-FILE` option.

-SAVEIMAGE, -SI

Saves the existing firmware image in the flash memory of the adapter specified by the `-NIC` option to a disk file. The destination file is specified by the `-FILE` option.

-UP, -UPDATE=[ROMType] [-FILE=Filename]

Updates the flash memory of the adapter specified by the `-NIC` option with the given FLB flash firmware image file. If `-QUIET` is not specified, the user is prompted to confirm overwriting the existing firmware image before upgrading. If a file name is not provided, the utility looks for the default *bootimg.FLB* file which contains all image types.

The following examples use PXE, but could also be *efi* or *efi64*:

```
Bootutil -Nic=x -up=pxeBootutil -Nic=x -up=pxe -file
=pxeimg.flb
```

If your adapter supports a combo option rom, the following command option automatically detects the supported combo image and programs the adapter with that image:

```
-up=combo
```

`-up=combo up=pxe+efi` is valid.

PXE Boot Configuration Options

All of these options can be used with `-NIC` or `-ALL`:

-DEFAULTCONFIG, -DEFCFG

Sets the PXE configuration of the selected NIC back to default settings.

-MODE

This parameter should not be used except as advised by Intel Support personnel.

-TITLEENABLE, -TLE

Enables Initial Title Message.

-TITLEDISABLE, -TLD

Disables Initial Title Message.

Examples

This section includes examples of how to enter some typical Intel Ethernet Flash Firmware Utility command lines.

To get help descriptions:

```
BootUtil -?
```

To enable the flash firmware on the first network adapter for the system to be capable of executing the flash firmware:

```
BootUtil -NIC=1 -FLASHENABLE
```

To disable the flash firmware on all the network adapters:

```
BootUtil -ALL -FD
```

To display BootUtil FLB flash firmware types and versions:

```
BootUtil -IMAGEVERSION
```

To update all ports of a supported NIC with PXE, use either of the following:

```
BootUtil -UP=PXE -ALL# Assumes input file is bootimg.flb
BootUtil -UP=PXE -ALL -FILE=BOOTIMG.FLB# Uses an explicit user-specified file
```

To update a combo image on supported adapter (e.g., pxe+efi). This command will succeed if the PXE+EFI combination is supported on NIC #2. If not, an error is displayed to the user:

```
BootUtil -UP=Combo -NIC=2 -FILE=BOOTIMG.FLB
```

Note: The `-UP` and `-UPDATE` commands are equivalent and interchangeable.

To enable PXE firmware on the third network port in the system:

```
BootUtil -BOOTENABLE=PXE -NIC=3
```

Note:

This command will work only if PXE is part of a combo Option ROM and not a discrete Option ROM.

To disable the firmware on the second network port in the system:

```
BootUtil -NIC=2 -BOOTENABLE=DISABLED
```

Exit Codes

BootUtil returns an exit code to the OS environment. Possible exit codes are listed in the following table:

Code Name	Value
0	All Intel Ethernet Flash Firmware Utility operations completed successfully
1	Bad command-line parameter
3	No supported network port detected 2 Invalid network port selected
4	FLASH operation failed
5	EEPROM read operation failed
6	EEPROM write operation failed

Code Name	Value
7	File content is missing or corrupt
8	Operator termination
9	LOM not supported for selected operation
10	Network port is not bootable
11	FLASH memory is disabled for flash firmware
12	FLASH memory too small for selected firmware
14	System memory allocation failed
15	File operation failed

Note:

EFI versions of this tool may report an incorrect error code when no adapter is installed. This is due to a known limitation in the UDK2015 UEFI Development Kit (UDK) build environment.

Intel® Dynamic Device Personalization Tool

Contents

Overview

The Intel® Dynamic Device Personalization Tool is a utility for detecting the Dynamic Device Personalization (DDP) profiles loaded on the following devices:

- Intel® Ethernet 800 Series
- Intel® Ethernet X710 Series

Dynamic Device Personalization (DDP) allows you to change the packet processing pipeline of a device by applying a profile package to the device at runtime. Profiles can be used to, for example, add support for new protocols, change existing protocols, or change default settings. See the Linux ice or i40e driver README for more information.

> **Note:** The Intel Dynamic Device Personalization Tool requires the following:
>
> - ice devices: NVM version 2.2 or later
> - i40e devices: Linux i40e base driver version 2.7.27 or later.

The output of this tool can be sent to the console or to an XML or JSON file.

Installation

See Installing and Uninstalling Intel® Network Connection Tools for installation instructions.

Running the Utility

When run with no parameters, the Intel Dynamic Device Personalization Tool displays information about all supported devices.

```
ddptool [parameters] [argument]
```

Command Line Parameters

-a

Displays information about all functions for all supported devices. This parameter cannot be used with `-f`.

-h, --help, -?

Displays command line help.

-f *FILENAME*

Displays information about the profile contained in the specified package file. This parameter cannot be used with `-a`, `-i`, or `-s`.

-i *DEVNAME*

Displays information for the specified network interface name. Running Intel Dynamic Device Personalization Tool without any parameters will provide network interface names. This parameter cannot be used with `-f`.

-j *FILENAME*

Outputs in JSON format to a file. If `FILENAME` is not specified, output is sent to standard output.

-l

Silent mode

-s *LOCATION*

Displays information about the device located at the specified PCI location `-s dddd:bb:ss.f` (all numbers in hex), where:

d:

domain

b:

bus

s:

slot

f:

function

This parameter cannot be used with `-f`.

-v

Displays the version of the Intel Dynamic Device Personalization Tool.

-x *FILENAME*

Outputs in XML format to a file. If `FILENAME` is not specified, output is sent to standard output.

Examples

The following is an example of the Intel Dynamic Device Personalization Tool output:

```
# ddptoolIntel(R) Dynamic Device Personalization ToolDDP
Tool version 1.0.1.4Copyright (C) 2019 - 2020 Intel Corp
oration.NIC  DevId D:B:S.F       DevName         TrackId
  Version      Name==== ===== ============  ============
=== ======== ============ ================================
1)   1588  0000:01:00.0  enp1s0f0       80000008 1.0.3.
0      GTPv1-C/U IPv4/IPv6 payload2)   1588  0000:01:00.
1  enp1s0f1        80000008 1.0.3.0      GTPv1-C/U IPv4/
IPv6 payload3)   1584  0000:03:00.0  enp3s0f2         -
      -            Unsupported FW version4)   1584  0000
:05:00.0  enp5s0f3        DE010001 1.0.0.0      MPLSoUDP
/GRE tunnels demo5)   1584  0000:06:00.0  N/A
  80000008 1.0.3.0      GTPv1-C/U IPv4/IPv6 payload6)
1584  0000:07:00.0  enp8s0f4        -        -
  No profile loaded7)   1593  0000:60:00.0  ens261f0
    C0000002 1.3.17.0     ICE COMMS Package8)   1593  00
00:60:00.1  ens261f1        C0000002 1.3.17.0     ICE CO
MMS Package
```

To send the output to the console in JSON format:

```
# ddptool -jIntel(R) Dynamic Device Personalization Tool
 DDPTool version 1.0.1.4Copyright (C) 2019 - 2020 Intel
Corporation.{    "DDPInventory": [      {         "devic
e": "158B",         "address": "0000:02:00.0",         "
name": "enp1s0f0",          "display": "Intel(R) Ethernet
 Network Adapter XXV710-2",         "DDPpackage": {
    "track_id": "80000008",         "version": "1.0.3.0"
,          "name": "GTPv1-C/U IPv4/IPv6 payload"
}      },      {         "device": "1584",         "addr
ess": "0000:86:00.0",         "display": "Intel(R) Ether
net Network Adapter XL710-Q1",         "DDPpackage": [
         {             "track_id": "DE010001",
   "version": "1.0.0.0",             "name": "MPLSoUDP/G
RE tunnels demo"           },           {             "t
rack_id": "DE010002",             "version": "1.0.0.0",
            "name":"MPLSoUDP/GRE PTYPE demo"           }
,           {             "track_id": "DE010003",
      "version": "1.0.0.0",             "name": "MPLSoUD
P/GRE decap demo"           }          ]       }   ]}
```

To send the output to the *ddpout.xml* file:

```
ddptool -x ddpout.xml
```

The *ddpout.xml* file will look like the following:

```
<DDPInventory lang="en">   <Instance device="1589" locat
ion="0000:02:00.0" name="enp1s0f0" display="Intel(R) Eth
ernet Converged Network Adapter X710-T">      <DDPpackag
e track_id="80000008" version="1.0.3.0" name="GTPv1-C/U
IPv4/IPv6 payload"></DDPpackage>   </Instance>   <Instan
ce device="1589" location="0000:02:00.1" name="enp1s0f1"
 display="Intel(R) Ethernet Converged Network Adapter X7
10-T">      <DDPpackage track_id="80000008" version="1.0
.3.0" name="GTPv1-C/U IPv4/IPv6 payload"></DDPpackage>
 </Instance>   <Instance device="1589" location="0000:02
:00.2" name="enp1s0f2" display="Intel(R) Ethernet Conver
ged Network Adapter X710-T">      <DDPpackage track_id="
80000008" version="1.0.3.0" name="GTPv1-C/U IPv4/IPv6 pa
yload"></DDPpackage>   </Instance>   <Instance device="1
589" location="0000:02:00.3" name="enp1s0f3" display="In
tel(R) Ethernet Converged Network Adapter X710-T">
<DDPpackage track_id="80000008" version="1.0.3.0" name="
GTPv1-C/U IPv4/IPv6 payload"></DDPpackage>   </Instance>
   <Instance device="1589" location="0000:03:00.0" name=
"enp1s0f4" display="Intel(R) Ethernet Converged Network
Adapter X710-T">      <DDPpackage track_id="80000008" ve
rsion="1.0.3.0" name="GTPv1-C/U IPv4/IPv6 payload"></DDP
package>   </Instance></DDPInventory>
```

-f Examples

The following are examples of using the `-f` parameter.

Example 1:

```
# ddptool -f gtk.pkgo============  =================  ==
======  ============================File Name     TrackI
d            Version   Name============  ===============
==  ========  ============================gtk.pkgo
80000008          1.0.3.0   GTPv1-C/U IPv4/IPv6 payload
============  =================  ========  =============
===============
```

Example 2:

```
ddptool -f gtk.pkgo -j{   "DDPInventory": {      "file_n
ame": "gtp.pkgo"      "DDPpackage": {         "track_id"
: "80000008",         "version": "1.0.3.0",         "nam
e": "GTPv1-C/U IPv4/IPv6 payload"      }   }}
```

Example 3:

```
ddptool -f gtk.pkgo -x<?xml version="1.0" encoding="UTF-
8"?><DDPInventory lang="en">  <Instance file="gtp.pkgo">
    <DDPpackage track_id="80000008" version="1.0.3.0" na
me="GTPv1-C/U IPv4/IPv6    payload"></DDPpackage>  </Ins
tance></DDPInventory>
```

Exit Codes

Upon exit, when possible, the Intel Dynamic Device Personalization Tool reports an overall status code to indicate the results of the operation. In general, a non-zero return code indicates an error occurred during processing.

Value	Description
0	Success
1	Bad command line parameter
2	An Internal error has occurred
3	Insufficient privileges to run the tool

Value	Description
4	No supported adapter found
5	No driver available
6	Unsupported base driver version
7	Cannot communicate with one or more adapters
8	Lack of DDP profiles on all devices
9	Cannot read all information from one or more devices
10	Cannot create output file
11	Cannot find specific devices
12	Cannot parse the DDP package file

Intel® DIAGS

Contents

Overview

Intel's diagnostic software lets you test the adapter to see if there are any problems with the adapter hardware, the cabling, or the network connection. You can also use diagnostics to isolate problems during troubleshooting. DIAGS is a single program that operates identically under multiple operating systems.

These utilities are designed to test hardware operation and confirm the adapter’s ability to communicate with another adapter in the same network, primarily confirming cabling. They should not be used to judge performance.

They can test the adapter whether or not there is a responder on the network. In order to do a thorough test, however, you should set-up a second system on the network as a responder prior to starting a test.

Command Line Parameters

DIAGS supports the following command line parameters:

VERSION:

Print Diags version information.

?:

Print command line usage instructions.

> **Note:**
>
> In general for any OS, this utility may unload (or set to idle) the drivers for the adapters selected through this utility, if they were loaded.

Ethernet Port Configuration Tool (EPCT)

Contents

Overview

The Ethernet Port Configuration Tool (EPCT) is a command line utility that allows users to change the link type of a device. The supported types are defined within the adapter's NVM. This utility displays only the devices that potentially support reconfiguration.

Note:

- A reboot is required to apply configuration changes.
- You may lose link if you change the link type of your device from any port option that contains three to seven ports to a port option that enables multi-lane interfaces, such as 2x100G, 2x50G or 1x100G. One of the following methods may resolve the issue:
  - Use the utility to change the port option to 8x10G; reboot your system; change to your originally desired configuration.
  - Completely power cycle your system.
- If the tool displays an error such as "Access error" or "Cannot initialize port," you may be using an outdated driver. Please download the latest driver from https://support.intel.com and try again.

Supported Operating Systems

- Microsoft Windows Server*
- Linux* Kernel
- Red Hat* Enterprise Linux*
- SUSE* Linux Enterprise Server
- UEFI
- VMware* ESXi*
- FreeBSD

Note:

On systems running Linux, FreeBSD, or ESXi, the base driver must be present for EPCT to function correctly.

Intel® Scalable I/O Virtualization Support

Intel® Scalable I/O Virtualization (Intel® Scalable IOV) allows you to share a physical device across multiple virtual machines and applications. See Intel® Scalable I/O Virtualization Support in the user guide for an explanation of this feature.

You can use this Ethernet Port Configuration Tool (EPCT) to enable Intel Scalable IOV. If EPCT is not available, you can also enable Intel Scalable IOV through your system's HII interface (if it has one). The recommended method is to use the EPCT tool.

To enable or disable Intel Scalable IOV using the EPCT tool, use one of these commands:

```
epct -nic=1 -set 'siov enable'
```

or:

```
epct -nic=1 -set 'siov disable'
```

Where `-nic=1` specifies the Intel Ethernet device.

Installation

See Installing and Uninstalling Intel® Network Connection Tools for installation instructions.

Running the Utility

Use the following syntax to run EPCT:

```
epct [options]
```

Using the `/?` option will display a list of supported command line options.

See Options below for this tool’s supported parameters.

Note:

If the tool displays the error: “Unable to load the driver. Please close all other applications and try again”, you have a mix of old and new versions of the utility tool on your system. Quit all open applications and retry your operation. If the issue persists:

1. Download the latest version of the utility tools.
2. Run the uninstall script to remove the old version of the tool driver.
3. Run the install script from the downloaded tools package.
4. Retry your operation.

You may also need to download and install the latest Intel Ethernet driver or Intel® PROSet package for your device.

Basic Usage Examples

The following show some basic usage examples for EPCT:

```
epct -devicesepct -devices brandingepct -nic=1 -getepct
-nic=1 -set 2x1x50epct -help
```

See Detailed Usage Examples below for additional examples.

Options

The Ethernet Port Configuration Tool can be run with any of the following command line options.

Note:

- You can use the slash / character in place of the dash - character.
- All options are case-sensitive.

-h, -help, -?

Displays help for the command or the parameter.

You can also use the following to display help for the specified parameter:

```
-h <parameter name>-help <parameter name>-? <paramet
er name>
```

-devices [branding] <option>

Displays supported devices present in the system. If `branding` is specified, then the branding view is displayed. If an option is specified, the value for that setting is also displayed.

Possible values for `<option>` are:

siov:

Displays the status of Intel Scalable IOV.

tx_balancing:

Displays the device’s transmit balancing setting.

-get <option>

Displays the configuration for the specified option on the device specified by `-nic` .

If an option is not specified, `-get` displays the port configuration for the specified device.

- `Active` indicates the currently used configuration.
- `Pending` indicates the configuration the device will use after the system reboots.

Possible values for `<option>` are:

siov:

Displays the status of Intel Scalable IOV.

tx_balancing:

Displays the device’s transmit balancing setting.

max_pwr:

Displays the maximum power options of the QSFP/SFP cages.

See -get Examples below for example usage of this option.

-location <SS:BBB>

Specify a device for this instance of the tool to update, where `<SS:BB>` means:

SS:

The PCI segment of the desired device.

BBB:

The PCI bus of the desired device.

Do not specify `-nic` in the same command as `-location`.

-nic=<device_index>

Selects the device at the specified index. Do not specify `-location` in the same command as `-nic`.

-set <option>

Configures the selected device with the specified option. Valid values for `<option>` are:

siov enable|disable:

Enables or disables Intel Scalable IOV.

tx_balancing enable|disable:

Enables or disables the transmit balancing feature, to improve transmit performance.

max_pwr X:

Sets the maximum allowed power to `X` for the QSFP/SFP cage.

<port configuration string>:

Specifies the configuration to set for the desired quad, port, or speed.

A port configuration string is defined as:

- `QxPxS` - if all port speeds are the same across both quads and all lines, or

- `P1xS1-P2xS2` - if each quad has a specific speed, or
- `P11xS11+<…>+P1nxS1n-P21xS21+<…>+P2mxS2m`

  Where:

  Q:

  The desired quad number.

  P:

  The desired port number.

  S:

  The desired port speed.

  n:

  The desired port/speed combination for quad 0.

  m:

  The desired port/speed combination for quad 1.

  For example:

  ```
  epct -nic=1 -set 2x25+2x10-2x10
  ```

See -set Examples below for example usage of these options.

Note: A reboot is required after changing the port settings.

Detailed Usage Examples

Note:

Some configurations shown in the examples below may not apply to all adapters.

The following examples show the tool's `-devices` option, the `-get` option, and the `-set` option.

-devices Examples

Typing the following at the prompt:

```
epct -devices
```

Will display:

```
NIC Seg:Bus:Fun   Ven-Dev   Connector Ports Speed    Qua
ds  Lanes per PF=== ============= ========= ========= ==
```

```
=== ======== ====== ============1)  000:012       8086-1
583 QSFP      2     50 Gbps  Single  12)  000:015
8086-1584 QSFP      4     25 Gbps  Single  13)  000:017
      8086-1589 SFP       4     25 Gbps  Single  14)  00
0:030:00-01 8086-1588 QSFP      2     25 Gbps  Single  1
    000:030:02-05 8086-1587 QSFP      4     10 Gbps  Sin
gle  1
```

The following shows how to use the `branding` option:

```
epct -devices brandingNIC Seg:Bus Ven-Dev   Mode
Adapter Name=== ======= ========= =========== ==========
====================================1)  000:012 8086-1583
 N/A         Intel(R) Ethernet Converged Network Adapter
                                 XL7102)  000:029 8086-
1583 4x10        Intel(R) Ethernet Converged Network Ada
pter                                 XL7103)  000:129 8
086-1599 2x25,2x10-  Intel(R) Ethernet Controller E810-X
XV for                   2x10        backplane4)  000
:132 8086-159A 2x50        Intel(R) Ethernet Controller
E810-XXV for                              QSFP
```

-get Examples

Typing the following at the prompt:

```
epct -nic=1 -get
```

Will display:

```
Available Port Options:=================================
=========================================        Port
                          Quad 0           Quad 1Option
 Option (Gbps)                    L0  L1  L2  L3   L4  L
5  L6  L7======= =============================    ======
========== ================        4x25
         -> 25  25  25  25   -   -   -   -        2x1x10
0                      -> 100 -   -   -    100 -   -
-Active  2x2x25                     -> 25  25  -   -
   25  25  -   -        2x50                        ->
 50  -   50  -    -   -   -   -Pending 2x1x50
```

```
             -> 50  -   -   -    50  -   -   -        2x
4x10                        -> 10  10  10  10   10  10
10  10        2x25+2x10-2x10               -> 25  25  1
0  10   10  10  -   -        1x10+1x25+1x10+1x25-2x10
   -> 10  25  10  25   10  10  -   -
```

To display the current setting for the transmit balancing feature on a specific device:

```
epct -nic=1 -get tx_balancing
```

To display the minimum and maximum power allowed for the QSFP/SFP cage on a specific device:

```
epct -nic=1 -get max_pwr
```

For example, the above will display:

```
Seg:Bus:Fun    Ven-Dev      MIN PWR ALLOWED    MAX PWR A
LLOWED    MAX PWR SET===========    =======      =======
========    ===============    ===========000:012
 8086-1583    1                  5.5                3.5
```

-set Examples

To set two ports to 50Gbps (first port starts with lane L0 in quad 0 and second with lane L4 in quad 1):

```
epct -nic=1 -set 2x1x50
```

To set the first and second ports to 25Gbps (respectively lanes L0 and L1 in quad 0), the third and fourth port to 10Gbps (respectively lanes L2 and L3 in quad 0), and the fifth and sixth port to 10Gbps (respectively lanes L4 and L5 in quad 1):

```
epct -nic=1 -set 2x25+2x10-2x10
```

To enable the transmit balancing feature on a specific device:

```
epct -nic=1 -set 'tx_balancing enable'
```

To set the maximum allowed power for a QSFP cage:

```
epct -nic=1 -cage=2 -set 'max_pwr 2.5'
```

Note:

- A reboot is required after changing the port settings.
- In Windows, you must use double quotation marks instead of single quotation marks. For example:

```
epct -nic=1 -set "tx_balancing enable"epct -nic=1
 -cage=2 -set "max_pwr 2.5"
```

Exit Codes

Upon exit, when possible, the EPCT reports an overall status code to indicate the results of the operation. In general, a non-zero return code indicates an error occurred during processing.

Value	Description
0	Success
1	No supported adapter found
2	Insufficient privileges to run the tool
3	No driver available
4	Unsupported base driver version
5	Bad command line parameter
6	Invalid adapter selected
7	Unsupported ports configuration selected

Value	Description
8	Adapter does not support ports configuration
9	Memory allocation error
10	Adapter access error
13	Cannot set new port option. Pending reboot detected
14	The device is in recovery mode
15	The requested feature is not supported on this device. You may get this error if your system/device/operating system combination does not support the option you tried to set.
25	Setting value is out of range

Note:

EFI versions of this tool may report an incorrect error code when no adapter is installed. This is due to a known limitation in the UDK2015 UEFI Development Kit (UDK) build environment.

Troubleshooting

Issues with breakout cables

Use of the 4x25 quad breakout or 1x100 port option will only work on Port 2 of Intel® Ethernet Network Adapter E810-C-Q2 products.

Unexpected PF mapping

Physical Function (PF) to physical lane mapping is dependent upon the hardware and may change across different MAC port options. This may be most noticeable when using a breakout cable, in which case the labeling on the cable may not align with the device port assignment.

For example, inserting a 4-port breakout cable into either QSFP cage and configuring the device in 2x2x25 mode may result in the two active PFs being assigned to the third and fourth cables of the breakout connector.

Possible misconfiguration of the Ethernet port

You may see an informational message stating that a potential misconfiguration of the Ethernet port was detected. This is to alert you that your device is being underutilized. If this was intentional, you may ignore this message. For example, setting your Intel® Ethernet Network Adapter E810-C-Q2 to 2x2x25 is valid, but it does not use the full capabilities of the device. If you see this message, and the configuration was not intentional, you may use EPCT to correct the configuration.

Intel® Ethernet NVM Update Tool

Contents

Overview

The Intel® Ethernet NVM Update Tool is a utility for updating the non-volatile memory (NVM) of Intel® Ethernet devices. It can also update the PHY firmware (FW) on some devices.

Supported Operating Systems

- Microsoft* Windows*
- Microsoft Windows Server*
- Linux* Kernel
- Red Hat* Enterprise Linux*
- SUSE* Linux Enterprise Server
- UEFI
- VMWare* ESXi*
- FreeBSD

Note:

- On systems running Linux, FreeBSD, or ESXi, the base driver must be present for the Intel Ethernet NVM Update Tool to function correctly.
- On Microsoft Windows systems, if the driver required to run the tool is not present on the system, it will automatically be installed when the Intel Ethernet NVM Update Tool is run.

Available Modes

The Intel Ethernet NVM Update Tool operates in one of these modes, which are described in the following subsections:

- Interactive mode
- Inventory mode
- Update mode

Interactive Mode

Interactive mode provides an interactive session of the Intel Ethernet NVM Update Tool. Enter Interactive mode by *not* specifying the `-u` or `-i` parameters when you run the Intel Ethernet NVM Update Tool. All other parameters work with Interactive mode.

Interactive mode runs an inventory on the system and displays the available devices. You can then follow the prompts to specify which device or devices to update.

Inventory progress and debug information are not provided by default. These can be logged to a file by specifying the `-l <filename>` option or displayed on the console (if no file name is specified).

Note:

A configuration file named *nvmupdate.cfg* must be present in the same directory as the Intel Ethernet NVM Update Tool for Interactive mode to function correctly.

Inventory Mode

Inventory mode attempts to discover all Intel Ethernet devices present in the system and report the status of each device found. The Intel Ethernet NVM Update Tool performs an integrity check on each device's NVM image to determine whether a correct image (i.e., valid pointers and checksum) is present. If you specify a configuration file, Inventory mode will only attempt to discover those devices listed in that configuration file. Inventory mode will also report whether an update to the current NVM image is available.

Inventory progress and debug information are not provided by default. These can be logged to the file by specifying `-l <filename>` option or displayed on the console (if no file name provided).

Inventory mode writes the following information to the XML-formatted results file:

- Vendor ID - The vendor ID (typically 8086) in hexadecimal format
- Device ID - The device ID in hexadecimal format
- Subdevice ID - The subdevice ID in hexadecimal format
- Subvendor ID - The subvendor ID in hexadecimal format
- Bus - PCI numbered bus identifier
- Dev - PCI numbered device identifier
- Function - PCI numbered function identifier
- Name - Branding string for the device
- EEPROM/NVM - EtrackID version
- ID EEPROM - ID EEPROM version
- OROM - subcomponents and version
- Module type - Displays whether a PHY NVM update is available
- Update availability - Displays whether an NVM update is available
- LAN MAC Address
- SAN MAC Address
- Alternative MAC Address
- Vital Product Data (VPD) variables in a `Key:Value` format
- SREV - The module's security revision number in hexadecimal format
- MinSREV - The module's minimum security revision number in hexadecimal format

Note:

SREV and MinSREV values will only be captured if `-optinminsrev` is specified in the command line or if `MINSREV:TRUE` is specified in the configuration file.

Update Mode

Update mode attempts to discover all devices listed in the configuration file. You must specify a configuration file in order to use Update mode. For each discovered device, the Intel Ethernet NVM Update Tool compares the currently installed NVM image version with a list of update versions in the configuration file. An update is only applied if a newer image is available (with some exceptions described later). The Intel Ethernet NVM Update Tool writes the status of the update (success or failure) to the results file.

The Intel Ethernet NVM Update Tool only updates the entire EEPROM/NVM image. It does not update specific bits/words. Vital Product Data (VPD), end-user, and device-specific data are preserved.

For each device specified in the configuration file, Update mode writes the Inventory mode information (listed previously) to the results file, plus the following:

- Version - the reported NVM version is the post-update version
- Status - update success or failure

Note: Interrupting the update process may damage your device.

Firmware Recovery Mode

When a device is in Firmware Recovery mode it will not pass traffic or allow any configuration; you can only attempt to recover the device’s firmware. A device will enter Firmware Recovery mode if it detects a problem that requires the firmware to be reprogrammed.

Note:

- Before starting the recovery process, make sure that your operating system, drivers, and tools have been installed properly. You must use the out-of-tree driver. Using the in-box or kernel driver may result in a “*Cannot initialize port*” warning.
- You must power cycle your system after using Recovery Mode to completely reset the firmware and hardware.

See Firmware Recovery Mode in the Device Features chapter of the user guide for more information.

Pending Reboot State

When a device is in the Pending Reboot state, it is waiting for a system reboot. You cannot modify or update a device when it is in the Pending Reboot state. A device may enter the Pending Reboot state after an NVM update.

Firmware Minimum Security Revision

Many Intel Ethernet devices support a Security Revision (SRev) and a Minimum Security Revision (MinSRev). The Intel Ethernet NVM Update Tool allows you to view or update the MinSRev on supported devices.

See Firmware Security in the Device Features chapter in the user guide for additional background on these features.

Updating Multiple Devices Concurrently

You can use the `-location` command line parameter to run multiple instances of the tool, each updating a specific device. You must use the `-location` parameter with either the `-u` or `-i` parameter. The `-location` parameter is not available in Interactive mode.

Updating Intel Ethernet 800 Series Devices in a Linux System

This applies to Intel® Ethernet 800 Series devices in systems running Linux Stable Kernel 5.1 (and above), RHEL 8 (and above), and SLES 15 (and above). For devices in these situations, the Intel Ethernet NVM Update Tool will use the devlink interface to update the device NVM. In this situation, the `-p` command line parameter (suppress update of the OROM) does not function.

You can force the tool to use the ioctl interface, instead of the devlink interface, by specifying `-if ioctl` in the command line. For example:

```
# nvmupdate64e -u -p -l nvmupdatelog.txt -c nvmupdate.cfg -if ioctl
```

Updating the NVM with a DPDK Driver

If all of the following are true:

- You want to update or inventory one of the following:
  - Devices based on the Intel® Ethernet 800 Series
  - Devices based on the Intel® Ethernet 700 Series
- You are using the DPDK driver
- The i40e/ice device driver is not bound to any port on the device.

Then you must do the following:

1. Bind the kernel driver to the device.
   - Make sure the i40e/ice kernel driver is installed.
   - Use **lspci** to discover the PCI location of the device port you want to update/inventory (in <Bus:Device.Function> format (e.g. 04:00.0))
   - Bind the port with the kernel driver:

     ```
     usertools/dpdk-devbind.py -b <i40e|ice> <B:D.F>
     ```

2. Run Nvmupdate.
   - Update mode example:

```
nvmupdate -u -l -c nvmupdate.cfg
```

- Inventory mode example:

  ```
  nvmupdate -i -l -c nvmupdate.cfg
  ```

3. Reboot the system, if required.
4. Restore your initial driver configuration by loading the DPDK driver:

   ```
   usertools/dpdk-devbind.py -b igb_uio <B:D.F>
   ```

Running the Utility

Use the following syntax to run the Intel Ethernet NVM Update Tool:

```
nvmupdate [options]
```

Using the `/?` option will display a list of supported command line options.

When running the utility, take note of the following:

- Progress and debug information are not provided by default.
  - Specify `-l <filename>` to send progress and debug information to a log file.
  - Specify `-l` without a filename to send the information to the console.
- More than one update may be required to bring your device's NVM fully up to date. In this situation, the tool will display and log a message indicating that another update is required.
- If both an NVM image and an FLB file are provided for the update, the OROM update will come from the FLB file.
- The Intel Ethernet NVM Update Tool supports using a PLDM package as the update image. If the PLDM package contains more than one NVM image, the tool will not apply the update and will exit with error code 23.
- Some Intel® Ethernet Converged Network Adapter X710-T4 adapters may display the message *"Image differences found at offset 0x7..."* when performing an update using the 700 Series NVM Update Package. This behavior is expected. Updating the Option ROM on a device with Device ID 1586 is a two-step process. The first update will change the Device ID to 1589. Reboot your system and run the update tool a second time to update the Option ROM for the new Device ID.
- If the tool indicates that a power cycle is required, make sure that your system powers off completely. A system reset or soft power cycle will result in an incomplete update. We recommend you disconnect the power cable for 30

seconds to ensure a complete power off.

- During an inventory operation, you may see link temporarily dropped until the Intel Ethernet NVM Update Tool completes its current operation. This is because the PHY controller is stopped to avoid issues when the tool accesses the PHY NVM interface.
- If recovery is requested but there are no devices in recovery mode, the Intel Ethernet NVM Update Tool will return success.
- On a system running VMWare ESXi 6.5 and later, while Secure Boot Mode is enabled, the Intel Ethernet NVM Update Tool cannot be run from datastores. If Secure Boot is enabled, you can run the Intel Ethernet NVM Update Tool from the */tmp* directory. Or you can disable Secure Boot, run the Intel Ethernet NVM Update Tool, and then re-enable Secure Boot.

Command Line Parameters

When run with no parameters, the Intel Ethernet NVM Update Tool runs in Interactive mode.

Note:

The `-i` and `-u` parameters are mutually exclusive; do not use both parameters at the same time.

-h, -?

Display command line usage help.

-a *<path>*

Specify a path for all file operations. The path is applied to all operations (such as locating the configuration file and NVM images, as well as where the log file and results file will be generated). Any path specified in the config file will be appended to the `-a` path. This parameter is used only with the `-i` or `-u` parameters.

-b

Save a backup copy of the current NVM image(s). The NVM Update tool creates a subdirectory using the NIC(s) MAC address as the directory name and stores the backup files there. Use this parameter only with the `-u` parameter.

-c *<file name>*

Specify the name of the configuration file. This is a text file that contains descriptions of networking devices and firmware versions for those devices. The syntax of the configuration file can be found in the Configuration File Syntax section below. If `-c` is not specified, nvmupdate.cfg is used. If you do not specify a config file, *nvmupdate.cfg* must be present in the same directory as the Intel Ethernet NVM Update Tool for Interactive mode to function correctly.

-f

Force the NVM update. The tool will skip binary verification and apply the update even if the image version is the same.

`-i`

Inventory mode. Lists the devices in the system and indicates the status of each device.

-if <interface>

Force the tool to use the specified driver interface. Supported values for `<interface>` are: `ioctl`.

`-l` ***<file name>***

Specify the name of the log file. This is a text file that contains a history of the Intel Ethernet NVM Update Tool's execution, including the success or failure status for each operation, and what adapters and OROMs were discovered. The log file is overwritten each time the Intel Ethernet NVM Update Tool is executed. If no file name is specified, the log messages are displayed on the system console.

-location <SS:BBB>

Specify a device for this instance of the tool to update, where `SS` is the PCI segment of the desired device and `BBB` is the PCI bus of the desired device. You cannot run multiple instances of the tool on a specific device.

> **Note:** `-location` must be used with the `-i` or `-u` parameter.

`-m` ***<MAC address>***

Update only the device with the specified MAC address. This is the LAN MAC address, not the SAN or AltMAC address.

> **Note:** The device must have an entry in the configuration file.

`-o` ***<file name>***

Specify the name and path of the results file. This is an XML file that contains the inventory/update results. If no file name is specified, the inventory/update results are displayed on the system console. Like the log file, the results file is overwritten each time the Intel Ethernet NVM Update Tool is executed.

-optinminsrev

In Update mode, `-optinminsrev` sets the minimum security revision of the NVM that is allowed to be installed on all devices defined in the configuration file, even if `MINSREV` is set to `FALSE` or is missing for a device. If `-optinminsrev` is not specified, only devices that have `MINSREV` set to `TRUE` in the configuration file will have their minimum security revision increased.

In Inventory mode, displays the device's SRev and MinSRev values in the results XML file.

> **Note:** `-optinminsrev` must be used with the `-c` parameter.

`-p`

Keep existing option ROM image (suppress update of the OROM).

`-r`

Rollback mode. Must be used with the `-m` parameter. You must have previously created a backup with the `-b` parameter for a restore point. The Intel Ethernet NVM Update Tool blocks rollback if the restore point NVM version is too old or does not contain critical updates.

`-rd`

Reset user settings to default values during update.

`-RecoveryDevices`

Must be used only with the `-i` or `-u` parameters. If you specify `-RecoveryDevices`, only devices in recovery mode will be displayed. If you specify `-u -RecoveryDevices`, only devices in recovery mode will be updated.

`-s`

Silent mode. Specifying this option will suppress all output to the display. The Intel Ethernet NVM Update Tool output is limited to the results file and the logfile.

`-sv`

Skip image verification.

`-u`

Update mode. Updates the devices in the system. You must provide a configuration file to use update mode.

`-UpdateDevices`

Must be used only with the `-i` or `-u` parameters. Ignores devices in recovery mode during update. If you specify `-i -UpdateDevices`, devices in recovery mode will not be displayed. If you specify `-u -UpdateDevices`, devices in recovery mode will not be updated.

`-v`

Display the version of the QV SDK and Ethernet driver installed in the system (base driver for Linux, FreeBSD, ESXi, and QV driver for Windows).

Command Line Examples

For these examples, enter Inventory mode and capture the status to a log file.

On an x64 Microsoft Windows system, type:

```
nvmupdatew64e -i -l nvmupdatelog.txt
```

On an x64 Linux system, type:

```
nvmupdate64e -i -l nvmupdatelog.txt
```

Example Inventory output:

```
Num Description                              Ver.(hex)    DevI
d  S:B     Status=== ==================================
===========  =====  ======  ================1)  Intel(R)
 Ethernet Connection X722                    37CC    00:004   R
ecovery Mode2)  Intel(R) Ethernet 10G 2P X550-t    1.32(
1.20)   1563   00:007  Update available    Adapter
```

For these examples, enter inventory mode and capture only devices in recovery mode to a log file.

On an x64 Microsoft Windows system, type:

```
nvmupdatew64e -i -RecoveryDevices nvmupdatelog.txt
```

On an x64 Linux system, type:

```
nvmupdate64e -i -RecoveryDevices nvmupdatelog.txt
```

Example Inventory output:

```
Num Description                              Ver.(hex)    DevId
  S:B     Status=== ================================== ==
=========  =====  ======  ================01) Intel(R) E
thernet Connection X722                      37CC    00:004   Reco
very Mode
```

For these examples, enter Inventory mode and capture only devices with updates available to a log file.

On an x64 Microsoft Windows system, type:

```
nvmupdatew64e -i -UpdateDevices nvmupdatelog.txt
```

On an x64 Linux system, type:

```
nvmupdate64e -i -UpdateDevices nvmupdatelog.txt
```

Example Inventory output:

```
Num Description                       Ver.(hex)   DevId
  S:B   Status=== ================================== ===
======== ===== ====== ================02) Intel(R) Ether
net 10G 2P X550-t    1.32(1.20)  1563  00:007 Update ava
ilable    Adapter
```

For these examples, enter interactive mode and display output and status to the screen.

On an x64 Microsoft Windows system, type:

```
nvmupdatew64e -l
```

On an x64 Linux system, type:

```
nvmupdate64e -l
```

For these examples, enter Update mode and specify a log file and configuration file.

On an x64 Microsoft Windows system, type:

```
nvmupdatew64e -u -l nvmupdatelog.txt -c nvmupdate.cfg
```

On an x64 Linux system, type:

```
nvmupdate64e -u -l nvmupdatelog.txt -c nvmupdate.cfg
```

Example Update output:

```
Num Description                       Ver.(hex)   DevId
  S:B   Status=== ================================== ===
======== ===== ====== ===================1)  Intel(R) Et
hernet Converged        6.01(6.01)  1521  00:001 Rollbac
k Blocked    Network Adapter XL710-Q22)  Intel(R) Ethern
et 10G 2P X550-t    1.32(1.20)  1563  00:007 Update succ
essful    Adapter3)  Intel(R) Ethernet Controller X540
4.04(4.04)  1560 00:132  Access error4)  Intel(R) Ethern
et Connection X722              1560 00:132  RECOVERY su
ccessful    for 10GbE SFP+
```

Exit Codes

Upon exit, when possible, the Intel Ethernet NVM Update Tool reports an overall status code to indicate the results of the operation. In general, a non-zero return code indicates an error occurred during processing.

Value	Description
0	All operations completed successfully.
1	General tool execution error.
2	The configuration file could not be opened/read, or a syntax error was discovered in the file.
3	The inventory process failed.
4	A file error occurred when accessing the results file.
5	Bad command line parameter.
6	An error occurred when updating a firmware module.
7	A file error occurred when creating/writing the log file.
8	An error occurred accessing the device.
12	The EEPROM MAP file could not be opened/read, or a syntax error was discovered in the file.
15	Another instance of the Intel Ethernet NVM Update Tool is already running.
18	An error occurred during reset.

Value	Description
19	Device not found.
20	Communication with base driver failed. Please verify that the base driver is present.
21	Unsupported NVM image discovered. Please upgrade to the latest version of the Intel Ethernet NVM Update Tool.
22	The image backup process failed.
23	The requested image cannot be applied over the existing NVM content. Please download the most recent update package and retry the update.
24	Cannot restore manufacturing data.
25	Update stopped due to Rollback Revision mismatch. The Intel Ethernet NVM Update Tool blocks rollback if the restore point NVM version is too old or does not contain critical updates.
26	The selected adapter cannot be updated due to strict MMIO memory settings in the kernel. Set the iomem kernel parameter to `relaxed` and reboot the system before running this utility again. Consult the utility documentation for more information.
30	MINSREV update failed. The minimum security revision was not set for the device.
31	Image prerequisite check failed.
35	The tool cannot find the correct preservation rules in the NVM on the device. User settings cannot be preserved with this update. To update the NVM, overwrite the user settings by using the `-rd` switch.

Value	Description
36	Parallel execution of Intel Ethernet NVM Update Tool for this device is not allowed. Either there is already an instance of Intel Ethernet NVM Update Tool running on this device or the device does not support the `-location` parameter.
37	Requested image cannot be applied over existing NVM content. Please download the most recent update package and retry update.
38	NVM update functionality is not supported on this device using this driver. Please download the latest driver from SourceForge.
39	NVM update functionality in recovery mode is not supported on this device using this driver. Please download the latest driver from SourceForge.
40	Individual updates are not supported on this device using the kernel driver. Please download the latest driver from SourceForge.
50	Please perform the indicated reset action and run the Intel Ethernet NVM Update Tool again to complete the update.
51	Update available for one or more adapters.

Note:

EFI versions of this tool may report an incorrect error code when no adapter is installed. This is due to a known limitation in the UDK2015 UEFI Development Kit (UDK) build environment.

Configuration File

The configuration file is a text file containing a short header followed by a series of *device blocks*. The device blocks contain a series of `tag:value` pairs, each of which describes an Intel networking device and specifies approved versions of firmware for that device.

The Intel Ethernet NVM Update Tool checks a device to see if it is a candidate for update. The check matches on the following fields:

- EtrackId (from the REPLACES field) + VENDOR + DEVICE + SUBVENDOR + SUBDEVICE
- EtrackId (from the REPLACES field) + VENDOR + DEVICE
- VENDOR + DEVICE + SUBVENDOR + SUBDEVICE
- EtrackId (from the REPLACES field) + VENDOR + DEVICE + SUBVENDOR + SUBDEVICE + PBA
- EtrackId (from the REPLACES field) + VENDOR + DEVICE + PBA
- VENDOR + DEVICE + SUBVENDOR + SUBDEVICE + PBA

Restrictions and Rules

- You cannot update the EEPROM and the NVM in the same session. Only one should be specified in the configuration file.
- You can update the EEPROM and the OROM in the same session. The EEPROM will be updated first, followed by the OROM.
- You can update the NVM and the OROM in the same session. The NVM will be updated first, followed by the OROM.
- Inventory mode does not consider potential EEPROM/NVM image changes if a sequential OROM update is requested (e.g. an additional reboot might be required).
- On multiport adapters, due to there being only one EEPROM/flash that is shared by all ports, only port 0 will be updated (or the port with the MAC address specified with the -m command line parameter).
- Blank lines or lines beginning with a semicolon (comments) are ignored.
- Unrecognized keywords in the configuration file will result in the NVM Update tool exiting abnormally and reporting a syntax error (exit code=2). Run the NVM Update tool again, with the Enable Logging command line parameter (`-l`), to see what specific line is in error.
- An OROM update is allowed only if the specified OROM component set matches combo rules (the vendordevice pairs defined in the *.flb* file).
- An OROM downgrade is allowed only if a value of `TRUE` is assigned to the `OROM DOWNGRADE` key.
- If the configuration file specifies the OROM IMAGE field, but does not specify any OROM components (PXE, EFI, etc.), the device's current set of OROM components is used as the component list.
- An OROM update preserves the current configuration of components, or resets the configuration to the default if new settings do not allow for configuration preservation (e.g., when combo rules differ).
- EtrackId of EEPROM or NVM and OROM components version values specified in the configuration file are compared against values stored in .eep/.bin or .flb files. If a mismatch is found, the inventory/update is aborted.
- If the EtrackId of EEPROM or NVM and OROM components version values are not specified in the configuration file, then values from .eep/.bin or .flb are taken without verification.

- EEP/NVM/OROM update is always allowed if running image versions cannot be specified or are invalid.
- EEPROM/NVM downgrade is allowed only when the REPLACES key is specified with an active EtrackID value.
- A candidate match based on sub IDs requires that both SUBVENDOR and SUBDEVICE fields are present in the configuration file.
- If the REPLACES field is not present, you must include the SUBVENDOR and SUBDEVICE fields in the configuration file. SUBVENDOR and SUBDEVICE are optional if REPLACES is present.
- The Intel Ethernet NVM Update Tool will not report errors for devices that are not specified in the config file.
- If recovery is requested but there are no devices in recovery mode, the NVM Update tool will return success.

Configuration File Syntax

All configuration file fields are required, except for those indicated as "[Optional]".

See Configuration File Examples below for examples of this syntax in practice.

CURRENT FAMILY: <version>

[Optional] Defines OEM version (x.y.z) for the file.

CONFIG VERSION: <version>

Defines the version of the configuration file format in use. The parameter is in the format `<x.y.z>`.

BEGIN DEVICE

Demarcates the start of a device definition block.

DEVICENAME: xxxx

[Optional] The branding string of the device. This is only used to make the file easier to read; it is not processed by the tool.

VENDOR: xxxx

The vendor ID, typically 8086, in hexadecimal.

DEVICE: xxxx

The device ID number, in hexadecimal. You can specify more than one device ID, separated by spaces.

SUBVENDOR: xxxx

The subvendor ID number, in hexadecimal. You can specify more than one subvendor ID, separated by spaces.

SUBDEVICE: xxxx

The subdevice ID number, in hexadecimal. You can specify more than one subdevice ID, separated by spaces.

REVISION: xxxx

[Optional] The revision number, in hexadecimal. If not provided, revision numbers of all devices in the system will be considered to match configuration file definitions.

CURRENT GFID: <GFID number>

[Optional] If specified, this value must match the `CURRENT GFID` value on the update image for the update to proceed. If `CURRENT GFID` is not specified, but `ORIGINAL GFID` is, then nvmupdate will use the `ORIGINAL GFID` value in both cases. If neither is specified, then the `<GFID number>` value will be taken directly from the image in the *.flb* file and compared to the device. Specifying a `CURRENT GFID` value reduces the number of *.flb* file reads, which reduces the time required for the update to complete.

ORIGINAL GFID: <GFID number>

[Optional] If specified, this value must match the `ORIGINAL GFID` value on both the pre-update device and the update image for the update to proceed. If `ORIGINAL GFID` is not specified, but `CURRENT GFID` is, then nvmupdate will use the `CURRENT GFID` value in both cases. If neither is specified, then the `<GFID number>` value will be taken directly from the image in the *.flb* file and compared to the device. Specifying an `ORIGINAL GFID` value reduces the number of *.flb* file reads, which reduces the time required for the update to complete.

GFID EXACT: <TRUE/FALSE>

[Optional] If specified, only devices that exactly match the GFID (origin and current) will be updated.

MINSREV: <TRUE/FALSE>

[Optional] In Update mode, sets the minimum security revision of the NVM that is allowed to be installed on the device. If set to `TRUE`, sets `MINSREV` to the SREV value of the firmware currently installed on the device (This is set after the update, if an update is specified in the configuration file). If set to `FALSE`, or not included, the `MINSREV` value is not changed.

In Inventory mode, displays the device's SRev and MinSRev values in the results XML file.

PBA: <pba_number>

[Optional] The PBA number, in either the first six characters of the PBA (123456) or the full PBA (123456-000) form. If PBA is not provided, then all devices in the system are considered to match configuration file definitions.

PXE: <version>

[Optional] The version number (x.xx.xx), in decimal format, for the PXE firmware subcomponent. The string `NONE` (or value 0.0.0) indicates that PXE should not be installed on this device. If `<version>` is not specified, then the value is taken directly from the *.flb* file.

EFI: <version>

[Optional] The version number (x.xx.xx), in decimal, for the EFI boot firmware subcomponent. The string `NONE` (or value 0.0.0) indicates that EFI should not be installed on this device. If `<version>` is not specified, then the value is taken directly from the *.flb* file.

EFI_IA64: <version>

[Optional] The version number (x.xx.xx), in decimal, for the EFI IA64 boot firmware subcomponent. The string `NONE` (or value 0.0.0) indicates that `EFI_IA64` should not be installed on this device. If `<version>` is not specified, then the value is taken directly from the *.flb* file.

ISCSI: <version>

[Optional] The version number (x.xx.xx), in decimal, for the iSCSI boot firmware subcomponent. The string `NONE` (or value 0.0.0) indicates that iSCSI boot should not be installed on this device. If `<version>` is not specified, then the value is taken directly from the *.flb* file.

SMCLP: <version>

[Optional] The version number (x.xx.xx), in decimal, for the SMCLP firmware subcomponent. The string `NONE` (or value 0.0.0) indicates that SMCLP should not be installed on this device. If `<version>` is not specified, then the value is taken directly from the *.flb* file.

OROM IMAGE: <filename>

The name of the file that contains OROM components for this device (typically an *.flb* file).

OROM DOWNGRADE: <TRUE/FALSE>

[Optional] If set to `TRUE`, allows the option ROM image to be downgraded to a previous version. `OROM DOWNGRADE` permission is set to `FALSE` by default.

SKIP OROM: <TRUE/FALSE>

[Optional] If set to `TRUE`, keeps the existing option ROM image (suppresses update of the OROM).

EEP IMAGE: <filename>

The name of the file containing the EEPROM image for this device (typically an *.eep* file).

EEPROM MAP: <filename>

[Optional] The name of the text file that contains a list of words to be preserved, overwritten, or excluded from verification by the NVM Update tool. This key is to be part of each device configuration specified within a device block in the configuration file. `EEPROM MAP` is optional and has no default value. If no value is specified in the key, a configuration file error (exit code 2) is returned. Details regarding the format and content of this text file are given in the EEPROM Map Text File Syntax section below.

BEGIN IDEEPROM: ID=<ideeprom_id>

Demarcates the beginning of an ID EEPROM subsection. `ID=<ideeprom_id>` is optional and is the OCP_ID of an ID EEPROM image to use for the update in hexadecimal.

An ID EEPROM definition block may contain the following fields:

```
IMAGE: <filename>    The file that contains the ID E
EPROM image.  DOWNGRADE: <TRUE|FALSE>    [Optional]
If set to TRUE, allows the ID EEPROM image to    be
downgraded to a lower version. IDEEPROM DOWNGRAGE
 permission is set to FALSE by default.  VERSION: <v
ersion>    [Optional] the version number of the ID E
EPROM image.
```

END IDEEPROM

Demarcates the end of the ID EEPROM subsection.

NVM IMAGE: <filename>

[Optional] The name of the file that contains the entire NVM image (EEPROM, OROM and FW) for this device (typically a *.bin* file).

IMAGE DOWNGRADE: <TRUE/FALSE>

[Optional] If set to `TRUE`, allows the EEPROM or NVM image to be downgraded to a previous version. `IMAGE DOWNGRADE` permission is set to `FALSE` by default.

NETLIST IMAGE: <filename>

The name of the file that contains the Netlist module.

NETLIST VERSION: <version>

[Optional] Only used with `NETLIST IMAGE`. The `<version>` number is `<BRVmajor.BRVminor.BRVrevision.CV>`, in decimal, for the Netlist module, where:

BRVmajor:

Base release version, Major

BRVminor:

Base release version, Minor

BRVrevision:

Base release version, Revision number

CV:

Customer version number

NETLIST TYPE: <type>

Type of the associated Netlist module, in decimal.

NETLIST DOWNGRADE: <TRUE/FALSE>

[Optional] If set to `TRUE`, allows the Netlist module to be downgraded to a previous version. `NETLIST DOWNGRADE` permission is set to `FALSE` by default.

SKIP NETLIST: <TRUE/FALSE>

[Optional] If set to `TRUE`, keeps the existing Netlist module (suppresses update of the Netlist module).

EEPID: xxxx

[Optional] The unique 32-bit ETrack-ID image tag, in hexadecimal.

REPLACES: xxxx

[Optional] EEPID of the old image that the current image replaces. You can specify multiple old images by separating them with a space. Used to handle customized NVM settings that may have the same vendor and device ID information as other devices (for driver compatibility reasons) but need specific update files. If not present, then the section will wildcard match on the listed `VENDOR`, `DEVICE`, and `SUBSYSTEM ID` fields.

MODE: <device_mode>

[Optional] List of modes which are supported by the device. If `MODE` is not specified, devices that have a valid update or are in recovery mode will all be updated. Valid values for `<device_mode>` are:

UpdateDevices:

Ignore devices in recovery mode.

RecoveryDevices:

Only update devices in recovery mode.

RESET TYPE: <reset_type>

[Optional] The type of reset required after updating the EEPROM. Allows the developer of the EEPROM package to specify the correct reset type. `RESET TYPE` is ignored if the device's EEPROM is not updated (i.e., Option ROM update only). Valid values for `<reset_type>` are:

NONE:

No reset required.

PHY:

PHY reset for 10G and 40G adapters. This reset issue executed only if the PHY NVM or PHY module was updated.

HICR:

Firmware reset (initiated by software) that is only usable on HICR enabled devices. This reset is executed only if the EEPROM was updated (1G and 10G adapters) or if the MAC NVM was updated (10G adapters).

BMC:

Firmware reset initiated by the BMC via SMBus or NC-SI. The tool tells the user to reset the device via BMC or a power cycle of the system.

DELAYED_REBOOT:

The device will not load and run the update until your system is rebooted. Once the system is rebooted, the update will load on the device.

REBOOT:

The tool tells the user to reboot the system (only if the NVM/EEPROM was updated). This reset type is supported on all devices.

POWER:

The tool tells the user to power cycle the system (only if the NVM/EEPROM was updated). This reset type is supported on all devices.

You can specify more than one `RESET TYPE` per device. If more than one type is specified, and the conditions are met for more than one type, only the reset with the highest priority will be executed:

- `POWER` has the highest priority.
- `REBOOT` has the second highest priority.
- `DELAYED_REBOOT`, `BMC`, `HICR`, and `PHY` all share the lowest priority.

PHY DOWNGRADE: <TRUE/FALSE>

[Optional] If set to `TRUE`, allows the PHY NVM image to be downgraded to a lower version. `PHY DOWNGRADE` permission is set to `FALSE` by default.

SKIP PHY: <TRUE/FALSE>

[Optional] If set to `TRUE`, keeps the existing PHY NVM image (suppresses update of the PHY NVM).

PHY IMAGE: <filename> <phy_family>

The name of the file that contains the PHY FW image for this device followed by the PHY family ID. The PHY family ID is compared to the device's NVM. The PHY FW will only update if the values match.

PHY VERSION: <version> <filename>

[Optional] The version number of the PHY NVM contained in <image file>.

BEGIN PHY: ID=<phy_id> , FAMILY=<phy_family>

Demarcates the beginning of a PHY definition block, where:

ID=<phy_id>:

The PHY ID in hexadecimal.

Family=<phy_family>:

The PHY Family ID in hexadecimal.

The PHY ID and PHY Family ID are compared to the device’s NVM. The PHY FW will only update if the values match.

A PHY definition block may contain the following fields:

DOWNGRADE: <TRUE/FALSE>

[Optional] If set to `TRUE`, allows the PHY NVM image to be downgraded to a lower version. `PHY DOWNGRADE` permission is set to `FALSE` by default.

IMAGE: <filename>

The name of the file that contains the PHY FW image for this device.

VERSION: <version>

[Optional] The version number of the PHY NVM.

END PHY

Demarcates the end of a PHY definition block.

PAL IMAGE: <image file>

File that contains the platform adaption layer image for this device.

PAL VERSION: <version>

[Optional] The version number of the platform adaption layer image.

PAL DOWNGRADE: <TRUE/FALSE>

If set to `TRUE`, allows the platform adaption layer image to be downgraded to a previous version. `PAL DOWNGRADE` permission is set to `FALSE` by default.

SKIP PAL: <TRUE/FALSE>

[Optional] If set to `TRUE`, keeps the existing platform adaption layer image (suppresses update of the platform adaption layer).

RO IMAGE: <filename>

The name of the file that contain RO module image for the device.

RO VERSION: <version>

The version number for the RO module, in hexadecimal format.

FEATURES: <feature>

[Optional] Device features which cannot be detected by the Intel Ethernet NVM Update Tool. Valid values for `<feature>` are:

DOUBLE_REBOOT_GPIO{n}:

The device circuit handling a double reboot using GPIO{n} (where `{n}` is an integer value from 0 to 5).

FORCE UPDATE: <TRUE/FALSE>

[Optional] If set to `TRUE`, updates the device regardless of the Pending Reboot state. If excluded or set to `FALSE`, will only update the device if it is not in Pending Reboot.

PRESERVATION: <preservation_type>

How much of the Preserved Field Area (PFA) is kept during the update.

Valid values for `<preservation_type>` are:

FULL:

No change to current PFA

SELECTIVE:

Preserves selected fields in current PFA

MANUFACTURING DEFAULTS:

Replace PFA with values from manufacturing defaults module

NONE:

No preservation (firmware overwrites current PFA with the default PFA from the image)

END DEVICE

Demarcates the end of a device definition block.

Configuration File Examples

The following is an example of a configuration file with one device block:

```
CURRENT FAMILY: 12.1.1CONFIG VERSION: 1.7.0BEGIN DEVICE
 DEVICENAME: Intel x540 Adapter VENDOR: 8086 DEVICE: 10C
9 SUBVENDOR: 8086 SUBDEVICE: A04C REVISION: 2 PXE: 1.3.0
4 EFI: 2.7.06 ISCSI: 3.8.17 SMCLP: 5.8.17 NVM IMAGE: nvm
Image.bin OROM IMAGE: BootImg.flb EEPID: 800007A9 REPLAC
ES: 80000692 EEPLAYOUT: 82599 RESET TYPE: POWEREND DEVIC
E
```

The following is an example of a device block with multiple device IDs:

```
BEGIN DEVICE VENDOR: 8086 DEVICE: 37D1 37D2 37D3 SUBVEND
OR: 8086 SUBDEVICE: 0 NVM IMAGE: nvm_image.binEND DEVICE
```

EEPROM Map Text File Syntax

See EEPROM MAP File Example below for examples of this syntax in practice.

EEPROM MAP FILE VERSION: 1.1.0

This version number will be used to find any mismatch between the NVM Update tool and EEPROM MAP file version. This is the only obligatory field in the file.

Data in the file are divided into 3 major blocks:

- PRESERVE
- VERIFICATION EXCLUDED
- OVERWRITE

Data from PRESERVE and VERIFICATION EXCLUDED blocks are excluded from image comparison. Image comparison occurs when the same version of the image is found on the NIC and in the files provided for the inventory/update, and after an update to verify if written image matches provided one. The comparison recognizes if the current image is corrupted. If the same set of data is provided in the PRESERVE and OVERWRITE blocks, data from OVERWRITE takes precedence. All PRESERVE actions occur before OVERWRITE actions, regardless of where they are in the map file.

There is no conflict if the same data are provided in VERIFICATION EXCLUDED and either PRESERVE or OVERWRITE blocks.

PRESERVE and VERIFICATION EXCLUDED blocks

Within the PRESERVE and VERIFICATION EXCLUDED blocks, multiple sections of the same type definition may be included. The following blocks are analyzed:

DIRECT - <offset> <bitmask> <count>

Section containing words' addresses. Data in this section (separated by white spaces) consist of:

<offset>:

word offset, e.g. "0x2E"

<bitmask>:

word bitmask, e.g. "0xFFFF"

<count>:

consecutive number of words, e.g. "1"

POINTER - <pointer> <offset> <bitmask> <count>

Section containing pointers and offsets. Data in this section (separated by white spaces) consist of:

<pointer>:

pointer address to selected area, e.g. "0x07"

<offset>:

word offset, e.g. "0x01"

<bitmask>:

word bitmask, e.g. "0xFFFF"

<count>:

consecutive number of words, e.g. "3"

DOUBLE POINTER - <pointer> <pointer offset> <offset> <bitmask> <count>

Section containing pointers, second level pointer offsets and third level offsets. Data in this section (separated by white spaces) consist of:

<pointer>:

pointer address to selected area (module pointer), e.g., "0x07"

<pointer offset>:

pointer offset (value calculated as a relative to module pointer), e.g. "0x02"

<offset>:

word offset, e.g. "0x0A"

<bitmask>:

word bitmask, e.g. "0xFFFF"

<count>:

consecutive number of words, e.g. "1"

TLV - <TLV type>

Section containing TLVs to be preserved, where:

<TLV type>:

TLV type (hex), e.g. "1A2B"

VPD

Section may contain one major VPD subsection (this subsection is optional and may be omitted):

ALL:

If set, preserves the VPD content during a recovery operation. This value is only for use if the device is in recovery mode. It will not affect normal operations.

OVERWRITE block

If values (and colons) are not provided in the below sections of the OVERWRITE block, the value from the NVM image is used for update. This is the desired way to exclude selected VPD areas from being preserved.

The EEPROM MAP file may contain comments after a ";" (semicolon). Each block and section needs to have BEGIN and END statements. If the file referenced in the configuration file is not provided, or the data in the file is of the wrong format, the Intel Ethernet NVM Update Tool will exit with exit code `EEPROM MAP FILE ERROR` (exit code: 12). If the file contains only the file version with no other definitions, the file is skipped.

The OVERWRITE block should contain blocks and sections with values:

DIRECT - <offset><value>

Section containing words' addresses and their values. Data in this section consist of:

<offset>:

word offset, e.g. "0x2E"

colon (:):

(optional)

<value>:

value (hex), e.g. "1A2B" (optional)

POINTER - <pointer> <offset><value>

Section containing pointers, offsets, and their values. Data in this section (separated by white spaces) consist of:

<pointer>:

pointer address to selected area, e.g. "0x07"

<offset>:

word offset, e.g. "0x00"

colon (:):

(optional)

<value>:

value (hex), e.g. "1A2B" (optional)

DOUBLE POINTER - <pointer> <pointer offset> <offset><value>

Section containing pointers, second level pointer offsets and third level offsets. Data in this section (separated by white spaces) consist of:

<pointer>:

pointer address to selected area (module pointer), e.g. "0x07"

<pointer offset>:

pointer offset (value calculated as a relative to module pointer), e.g. "0x02"

<offset>:

offset, e.g. "0x0A"

colon (:):

(optional)

<value>:

value (hex), e.g. "1A2B" (optional)

TLV - <TLV type><TLV_offset>: <value>

Section containing TLVs to be overwritten with new values.

<TLV type>:

TLV type (hex), e.g. "1A2B"

<TLV_offset>:

offset to first word in buffer specified for TAG which should be changed.

colon (:):

(optional)

<value>:

value (hex), e.g. "1A2B" (optional)

VPD - <key><value>

Section may contain three major VPD subsections (these subsections are optional and may be omitted):

READ:

WRITE:

ALL:

If set, does not preserve the VPD content. VPD content will be overwritten.

Data in the READ and WRITE (sub)sections consist of:

<key>:

one of valid VPD fields defined in PCI VPD specification (e.g. PN, EC, SN, Vx)

colon (:):

(optional)

<value>:

value (ASCII), e.g. "FFV13.00.00" (optional)

EEPROM MAP File Example

Example EEPROM MAP file contents with data to be preserved, overwritten, and excluded from comparison:

```
EEPROM MAP FILE VERSION: 1.0.0BEGIN PRESERVE    BEGIN DI
RECT        0x0B 0xFFFF 1    ;Subsystem ID    END DIRECT
    BEGIN POINTER        0x37 0x0 0xFFFF 6    ;Alternate
 MAC address    END POINTER    BEGIN DOUBLE POINTER
   0x0C 0x11 0x00 0xFFFF 1    END DOUBLE POINTEREND PRES
ERVEBEGIN VERIFICATION EXCLUDED    BEGIN DIRECT        0
x38 0x0003 1    ;APM Enable Port    END DIRECT    BEGIN
POINTER    0x28 0x0 0xFFFF 6    END POINTEREND VERIFICAT
ION EXCLUDEDBEGIN OVERWRITE    BEGIN DIRECT        0x00
: 001B    ;MAC address        0x01 : 1234        0x02 :
5678    END DIRECT    BEGIN POINTER        0x29 0x01 : 1
111    END POINTER        BEGIN DOUBLE POINTER        0x
0C 0x11 0x00 : ABCD    END DOUBLE POINTER    BEGIN VPD
      ALL    ;reset the VPD content        BEGIN READ
         V1        END READ        BEGIN WRITE
  V3 : DTINIC        END WRITE    END VPDEND OVERWRITE
```

Troubleshooting

Note:

If the tool displays an error such as *"Access error"* or *"Cannot initialize port"* you may be using an outdated driver. Please download the latest driver from https://support.intel.com and try again.

Note:

If the tool displays the error: *"Unable to load the driver. Please close all other applications and try again"*, you have a mix of old and new versions of the utility tool on your system. Quit all open applications and retry your operation. If the issue persists:

1. Download the latest version of the utility tools.
2. Run the uninstall script to remove the old version of the tool driver.
3. Run the install script from the downloaded tools package.
4. Retry your operation.

You may also need to download and install the latest Intel Ethernet driver or Intel PROSet package for your device.

NVM Update Reference Application (NVM URA) for Intel® Ethernet

Contents

Overview

The NVM Update Reference Application (NVM URA) is a reference application to help you create a customized NVM update tool.

- When you run the NVM URA tool, it generates the *nvm_ura.log* log file. This file is overwritten each time you run the tool.
- You can only update one adapter at the time.
- If you run NVM URA without specifying any options, it will display the tool version and the list of supported adapters.

System Requirements

- Your system must be running 64-bit Linux* or 64-bit FreeBSD*.
- The base networking driver must be installed on the system.
- You must run NVM URA with root privileges.

Building and Installation

Building NVM URA on Linux

NVM URA is built by Makefile. Run the following commands:

```
cd <location-of-nvm-ura>make
```

Building NVM URA on FreeBSD

NVM URA is built by Makefile. Run the following commands:

```
cd <location-of-nvm-ura>gmake
```

Options

-h

Displays command line help.

-a

Display the adapter PHY version.

-b

Saves a backup copy of the NVM memory content of the selected network adapter to a file.

-c *<file>*

Programs the NVM from the file.

-d

Saves a backup copy of the ID EEPROM content of the selected network adapter to the file *<mac_address>_eeid.bin*.

-e

Programs the PHY NVM. If this option is used along with the `-c` option, PHY NVM will be programmed after the NVM update procedure.

-f

Triggers a firmware reset after the NVM update procedure. Must be used with the `-c` option.

> **Important:**
>
> This will cause the device to stop responding until the firmware reset completes.

-g

Program the ID EEPROM of the selected network adapter from a file.

-i

Displays adapter identification and versioning information.

-p *<preserve>*

Selects the preservation mode. Must be used with the `-c` option. Valid values for `<preserve>` are: `full`, `selective`, `factory`, `none`. If this option is not used, programs the NVM with full preservation.

-r *<file>*

Verifies the NVM against a file.

`-s <location>`

Selects the adapter by PCI location in the format, `[bb:dd]`, where:

b:

bus

d:

device

Both numbers must be in hexadecimal.

NVM Update Package

Contents

Overview

An NVM Update Package contains all the required files to update the NVM on the Intel® Ethernet adapters in your system. It contains the Intel® Ethernet NVM Update Tool, configuration file, updated NVM binaries, and required driver files.

Note:

Some Intel® Ethernet Converged Network Adapter X710-T4 adapters may display the message, “Image differences found at offset 0x7…,” when performing an update using the 700 Series NVM Update Package. This behavior is expected.

Updating the Option ROM on a device with Device ID 1586 is a two step process. The first update will change the Device ID to 1589. Reboot your system and run the update tool a second time to update the Option ROM for the new Device ID.

Adapters Based on the Intel® Ethernet 800 Series

NVM binaries in the NVM Update Package are in the Platform Level Data Model (PLDM) format defined by the DMTF in DSP0240. Refer to your vendor's platform documentation for instructions on how to update your device NVM using the baseboard management controller (BMC) or UEFI.

Limitations and Prerequisites

This package is intended to be used on Intel branded adapters. Please contact your OEM vendor for an appropriate package. In some cases this package may update an OEM device. This package only updates the NVM image for the device family listed on the package. Each Intel Ethernet product family has its own NVM Update Package.

Important: DO NOT:

- Power down your system during the NVM Update.
- Remove the NIC before the NVM Update completes.
- Interrupt the NVM Update in any other way.

Taking any of these actions could make your device unusable.

Link Loss During and After NVM Update

When you update a device based on the Intel® Ethernet Controller X710 and Intel® Ethernet Controller XL710, the device may lose the link during and after the update. Power cycle your system after the NVM update completes to resolve the issue.

Linux, FreeBSD, and ESX Requirements

The base driver for your NIC must already be installed.

UEFI Requirements

Create a bootable disk or other media and extract the update package onto the media. Boot your system from the media and run the tool from there.

Firmware Recovery Mode

When a device is in Firmware Recovery mode, it will not pass traffic or allow any configuration; you can only attempt to recover the device's firmware. A device will enter Firmware Recovery mode if it detects a problem that requires the firmware to be reprogrammed.

Note:

Before starting the recovery process, make sure that your operating system, drivers, and tools have been installed properly. You must use the out-of-tree driver. Using the in-box or kernel driver may result in a *Cannot initialize port* warning.

Note:

You must power cycle your system after using Recovery Mode to completely reset the firmware and hardware.

Updating Your NVM Using Interactive Mode

1. Extract the update package into a temporary folder.
2. Start the Intel Ethernet NVM Update Tool by running the executable. For example, on an x64 Microsoft* Windows* system, type:

   ```
   nvmupdatew64e
   ```

   On an x64 Linux* system, type:

   ```
   nvmupdate64e
   ```

3. Follow the prompts to update the NVM image on the desired device.

Using a Script

You can use a script to perform an inventory of all the Intel Ethernet devices in the system or update the Intel Ethernet devices in the system.

Update script example:

```
nvmupdate64e -u -l -o results.xml -b -c nvmupdate.cfg
```

This script causes the Intel Ethernet NVM Update Tool to check the installed Intel Ethernet devices against those contained in *nvmupdate.cfg*. If a device contains an NVM version older than that specified in the config file, the utility will update the device’s NVM. It will create an XML file containing the results of the update.

Note:

Using `-b`, which is optional, will create a backup of the current NVM image(s). This may add about 30% to the tool’s execution time.

Inventory script example:

```
nvmupdate64e -i -l -o inventory.xml
```

This script causes the Intel Ethernet NVM Update Tool to perform an inventory of all the Intel Ethernet devices in the system and creates an output file (called *inventory.xml*) of the results of the inventory.

Results

The Intel Ethernet NVM Update Tool will return an exit code of zero if the update completed successfully.

About Intel PROSet®

Intel® PROSet is a suite of software tools to configure Intel Ethernet devices on Microsoft Windows operating systems. Intel PROSet software includes the following:

App Name	Purpose
Intel® PROSet for Windows* Device Manager	An extension to the Windows Device Manager. When you install Intel PROSet for Windows Device Manager, additional tabs are automatically added to Device Manager.
Intel® PROSet Adapter Configuration Utility (Intel® PROSet ACU)	A graphical user interface that allows you to configure and manage supported Intel Ethernet adapters.
Intel® PROSet for Windows PowerShell* software	Software that contains PowerShell cmdlets that allow you to configure and manage the Intel Ethernet adapters and devices present in your system.

For more information on installing or using Intel PROSet, refer to the following:

- About Intel® Ethernet Cmdlets
- Compatibility Notes for Intel Ethernet Cmdlets and Intel PROSet
- Installing Intel Ethernet Cmdlets
- Installing Intel PROSet
- Configuring Features with Windows PowerShell
- Configuring Features with Intel PROSet Adapter Configuration Utility
- Configuring Features with Intel PROSet for Windows Device Manager
- Changing Intel Ethernet Settings Under Windows Server Core
- Diagnostics in Intel PROSet

- Diagnostics in Ethernet Cmdlets

Note:

- You must have administrator rights to install or use Intel PROSet.
- Intel PROSet requires the latest driver and software package for your Intel Ethernet devices.

About Intel® Ethernet Cmdlets

In addition to Intel PROSet, Intel provides Ethernet cmdlets for Windows PowerShell to:

- Display information about Intel Ethernet devices in the system
- Configure device settings
- Configure and gather firmware logs for debugging supported devices
- Blink the LED on the specified port of an Intel Ethernet device for a defined number of seconds
- Run hardware diagnostics

These Ethernet cmdlets do not require Intel PROSet to be installed on the system.

For more information on installing or using Intel Ethernet cmdlets, refer to the following:

- Compatibility Notes for Intel Ethernet Cmdlets and Intel PROSet
- Installing Intel Ethernet Cmdlets
- Configuring Features with Windows PowerShell
- Diagnostics in Ethernet Cmdlets
- Firmware Logging

Compatibility Notes for Intel Ethernet Cmdlets and Intel PROSet

Note the following compatibility requirements for Intel Ethernet cmdlets and Intel PROSet.

For Intel PROSet, the installer will automatically detect and install the components compatible with your operating system.

Device Compatibility

The following devices either do not support Intel Ethernet cmdlets, Intel PROSet, or some of its components.

Software	Limitations
Ethernet cmdlets	• No support: ◦ Intel Ethernet Connection E822 ◦ Intel Ethernet 500 Series and lower • Support varies: ◦ Any platform with a System on a Chip (SoC) processor
Intel PROSet	• No support: ◦ Intel Ethernet Connection E822 • PowerShell only: ◦ Any platform with a System on a Chip (SoC) processor

Note:

- Intel Ethernet Connection E822 does not support Intel PROSet or Ethernet cmdlets because it does not support Microsoft operating systems.
- If designated above as "PowerShell only," this means the device supports only Intel PROSet for Windows PowerShell software. It does not support Intel PROSet ACU or Intel PROSet for Windows Device Manager.
- For platforms with an SoC processor, this includes a platform with either a server controller (designated by an initial E or X, such as E823 or X722) or both a server and client controller (designated by an initial I, such as I210).

Operating System Compatibility

Intel PROSet or its components, or Intel Ethernet cmdlets, are compatible with the following operating systems.

Note:

Not all OS versions listed below are supported in the newest releases of Intel Ethernet software. Refer to Supported Operating Systems for currently supported versions.

- Intel Ethernet cmdlets:
  - Microsoft Windows Server 2016 and later

    - Microsoft Azure Stack HCI
    - Microsoft Windows PowerShell version 5.1 and later
- Intel PROSet for Windows PowerShell software:
    - Any supported version of Microsoft Windows Server
    - Any supported version of Microsoft Windows
    - Microsoft Azure Stack HCI
- Intel PROSet Adapter Configuration Utility:
    - Microsoft Windows Server 2019 and later
    - Microsoft Windows 10 Version 1809, and later versions of Microsoft Windows 10
- Intel PROSet for Windows Device Manager:
    - Microsoft Windows Server 2016, and earlier versions
    - Microsoft Windows 10 Version 1709, and earlier versions

> **Note:**
>
> Intel PROSet is not supported on Microsoft Windows 11. Microsoft Windows 10 is the last Windows operating system that supports Intel PROSet.

Installing Intel Ethernet Cmdlets

> **Note:**
>
> We strongly recommend you install the latest base driver and NVM before installing Intel Ethernet cmdlets, or functionality could be reduced.

To install Intel Ethernet cmdlets:

1. Download the zip file with the adapter cmdlets from the Intel Download Center.
2. Save the zip file to the system you're trying to configure or debug.
3. Extract files from the zip file.
4. In the extracted files, double-click on the .exe file. The cmdlets will automatically install to the current user's PowerShell module directory (for example, to `C:\Users\Administrator\Documents\WindowsPowerShell\Modules\IntelEthernetCmdlets`).

After installation, you can run the cmdlets at the PowerShell prompt without manually importing the module.

Refer to the following for more information:

- The readme.txt file included in the download

- The cmdlet help in PowerShell
- In this user guide:
  - Configuring Features with Windows PowerShell
  - About Intel® Ethernet Cmdlets
  - Firmware Logging

Installing Intel PROSet

To download and install only Intel PROSet:

1. After the driver installation has finished, download and extract the **Wired_PROSet_XX.X_*.zip** file for your version of Windows, where `XX.X` is the release number. This file will install Intel PROSet on your system.
   - If you are running a 32-bit operating system, download **Wired_PROSet_XX.X_32.zip**.
   - If you are running a 64-bit operating system, download **Wired_PROSet_XX.X_x64.zip**.
2. In the extracted files, double-click on the **.exe** file to launch the installation of Intel PROSet.
3. In the install wizard that opens, follow the onscreen prompts to select the features you want to install.
4. The Intel PROSet software will install. Click **Finish** when prompted.

Note:

- Intel PROSet is not installed by default when you install only the device drivers. You must install Windows device drivers before installing Intel PROSet. See Install Windows Drivers for more information.
- You must have administrator rights to install or use Intel PROSet.
- You must install Intel PROSet if you want to use Intel® Advanced Network Services (Intel® ANS) or VLANs. Intel ANS teams and VLANs are not supported on all operating systems or devices. Refer to Adapter Teaming and Virtual LANs (VLANs) for more information.
- Intel PROSet requires the latest driver and software package for your Intel Ethernet devices.

If Intel PROSet was installed without Intel ANS support, you can install support by running *DxSetup.exe*, and then selecting the **Modify** option when prompted. From the Setup Options window, select **Intel® Advanced Network Services** and then click Next to continue with the installation wizard.

Command Line Installation for Intel PROSet

You can also run *DxSetup.exe* from the files downloaded from the software release. *DxSetup.exe* is an install utility that allows unattended installation of Intel PROSet from a command line.

> **Note:**
>
> - Intel PROSet cannot be installed with *msiexec.exe*. You must use *DxSetup.exe*.
> - If the operating system and installed adapters do not support a given feature, that feature will not be installed even if the command line option specifies it for installation.

By setting the parameters in the command line, you can enable and disable management applications. If software is already installed and parameters are not specified, only existing components are updated.

You can also use *DxSetup.exe* to modify or upgrade software:

- If a feature is already installed, the public property for that feature will default to 1.
- If a feature is not installed, the public property for that feature will default to 0.

Running *DxSetup.exe* without specifying properties will upgrade all installed software. You can remove installed software (except for base drivers) by setting the property to 0. If you uninstall Intel PROSet (`PROSET=0`), all features that rely on Intel PROSet will also be removed.

Command Line Parameters

DxSetup.exe supports the following command line parameters for fresh installations, modifying an installation, or upgrading the software:

PROSET

Installs Intel PROSet for Windows Device Manager or Intel PROSet ACU depending on your operating system.

0:

Do not install Intel PROSet. If Intel PROSet is already installed, it will be uninstalled.

1:

Install Intel PROSet feature.

> **Note:**
>
> - If `PROSET=0`, Intel ANS will not be installed.
> - If `PROSET=0` and Intel PROSet and Intel ANS are already installed, then Intel PROSet and Intel ANS will be uninstalled.

ANS

Installs Intel Advanced Network Services.

0:

Do not install Intel ANS. If Intel ANS is already installed, it will be uninstalled.

1:

Install Intel ANS. The `ANS` parameter requires `PROSET=1`.

> **Note:**
>
> If the `ANS` parameter is set to `ANS=1`, both Intel PROSet and Intel ANS will be installed.

ISCSI

0:

Do not install iSCSI. If iSCSI is already installed, it will be uninstalled.

1:

Install iSCSI. The `ISCSI` parameter requires `PROSET=1`.

PROSet_ACU_Shortcut

If Intel PROSet ACU is installed, creates or updates the "Intel® PROSet Adapter Configuration Utility" desktop shortcut.

0:

Do not create or update the desktop shortcut.

1:

Create or update the desktop shortcut. If this parameter is not specified, the shortcut will only be created or updated if it was created by a previous software installation.

RDMA_ROUTING

0:

Do not install RDMA routing.

1:

Install RDMA routing.

IWARP_FIREWALL

Installs the iWARP firewall rule.

0:

Do not install iWARP firewall rule.

1:

Install iWARP firewall rule. If `1` is selected, the following parameters are allowed in addition to `IWARP_FIREWALL`:

IWARP_FIREWALL_DOMAIN [0|1]:

Applies firewall rule to corporate domains.

IWARP_FIREWALL_PUBLIC [0|1]:

Applies firewall rule to public networks.

IWARP_FIREWALL_PRIVATE [0|1]:

Applies firewall rule to private networks.

FORCE

Forces the installer to install a feature, without checking for support.

0:

Check that the installed device supports a feature (for example, iSCSI) and only install the feature if such a device is found.

1:

Install the specified features regardless of the presence of supporting devices.

/q[r|n]

Silently installs the software. The following options are supported:

r:

Reduced GUI install (only displays critical warning messages).

n:

Silent install.

/l[i|w|e|a]

Specifies the log file option for Intel PROSet installation. The following log switches are supported:

i:

Log status messages.

w:

Log non-fatal warnings.

e:

Log error messages.

a:

Log the start of all actions.

/l* <pathfilename>

Use the log file ability of *SetupBD.exe*. If you do not specify the path and file name, *DxSetup.exe* creates a log file (named *SetupBD_<timestamp>.log*; for example, *SetupBD_18-04-2022_14-29-20.log*) in the same directory as the DxSetup/MSI log.

`/uninstall, /x`

Uninstall Intel PROSet.

Note:

- You must include a space between parameters.
- If you specify a path for the log file, the path must exist. If you do not specify a complete path, the install log will be created in the current directory.
- You do not need to specify default values. To install Intel ANS and Intel PROSet, the following examples are equivalent:

  ```
  DxSetup.exeDxSetup.exe PROSET=1 ANS=1
  ```

- The `ANS` parameter should only be set to `ANS=1` if `PROSET=1` is set.
- Even if `ISCSI=1` is passed, iSCSI using DCB will not be installed if the operating system and installed adapters do not support it. If `FORCE=1` is also passed, iSCSI will be installed if the operating system supports it.
- Public properties are not case sensitive. No white space is allowed between characters. For example, in the following, any white space in `PROSET=1` would make the setting invalid:

  ```
  DxSetup.exe /qn PROSET=1
  ```

Configuring Features with Windows PowerShell

You can use Microsoft PowerShell* cmdlets to configure and manage the Intel Ethernet devices present in your system. This functionality applies for the following:

- Intel® Ethernet cmdlets
- Intel® PROSet for Windows PowerShell* software

Note:

Refer to Compatibility Notes for Intel Ethernet Cmdlets and Intel PROSet for information on support limitations.

Module Names

The following table lists the module name in Windows PowerShell for each component.

Component	**Module Name**
Intel Ethernet cmdlets	IntelEthernetCmdlets
Intel PROSet for Windows PowerShell software	IntelNetCmdlets

Importing New Cmdlets

After installing Intel PROSet, use the `Import-Module` cmdlet to import the new cmdlets. You may need to restart Windows PowerShell to access the newly installed cmdlets. Refer to Module Names above for the available cmdlet modules.

To use the `Import-Module` cmdlet, you must specify the path. For example:

```
PS c:\> Import-Module -Name "C:\Program Files\Intel\Wired Networking\IntelNetCmdlets"
```

Note:

- If you include a trailing backslash (\) at the end of the `Import-Module` command, the import operation will fail. In Microsoft Windows 10 and Windows Server 2016, the auto-complete function appends a trailing backslash. If you use auto-complete when entering the `Import-Module` command, delete the trailing backslash from the path before pressing Return to execute the command.
- If you encounter issues with Intel Ethernet cmdlets, you may need to manually import the module using the instructions provided above.

See Microsoft TechNet for more information about the `Import-Module` cmdlet.

Changing Intel Ethernet Settings via Microsoft Windows PowerShell

You can use Windows PowerShell software to change most Intel Ethernet settings.

To configure Intel Ethernet device features using Windows PowerShell software, follow these general steps:

1. Install Intel PROSet or Intel Ethernet cmdlets, if you haven't already. See the following for more information:
   - Installing Intel Ethernet Cmdlets
   - Installing Intel PROSet
2. Open PowerShell.
3. At the PowerShell prompt, run your desired cmdlet.

Help Information for PowerShell Cmdlets

To get help information for both Intel Ethernet cmdlets and Intel PROSet for Windows PowerShell software:

- For a complete list of the cmdlets and their descriptions, type the following at the Windows PowerShell prompt. Refer to Module Names for the available cmdlet modules:

  ```
  PS C:\>get-help <module name>
  ```

- For detailed usage information for each cmdlet (including examples), type the following at the Windows PowerShell prompt:

  ```
  PS C:\> get-help <cmdlet_name> -full
  ```

- To show only examples for a cmdlet, type the following at the Windows PowerShell prompt:

  ```
  PS C:\> get-help <cmdlet_name> -examples
  ```

- To use the Minihelp property for any cmdlet in the module, append `| Select Minihelp`. For example:

  ```
  PS C:\> Get-IntelNetAdapterSetting -Name "<adapter_
  name>" -RegistryKeyword *RSS | Select Minihelp
  ```

Note: Online help (`get-help -online`) is not supported.

Additional Notes

- IntelNetCmdlets are digitally signed. Microsoft Windows operating systems check digital signatures online. Depending on your internet connection, this may result in a delay before any cmdlet operation (including get-help). If you have not already done so, make sure you use Import-Module to import the IntelNetCmdlets.

- If an adapter is bound to an Intel ANS team, do not change settings using the `Set-NetAdapterAdvanceProperty` cmdlet from Windows PowerShell, or any other cmdlet not provided by Intel. Doing so may cause the team to stop using that adapter to pass traffic. You may see this as reduced performance or the adapter being disabled in the Intel ANS team. You can resolve this issue by changing the setting back to its previous state, or by removing the adapter from the Intel ANS team and then adding it back.
- The `Get-IntelNetAdapterStatus -Status General` cmdlet may report the status "Link Up - This device is not linked at its maximum capable speed." In that case, if your device is set to auto-negotiate, you can adjust the speed of the device's link partner to the device's maximum speed. If the device is not set to auto-negotiate, you can adjust the device's speed manually, but you must ensure the link partner is set at the same speed.

Configuring Features with Intel PROSet Adapter Configuration Utility

Intel® PROSet Adapter Configuration Utility (Intel® PROSet ACU) is a graphical user interface that allows you to configure and manage supported Intel Ethernet Adapters.

Note:

Refer to Compatibility Notes for Intel Ethernet Cmdlets and Intel PROSet for information on support limitations.

To configure Intel Ethernet device features using Intel PROSet ACU, follow these general steps:

1. Select an adapter in the Adapter Selection panel.
2. Select a setting to configure from the Adapter Settings panel.
3. Select or enter the desired value(s) for the selected setting.
4. Click the **Apply Changes** button.

Configuring Features with Intel PROSet for Windows Device Manager

Intel® PROSet for Windows* Device Manager is an extension to the Windows Device Manager. When you install Intel PROSet for Windows Device Manager, additional tabs are automatically added to Device Manager.

Note:

> Refer to Compatibility Notes for Intel Ethernet Cmdlets and Intel PROSet for information on support limitations.

To configure Intel Ethernet adapter features using Intel PROSet for Windows Device Manager:

1. Open Device Manager in Windows.
2. Configure your desired settings in the appropriate tab in Device Manager.

Intel PROSet for Windows Device Manager uses the following additional Windows Device Manager tabs:

- **Link Speed**:
  - Allows you to change the adapter's speed and duplex setting, run diagnostics, and use the identify adapter feature.
- **Advanced**:
  - Allows you to customize how the adapter handles QoS packet tagging, Jumbo Packets, Offloading, and other capabilities. Some of the features might not be available depending on the operating system you are running, the specific adapters installed, and the specific platform you are using.
- **VLANs**:
  - Allows you to create, modify, and delete VLANs. You must install Intel Advanced Network Services (Intel ANS) in order to see this tab and use the feature. See Adapter Teaming for more information on Intel ANS.
- **Teaming**:
  - Allows you to create, modify, and delete adapter teams. You must install Intel Advanced Network Services in order to see this tab and use the feature. See Adapter Teaming for more information on Intel ANS.
- **Power Management**:
  - Replaces the standard Microsoft Windows Power Management tab in Device Manager. The standard Windows power management functionality is included on the Intel PROSet tab.

  > **Note:**
  >
  > - The options available on the Power Management tab are adapter and system dependent. Not all adapters will display all options. There may be BIOS or operating system settings that need to be enabled for your system to wake up. In particular, this is true for Wake from S5 (also referred to as Wake from power off).
  > - The Intel® Ethernet 500 Series does not support power management.
  > - If your system has a Manageability Engine, the Link LED may stay lit even if WoL is disabled.

Changing Intel Ethernet Settings Under Windows Server Core

You can use the Intel PROSet for Windows PowerShell software or Intel Ethernet cmdlets to change most Intel Ethernet settings under Windows Server Core. Please refer to their cmdlet help in PowerShell for more information.

For iSCSI Crash Dump configuration, use Intel PROSet for Windows PowerShell software and refer to the `aboutIntelNetCmdlets.help.txt` help file. iSCSI Crash Dump configuration is not supported in Intel Ethernet cmdlets.

Diagnostics in Intel PROSet

Intel's diagnostic software lets you test the adapter to see if there are problems with the adapter hardware, the cabling, or the network connection. Refer to About Intel PROSet® for an overview of this software.

Intel PROSet allows you to run the following types of diagnostic tests on supported Windows operating systems:

- **Connection test**: Verifies network connectivity by pinging the DHCP server, WINS server, and gateway.
- **Cable tests**: Provides information about cable properties.
- **Hardware tests**: Determines if the adapter is functioning properly.

Note:

The Cable Test is not supported on all adapters and will not run on Direct Attached Cables (DAC) or Fiber. The Cable Test will only be available on adapters that support it.

In Intel PROSet ACU, use the Diagnostics panel.

The availability of these tests is dependent on the adapter and operating system. Tests may be disabled if:

- The port is used as a manageability port.
- The tests are being run from a virtual machine.

Note:

At this time, Windows diagnostics are not supported on ports based on an Intel Ethernet Controller I225 and will fail.

Testing from Windows PowerShell

Intel provides two PowerShell cmdlets for testing your device:

- `Test-IntelNetDiagnostics` runs the specified test suite on the specified device. See the `Test-IntelNetDiagnostics` cmdlet help inside PowerShell plus the following table for more information.
- `Test-IntelNetIdentifyAdapter` blinks the LED on the specified device.

The following table describes the possible result codes from the `Test-IntelNetDiagnostics` cmdlet (where `<X>`, in the following table, indicates a value returned in the message).

ResultCode	Result Code	Description
RESULT_PING_GATEWAY_PASS	00	Successfully pinged gateway address <X>.
RESULT_PING_DHCP_PASS	01	Successfully pinged DHCP address <X>.
RESULT_PING_DNS_PASS	02	Successfully pinged DNS address <X>.
RESULT_PING_WINS_PASS	03	Successfully pinged WINS address <X>.
RESULT_TEST_ALREADY_RUNNING	04	An instance of this test is already running. You can only have one test running at a time.
RESULT_TCPIP_NOT_BOUND	05	TCP/IP protocol is not configured. To run this test configure this connection to use the TCP/IP protocol.
RESULT_INVALID_IP	07	The IP address for this connection is invalid. Possible cause: The system may be waiting for a response from a DHCP server.

ResultCode	Result Code	Description
RESULT_INVALID_ STATIC_IP	09	The IP address for this connection is invalid.
RESULT_ANS_BOUND	18	You cannot run this test while the device is included in a team or VLAN.
RESULT_FORCE_LOW_ SPEED	19	Auto-negotiation is not enabled on the device. The device has been configured to force a lower speed.
RESULT_AUTO_NEG_IN_ PROGRESS	20	Auto-negotiation is not complete on this device. Please wait and try again later.
RESULT_CAT5_CABLE_ NEEDED	21	A Category 5 (or better) cable is required to run at 1 Gbps. The cable connected to the device either is not Category 5 (or better) or has faulty wires.
RESULT_NOT_MAX_ SPEED	22	Link speed: <X> Mbps. The link partner is not capable of higher speeds.
RESULT_1000_NOT_ CHOOSEN	23	The link partner is not advertising a compatible speed. Please check that the link partner supports 1 Gbps.
RESULT_POWER_SAVER	24	

ResultCode	Result Code	Description
		Link speed has been reduced because a Power Saver option is enabled. Disable power saving in your OS settings and run the test again.
RESULT_SELF_FORCE_LOW_SPEED	25	The device is configured to force a lower speed.
RESULT_NO_CABLE_PROBLEMS	26	No cable problems detected.
RESULT_CANT_RUN_TEST	27	Could not run the test at this time. Please try again later.
RESULT_BAD_CONNECTION	33	The test detected a bad connection. Distance to problem: <X> meters.
RESULT_REMOTE_RECV_PASS	34	Passed
RESULT_REMOTE_RECV_FAIL	35	Failed
RESULT_REMOTE_RECV_UNKNOWN	36	Cable quality is unknown.
RESULT_REMOTE_LOCAL_PASS	37	Passed
RESULT_REMOTE_LOCAL_FAIL	38	Failed

ResultCode	Result Code	Description
RESULT_REMOTE_LOCAL_UNKNOWN	39	Cable quality is unknown.
RESULT_POLARITY_NORMAL	40	Cable polarity is normal.
RESULT_POLARITY_REVERSED	41	Cable polarity is reversed.
RESULT_CABLE_LENGTH	42	Cable length: <X> meters.
RESULT_FULL_SPEED	43	This device is running at full speed.
RESULT_POOR_AUTONEG	44	Cable quality is poor or no cable is connected. Possible causes: Faulty cable; faulty connector; or a speed/duplex mismatch. Verify that the speed/duplex setting on the switch/hub is configured for auto-negotiation.
RESULT_QUALITY_POOR	45	Cable quality is poor or no cable is connected. Possible causes: Faulty cable; faulty connector; or a speed/duplex mismatch.
RESULT_QUALITY_ADAQUATE	46	Cable quality is adequate.
RESULT_QUALITY_GOOD	47	Cable quality is good.

ResultCode	Result Code	Description
RESULT_QUALITY_ EXCELLENT	48	Cable quality is excellent.
RESULT_QUALITY_ UNKNOWN	49	Cable quality is unknown.
RESULT_QUALITY_IMPD_ MISMATCH	50	The test detected a frequency response that does not meet IEEE specifications.
RESULT_NO_LINK	51	This device does not have link. Make sure the cable is connected and the speed and duplex settings are configured correctly on the device and link partner.
RESULT_EEPROM_PASS	52	Passed
RESULT_FIFO_PASS	53	Passed
RESULT_REGISTER_PASS	54	Passed
RESULT_INTERRUPT_ PASS	55	Passed
RESULT_LOOPBACK_ PASS	56	Passed
RESULT_EXTENDED_ LOOPBACK_PASS	57	Passed
RESULT_EEPROM_FAIL	58	Failed

ResultCode	Result Code	Description
RESULT_FIFO_FAIL	59	Failed
RESULT_REGISTER_FAIL	60	Failed
RESULT_INTERRUPT_FAIL	61	Failed
RESULT_LOOPBACK_FAIL	62	Failed
RESULT_EXTENDED_ LOOPBACK_FAIL	63	Failed
RESULT_NVM_ INTEGRITY_PASS	64	Passed
RESULT_NVM_ INTEGRITY_FAIL	65	Failed
RESULT_DEVICE_NEEDS_ RESTART	66	This device requires a restart.
RESULT_BOOTED	67	Could not run the test at this time. The device may be connected to a remote target. Disconnecting the cable will enable the test.
RESULT_ MANAGEABILITY_SET	68	Could not run the test at this time. This device is used as a manageability port. Disconnecting the cable will enable the test.
RESULT_TEST_NOT_ SUPPORTED	69	The device does not support this test.

ResultCode	Result Code	Description
RESULT_CONNECTION_ TEST_FAILED	70	This test relies on a response from a gateway; DNS; DHCP; or WINS server and no such response was received. Any such server for this connection may be unavailable or misconfigured.
RESULT_UNKNOWN_ CONNECTION	71	Cable integrity is unknown.
RESULT_CONNECTION_ TEST_FAILED_APIPA	72	This test relies on a response from a gateway; DNS; DHCP; or WINS server and no such response was received. Any such server for this connection may be unavailable or misconfigured.This device is configured to automatically obtain an IP address but no DHCP server is present on the network. Windows selected an IP address using alternate private IP addressing.
RESULT_THERMAL_ SENSOR_PASS	73	Temperature is normal.
RESULT_THERMAL_ SENSOR_FAIL	74	Possible values: The device overheated and was stopped. The device overheated, and link speed was reduced.

Diagnostics in Ethernet Cmdlets

Intel's diagnostic software lets you test the adapter to see if there are problems with the adapter hardware. Refer to About Intel® Ethernet Cmdlets for an overview of this software.

Intel Ethernet Cmdlets allow you to run hardware diagnostic tests to determine the adapter is functioning properly, on supported Windows operating systems.

The availability of these tests is dependent on the adapter and operating system. Tests may be disabled if:

- The port is used as a manageability port.
- The tests are being run from a virtual machine.

Testing from Windows PowerShell

Intel provides the following PowerShell cmdlet:

- `Test-IntelEthernetHardware` runs hardware tests on the specified device. See the `Test-IntelEthernetHardware` cmdlet help inside PowerShell for more information.

Troubleshooting

This chapter describes troubleshooting tips and procedures, if you encounter issues with your Intel Ethernet device.

- Common Problems and Solutions
- Multiple Adapters
- Health Status Messages
- Safe Mode
- PF Message Queue Overflow
- Possible Misconfiguration of the Ethernet Port
- Other Performance Issues
- Diagnostics

Common Problems and Solutions

There are many simple, easy-to-fix problems related to network problems. Review each one of these before going further.

- **Check for recent changes to hardware, software, or the network that may have disrupted communications.**
- **Check the driver software.**

  - Make sure you are using the latest appropriate drivers for your adapter from the Intel support website.
  - Disable (or unload), then re-enable (reload) the driver or adapter.
  - Check for conflicting settings. Disable advanced settings such as teaming or VLANs to see if it corrects the problem.
  - Re-install the drivers.

- **Check the cable. Use the best available cabling for the intended data rate.**
  - Check that the cable is securely attached at both points.
  - Make sure the cable length does not exceed specifications.
  - Perform a cable test.
  - Replace the cable.
- **Check the link partner (switch, hub, etc.).**
  - Make sure the link partner is active and can send and receive traffic.
  - Make sure the adapter and link partner settings match one another, or are set to auto-negotiate.
  - Make sure the port is enabled.
  - Re-connect to another available port or another link partner.
- **Look for adapter hardware problems.**
  - Re-seat the adapter.
  - Insert the adapter in another slot.
  - Check for conflicting or incompatible hardware devices and settings.
  - Replace the adapter.
- **Check the Intel support website for possible documented issues.**
  - Select your adapter from the adapter family list.
  - Check the Frequently Asked Questions section.
  - Check the Knowledge Base.
- **Check your process monitor and other system monitors.**
  - Check to see that there is sufficient processor and memory capacity to perform networking activity.
  - Look for any unusual activity (or lack of activity).
  - Use network testing programs to check for basic connectivity.
- **Check your BIOS version and settings.**
  - Use the latest appropriate BIOS for your computer.
  - Make sure the settings are appropriate for your computer.

Additional Tips

The following troubleshooting tips assume that you have already reviewed the common problems and solutions earlier in this page,

Your computer cannot find the adapter.

Make sure your adapter slots are compatible for the type of adapter you are using.

Diagnostics pass but the connection fails.

Make sure the cable is securely attached, is the proper type and does not exceed the recommended lengths. Also, ensure the duplex mode and speed setting on the adapter matches the setting on the switch.

Adapter unable to connect to switch at correct speed.

> **Note:** This applies only to copper-based connections.

Make sure the adapter and the link partner are set to auto-negotiate.

Verify that you are running the latest operating system revision for your switch and that the switch is compliant with the proper IEEE standard:

- IEEE 802.3ad-compliant (gigabit over copper)
- IEEE 802.3an-compliant (10 gigabit over copper)

The device does not connect at the expected speed.

When Gigabit PHY Mode is forced to Primary mode on both the Intel adapter and its link partner, the link speed obtained by the Intel adapter may be lower than expected or link may not be established.

The adapter stops working without apparent cause.

Run the adapter and network tests described in :doc:../tools/proset/*diags-in-proset.rst*.

The Link indicator light is off.

- Run the adapter and network tests described in Diagnostics in Intel PROSet.
- Make sure the proper (and latest) driver is loaded.
- Make sure that the link partner is configured to auto-negotiate (or forced to match adapter)
- Verify that the switch is IEEE 802.3ad-compliant.

The link light is on, but communications are not properly established.

Make sure the proper (and latest) driver is loaded.

Both the adapter and its link partner must be set to either auto-detect or manually set to the same speed and duplex settings.

> **Note:**

> The adapter's link indicator light may be on even if communications between the adapter and its link partner have not been properly established. Technically, the link indicator light represents the presence of a carrier signal but not necessarily the ability to properly communicate with a link partner. This is expected behavior and is consistent with IEEE's specification for physical layer operation.

RX or TX light is off.

Network may be idle; try creating traffic while monitoring the lights.

The diagnostic utility reports the adapter is "Not enabled by BIOS."

The PCI BIOS isn't configuring the adapter correctly. See "PCI/PCI-X/PCI Express Configuration" later in this section.

The computer hangs when the drivers are loaded.

Try changing the PCI BIOS interrupt settings. See "PCI/PCI-X/PCI Express Configuration" later in this section.

The Fan Fail LED of the 10 Gigabit AT Server Adapter is on (red).

The fan cooling solution is not functioning properly. Contact customer support for further instructions.

PCI/PCI-X/PCI Express Configuration.

If the adapter is not recognized by your OS or if it does not work you may need to change some BIOS settings. Try the following only if you are having problems with the adapter and are familiar with BIOS settings:

- Check to see that the "Plug-and-Play" setting is compatible with the operating system you are using.
- Make sure the slot is enabled.
- Configure interrupts for level-triggering, as opposed to edge-triggering.
- Reserve interrupts and/or memory addresses. This prevents multiple buses or bus slots from using the same interrupts. Check the BIOS for IRQ options for PCI / PCI-X / PCIe.

Driver message: "Rx/Tx is disabled on this device because an unsupported SFP+ module type was detected."

You installed an unsupported module in the device. See the Feature Support Matrix for your device for more information on supported media types.

Multiple Adapters

When configuring a multi-adapter environment, you must upgrade all Intel adapters in the computer to the latest software.

If the computer has trouble detecting all adapters, consider the following:

- If you enable Wake on LAN (WoL) on more than two adapters, the Wake on LAN feature may overdraw your system's auxiliary power supply, resulting in the inability to boot the system and other unpredictable problems. For multiple desktop/management adapters, it is recommended that you install one adapter at a time and use the BootUtil utility (in `\APPS\BootUtil`) to disable the WoL feature on adapters that do not require WoL capabilities. On server adapters, the WoL feature is disabled by default.
- Adapters with Intel® Boot Agent enabled will require a portion of the limited startup memory for each adapter enabled. Disable the service on adapters that do not need to boot Preboot Execution Environment (PXE).

Health Status Messages

Intel® Ethernet 800 Series devices support asynchronous health status messages, which help you to debug system-level issues and diagnostics in the field. This feature is enabled by default and cannot be shut off.

When the firmware detects an abnormal event during initialization, it will push health status information to the base driver's system event log, such as dmesg or the Windows Event Log. Health status messages could encompass issues related to:

- Unsupported modules
- The NVM or option ROM
- Invalid link configuration
- Port speed
- Link partner
- Other issues

The system log will identify the device experiencing the issue, list information about the problem, and suggest a possible solution, such as updating to the latest NVM image or checking the cable or module.

See your system log for more information, if you are experiencing issues on Intel Ethernet 800 Series devices.

Safe Mode

Adapters based on the Intel® Ethernet 800 Series require a Dynamic Device Personalization (DDP) package file to enable advanced and performance features. If the driver detects a missing or incompatible DDP package file, the driver will go into Safe Mode. Safe Mode supports only basic traffic and minimal functionality, such as updating the NVM or downloading a new driver or DDP package.

Note:

- Safe Mode only applies to the affected physical function and does not impact any other PFs.

- Firmware Recovery Mode takes precedence over Safe Mode.

Safe Mode Errors and Messages

When the driver is in Safe Mode, the device drivers and utilities may log or display messages to help with troubleshooting. The following conditions may cause the driver to enter Safe Mode:

- The DDP package file was not found or couldn't be read.
- The DDP package file's version number, signature, or other metadata either aren't valid or aren't supported by the driver.
- An unknown error occurred when loading the DDP package.
- The driver couldn't load the DDP package file because a compatible DDP package is already present on the device.
- The device has a DDP package that isn't supported by the driver.

Resolving Safe Mode Issues

The device drivers and utilities may display the action to take to get out of Safe Mode, depending on the underlying cause. Possible actions could include the following:

- Wait for the device to reset.
- Install the latest driver.
- Download a new DDP package.
- Restart the adapter. If the problem persists, install the latest driver.
- Reboot the system. If the problem persists, update the NVM.

You can download the latest drivers and DDP packages from the Intel support website.

PF Message Queue Overflow

The device driver can detect some types of anomalous behavior. When it does, it will log the VF MAC address and associated PF MAC address. Using this information, you can check the virtual machine (VM) that is using the VF MAC address to ensure that the VM is operating correctly.

Possible Misconfiguration of the Ethernet Port

You may see an informational message stating that a potential misconfiguration of the Ethernet port was detected. This is to alert you that your device is being underutilized. If this was intentional, you may ignore this message.

For example, setting your Intel® Ethernet Network Adapter E810-C-Q2 adapter to *2x2x25* is valid, but it does not use the full capabilities of the device. If you see this message and the configuration was not intentional, you may use the Ethernet Port Configuration Tool (EPCT) to correct the configuration. See Ethernet Port Configuration Tool (EPCT) for more information on this tool.

Other Performance Issues

Attaining the best speed requires that many components are operating at peak efficiency. Among them are the following:

Cable quality and length

Do not exceed the maximum recommended length for your cable type. Shorter lengths often provide better results. Check for loose or damaged connectors. Check the cable for kinked or damaged sections.

Bus speed and traffic

The PCI bus speed accommodates the slowest PCI card installed. Check to see if you have a card that is slowing down your system.

Processor and memory

Check your performance monitoring programs to see if traffic is being affected by your processor speed, available memory or other processes.

Transmission frame size

Your network performance may be enhanced by adjusting or maximizing the transmission frame size. Operating systems, switches and adapters will impose varying limits on maximum frame size. See the discussion on Jumbo Frames for your OS.

Operating system

Networking feature implementation will vary by operating system version, such as offloading and multiprocessor threading.

Diagnostics

Multiple utilities are available to help troubleshoot and diagnose issues with your Intel Ethernet devices.

Linux Diagnostics

Intel Ethernet drivers use the *ethtool* interface for driver configuration and diagnostics, as well as displaying statistical information. The latest ethtool version is required for this functionality. Download it at ethtool.

Diagnostics in Intel PROSet

In Microsoft Windows, you can use Intel PROSet to conduct diagnostic software tests. See Diagnostics in Intel PROSet for details.

Diagnostics in Intel Ethernet Cmdlets

In Microsoft Windows, you can use Intel Ethernet Cmdlets to conduct diagnostic software tests. See Diagnostics in Ethernet Cmdlets for details.

Specifications

Note:

As of Release 29.1, this user guide no longer contains the following information related to hardware:

- Indicator lights
- Regulatory compliance statements

This information is always available in the product briefs for Intel Ethernet adapters, which are available publicly on intel.com. Search for the device's branding string (such as "Ethernet Network Adapter E810-C-Q1") plus "product brief" to view this information.

Indicator Lights

Information on indicator lights is no longer included in this software user guide.

See the hardware documentation for your device for details on indicator lights.

Regulatory Compliance Statements

Information on regulatory compliance and FCC certification classes is no longer included in this software user guide.

See the hardware documentation for your device for details on regulatory compliance.

Customer Support

Intel support is available on the web or by phone. Support offers the most up-to-date information about Intel products, including installation instructions, troubleshooting tips, and general product information.

Web and Internet Sites

Support: https://www.intel.com/support

Corporate Site for Network Products:
https://www.intel.com/products/ethernet/overview.htm

Notices and Disclaimers

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Your costs and results may vary.

Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

Performance varies by use, configuration, and other factors. Learn more at https://www.Intel.com/PerformanceIndex.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This software and the related documents are Intel copyrighted materials, and your use of them is governed by the express license under which they were provided to you (“License”). Unless the License provides otherwise, you may not use, modify, copy, publish, distribute, disclose or transmit this software or the related documents without Intel’s prior written permission.

This software and the related documents are provided as is, with no express or implied warranties, other than those that are expressly stated in the License.

Refer to the software download for this release for all licenses.

The Intel Proprietary License applies to all files on this product unless there is another license in the same folder as a file or files. If another license is present, then that license shall take precedence and apply to all files in the same folder.

In the event that the English language version of the Intel Proprietary license is accompanied by any other version translated into any other language, such translated version is provided for convenience purposes only and the English language version shall control.